GC28-1336-0
File No. S370-40

Program Product

MVS/Extended Architecture Message Library: Dump Output Messages

MVS/System Product:

JES3 Version 2	**5665-291**
JES2 Version 2	**5740-XC6**

IBM

First Edition (June, 1987)

This edition applies to Version 2 Release 2.0 of MVS System Product 5665-291 or 5740-XC6 and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370 Bibliography*, GC20-0001, for the editions that are applicable and current.

References in this publication to IBM products or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product in this publication is not intended to state or imply that only IBM's product may be used. Any functionally equivalent product 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 Corporation, Information Development, Department D58, Building 921-2, PO Box 390, Poughkeepsie, NY 12602. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1987

Preface

Dump Output Messages

This publication contains the messages that can appear during dump processing, and provides a single reference that the user can turn to during dump analysis. This publication contains messages from the following operating system components:

- Service Aids Components (AMD)
- SYMPTOM string/record
- IPCS/PRDMP
- Real Storage Manager (RSM)
- MVS Components (IEA)
- Scheduler (IEE)
- TSODATA (IKJ)
- Input Output Supervisor (IOS)
- System Resource Manager (SRM)
- Global Resource Serialization
- Data-In-Virtual
- Auxiliary Storage Manager (ASM)
- Virtual Storage Manager (VSM)

Message Descriptions

The messages are grouped in sections by their three-letter prefixes, which identify the components producing the messages. The prefixes are in alphabetic order, while in each message section, the messages are in alphanumeric order.

Each section containing numbered messages begins with a chart containing the component name, the audience for the messages and where they appear, the message format, publications related to or referenced by the messages, and other useful information.

The descriptions associated with each message provide the following information: an explanation of why the component issued the message, a series of actions that the operating system performs, and suggested diagnostic steps that the system programmer can follow.

Unnumbered Messages

This publication also contains unnumbered messages issued by Auxiliary Storage Manager (ASM) and Virtual Storage Manager (VSM).

Problem Determination Tables

Most of the message descriptions also contain a section devoted to problem determination. This portion of the explanation lists suggested actions as items in tables; you can find the problem determination tables immediately following the unnumbered VSM message section at the back of this book. You should perform the actions for the items listed in the message description before asking the IBM branch office serving your locality for programming or hardware support.

Message-to-Module Table

The message-to-module table at the back of this book lists the operating system modules that **detect** the need for the message, **issue** the message, and **contain** the message text.

Who Should Use This Book:

System programmers can use this book when analyzing dumps.

Contents

Service Aids Messages (AMD) AMD-1
AMDPRDMP Messages AMD-1

Symptom Record Messages (ASR) ASR-1

Interactive Problem Control System (IPCS) Messages (BLS) BLS-1
BLS01 Messages BLS-2
BLS03 Messages BLS-5
BLS04 Messages BLS-9
BLS05 Messages BLS-19
BLS06 Messages BLS-20
BLS07 Messages BLS-20
BLS17 Messages BLS-20
BLS18 Messages BLS-22
BLS19 Messages BLS-40
BLS20 Messages BLS-41
BLS21 Messages BLS-41
BLS22 Messages BLS-48

Real Storage Manager Messages (IAR) IAR-1

MVS Component Messages (IEA) IEA-1
IEA11 Messages (Supervisor) IEA-1
IEA21 Messages (RTM) IEA-2
IEA24 Messages (LOGREC) IEA-3
IEA31 Messages (CommTask) IEA-4

Master Scheduler Messages (IEE) IEE-1

TSO Programmer Messages (IKJ) IKJ-2

Input/Output (I/O) Supervisor Messages (IOS) IOS-1

System Resources Manager Messages (IRA) IRA-1

Global Resource Serialization Messages (ISG) ISG-1

Data-In-Virtual Messages (ITV) ITV-1

Unnumbered Messages Issued By Auxiliary Storage Manager (ASM) ASM-1

Unnumbered Messages Issued By Virtual Storage Manager (VSM) VSM-1

Problem Determination PD-1

TABLE I PD-1
TABLE II PD-4
TABLE III PD-5
TABLE IV PD-6
Problem Directory Records PD-6
Data Set Directory Records PD-6

Message to Module Table A-1

Summary of Amendments

Summary of Amendments
for GC28-1336-0
As Updated June 1987

This publication is new for MVS System Product Version 2 Release 2.0. It contains messages that can appear during dump processing. Some of these messages previously appeared in *MVS/XA System Messages* and the *MVS/XA Debugging Handbook*.

Service Aids Messages (AMD)

Component Name	AMD
Program Producing Message	Service Aids: AMDPRDMP
Audience and Where Produced	For the programmer: SYSPRINT data set. For the operator: console.
Message Format	AMDnnns text (in SYSPRINT) id nnn message serial number s Type code: A Action: the operator must perform a specific action. D Decision: the operator must choose an alternative. I Information: no operator action is required. text Message text.
Associated and Referenced Publication	*MVS/XA SPL: Service Aids*, GC28-1159

AMDPRDMP Messages

AMD150I FILE ddname CANNOT BE OPENED

Explanation: Either the data set associated with **ddname** could not be opened, or the required PRINTER DD statement is missing.

System Action: AMDPRDMP execution terminates.

Programmer Response: Probable user error. Ensure that a PRINTER DD statement is present and verify that all DD statements are correct.

Problem Determination: Table I, items 1, 2, 15, 29.

AMD151I PRINT DUMP TERMINATED - NOT ENOUGH STORAGE - INCREASE REGION SIZE

Explanation: Either the region size was less than the minimum required for execution or storage for buffer space was not available.

System Action: AMDPRDMP processing terminates.

Programmer Response: Increase the region size.

Operator Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 29. Save a listing of the control statements for AMDPRDMP in the order the control statements were executed.

AMD153I PERMANENT I/O ERROR ON ddname - EXECUTION TERMINATED

Explanation: An I/O error occurred on the device assigned to the data set **ddname**, and the associated DCB SYNAD routine has been entered.

System Action: AMDPRDMP execution terminates.

Programmer Response: Check **ddname** and make sure that that the proper device is specified.

Problem Determination: Table I, items 1, 2, 13, 29.

AMD158I to AMD165I

AMD158I **I/O ERROR ON DUMP**

Explanation: An I/O error occurred while AMDPRDMP was attempting to read a block from the input dump data set.

System Action: The system does not dump any more of the storage area named on the control statement it was processing when the I/O error occurred. AMDPRDMP checks the syntax of any remaining control statements but does not execute any until it encounters a valid NEWDUMP, NEWTAPE, or END control statement.

Operator Response: None.

Programmer Response: None.

AMD161I **FORMAT ERROR**

Explanation: An error occurred during the formatting of a control block.

System Action: Some data is not formatted and the control statement processing either terminates or moves on to the next logical report.

Programmer Response: None.

AMD162I **JOB jobname NOT FOUND**

Explanation: **jobname** was specified in a PRINT JOBNAME=jobname user control statement. The job could not be found in the storage dump.

System Action: AMDPRDMP execution continues with the next user control statement.

Programmer Response: None.

AMD163I **GO FUNCTIONS TO BE PERFORMED [ONGO operands]**

Explanation: On execution of a GO control statement, message AMD163I lists the functions to be performed.

System Action: The system performs the GO functions listed in the message. If the user previously issued the ONGO control statement the specified functions appear in this message. If the user did not specify an ONGO control statement the system performs the default functions, EDIT, SUMMARY, and PRINT CURRENT.

Programmer Response: None.

AMD164I **TAPE IS PRE-FORMATTED DUMP REMAINING PARAMETERS IGNORED**

Explanation: AMDPRDMP has determined that the input tape data set is not an AMDSADMP high-speed or system produced dump. The input block size is less than 134 characters.

System Action: AMDPRDMP prints the contents of the input tape with no formatting. The system ignores the current user control statement and obtains the next control statement.

Programmer Response: Probable user error. If user control statements are being entered from the system console, enter the NEWDUMP or NEWTAPE control statement to demount the current tape volume. Otherwise rerun AMDPRDMP using the correct dump tape.

Problem Determination: Table I, items 2, 13, 29.

AMD165I **ERROR IN PRECEDING CONTROL STATEMENT [error description]**

Explanation: A syntax error was detected during the scan of an AMDPRDMP control statement. If the AMDPRDMP program cannot diagnose the control statement error, the message will not contain an **error description**. Otherwise, **error description** can be one of the following:

ASID OMITTED FROM LIST
An operand was missing from the ASID subfield of the PRINT STORAGE statement.

ASID 0 OR FFFF IS NOT ALLOWED
ASIDs of 0 and FFFF are not valid ASIDs (they could not have been created by the operating system) and have special internal meanings to AMDPRDMP.

DELIMITER ERROR IN JOBNAME OPERAND LIST
The job names specified by the JOBNAME keyword must be separated by commas.

DELIMITER ERROR IN OPERAND FIELD OF aaaaaaaa
A delimiter error was encountered by an AMDPRDMP exit module in the operand field of its own verb; **aaaaaaaa** is replaced by that verb.

DELIMITER ERROR IN REAL OPERAND
Parameters of the REAL keyword of the PRINT verb must be separated by commas. During the scan, an invalid delimiter was found; or parentheses were found to have been omitted or incorrectly used.

DELIMITER ERROR IN STORAGE OPERAND
Parameters of the STORAGE keyword for the PRINT verb are not separated by commas; or parentheses are omitted or incorrectly used.

FILE OPERATION CANNOT BE PERFORMED ON D/A INPUT
The FILESEQ keyword parameter was used in the NEWDUMP control statement, but the corresponding dump data set was on a direct access device.

GO PARAMETER ENCOUNTERED IN ONGO OPERAND
The GO verb may not be specified as a verb in the ONGO list.

INVALID CVT ADDRESS SPECIFIED
The value specified for the CVT verb is invalid. This value must be specified as a 1 to 6 digit hexadecimal address.

INVALID DELIMITER FOLLOWING KEYWORD
A keyword is followed by an invalid delimiter.

INVALID DELIMITER FOLLOWING VERB
The delimiter separating two verbs or a verb and a keyword is invalid.

INVALID KEYWORD
A keyword is invalid.

INVALID KEYWORD IN OPERAND FIELD OF aaaaaaaa
The AMDPRDMP exit module encountered an invalid keyword in the operand field of its own verb. **aaaaaaaa** contains the verb.

INVALID NUMERIC IN REAL OPERAND
One of the digits in a real storage address did not fall within the range 0-9 or A-F.

INVALID NUMERIC IN STORAGE OPERAND
One of the digits in either an ASID or a storage address did not fall within the range 0-9 or A-F.

INVALID SEGTAB ADDRESS SPECIFIED
The value specified for the SEGTAB verb is invalid. This value must be specified as a 1 to 6 digit hexadecimal address.

INVALID VERB
AMDPRDMP does not recognize the verb of a control statement.

JOBNAME LENGTH GREATER THAN 8
A job name specified in the JOBNAME operand has a length greater than 8 characters.

KEYWORD LENGTH GREATER THAN 8
A keyword was found that has more than 8 characters.

LENGTH OF ASID IS GREATER THAN 4
The ASID specified in the storage operand list is longer than 4 hexadecimal digits.

LENGTH OF REAL ADDRESS GREATER THAN 8
A storage address in the REAL operand list was specified with more than 8 hexadecimal digits.

LENGTH OF STORAGE ADDRESS GREATER THAN 8
A virtual storage address, in the STORAGE operand list, is specified with more than 8 hexadecimal digits.

NAME MISSING FROM JOBNAME OPERAND LIST
No job name is specified for the JOBNAME keyword of the PRINT verb.

NEWDUMP KEYWORD VALUE ERROR
There is syntax error in the keyword parameters of the NEWDUMP control statement.

NO INPUT DD CARD
The value of the DDNAME keyword of the NEWDUMP control statement specified a DD statement which is not included with the JCL statements used to execute the AMDPRDMP service aid program.

OPERAND MISSING
The value for a keyword parameter is not specified.

POSITION 72 MUST BE BLANK
Control statements are coming from the reader and column 72 is not blank. Verbs must be entered in columns 1-71 only. After processing this message, AMDPRDMP will blank position 72 and attempt to process the statement.

STARTING ADDRESS IS NOT LESS THAN ENDING ADDRESS
An address pair was specified in which the first address was larger than the second.

STARTING ASID IS NOT LESS THAN OR EQUAL TO ENDING ASID
On a PRINT STORAGE request, an ASID range was specified in which the second ASID is less than the first ASID.

STARTING OMITTED IN PAGE OPERAND
TTR or SGG values in the operand of the PAGE keyword must appear in pairs. The first address of a pair was omitted.

STARTING OMITTED IN REAL STORAGE OPERAND
Values in the operand of the REAL keyword must appear in pairs. An odd number of addresses was specified in this list.

STARTING OMITTED IN STORAGE OPERAND
A virtual storage address in the STORAGE keyword must appear in pairs. The first address of a pair was omitted.

SYNTAX ERROR IN OPERAND FIELD OF aaaaaaaa
A syntax error was encountered by an AMDPRDMP exit module in the operand field of its own verb; aaaaaaaa is replaced by that verb.

TOO MANY JOBNAMES IN LIST
More than ten job names were specified for the JOBNAME keyword of the PRINT verb.

VERB LENGTH GREATER THAN 8.
The verb is longer than 8 characters.

System Action: AMDPRDMP ignores the verb or keyword it was scanning and looks for the next verb or keyword. If there are no more control statements for AMDPRDMP to examine and the SYSIN data set is providing the control statements, the next control statement is read. If the error occurred on a NEWDUMP control statement, the remaining statements will be scanned for SYNTAX errors, but no execution will take place until a correct NEWDUMP, NEWTAPE or END control statement is encountered. If user control statements are being entered from the system console, AMDPRDMP issues either message AMD155D or AMD283D to allow the operator to enter a new control statement.

Programmer Response: Probable user error. If the control statements entered by way of the card reader, rerun the job specifying the control statements in the proper syntax.

Problem Determination: Table I, items 2, 13, 29.

AMD166I FORMAT ERROR DURING JOBNAME SEARCH

Explanation: AMDPRDMP encountered a format error while attempting to locate the virtual storage assigned to a specified job.

System Action: AMDPRDMP execution continues and, if possible, the search is continued.

Operator Response: None.

AMD168I DUMP DATA SET EMPTY - DD ddname

Explanation: The dump data set described by DD statement **ddname** does not contain a core image dump or a preformatted dump.

System Action: If the user control statements are being entered from the SYSIN data set, AMDPRDMP scans the remaining control statements for syntax errors. The system will not execute any control statements until AMDPRDMP encounters a correct NEWDUMP or NEWTAPE statement. If user control statements are being entered from the console, the system issues message message AMD283D to allow the user to enter a new AMDPRDMP control statement.

Operator Response: Probable user error. AMDPRDMP cannot process the current input data set. If AMDPRDMP is to process additional dumps, the NEWDUMP or NEW TAPE control statements may be used to specify a different input data set. Otherwise, reply 'END' to message AMD283D to terminate the execution of AMDPRDMP.

AMD170I END OF FILE ON SYSIN - CONTROL PASSED TO OPERATOR

Explanation: AMDPRDMP has processed all user control statements in the SYSIN data set without encountering an END control statement.

System Action: AMDPRDMP issues message AMD155D to the system console.

Programmer Response: None.

AMD171I PROCESSING FOR CURRENT DUMP DISCONTINUED

Explanation: An error occurred that prevents further processing on the current dump.

System Action: AMDPRDMP scans all remaining user control statements in the SYSIN data set, but does not execute any of them until it encounters a valid NEWDUMP, NEWTAPE, or END verb.

Programmer Response: None.

AMD172I FUNCTION TERMINATED BY OPERATOR

Explanation: STOP has been entered in reply to message AMD156I.

System Action: AMDPRDMP ceases processing the current function statement and obtains the next user control statement.

Programmer Response: None.

AMD173I **SYSUT1 IS NOT DA - DUMP WILL BE PROCESSED ON TAPE**

Explanation: The SYSUT1 data set was not assigned to a direct access device.

System Action: Instead of processing the dump data set on direct access storage, AMDPRDMP processes the dump on the input tape.

Programmer Response: None.

AMD174I **[SYSUT1|SYSUT2] LOADED**

Explanation: The work data set indicated in the message text has been loaded from the input data set.

System Action: If the work data set is SYSUT1, module AMDPREAD will get the dump information from the SYSUT1 data set. If the work data set is SYSUT2, the system has successfully copied the data set and execution terminates.

Programmer Response: None.

AMD175I **NO TAPE DD CARD - SYSUT1 ASSUMED LOADED**

Explanation: A TAPE DD statement is not included in the AMDPRDMP JCL.

System Action: AMDPRDMP assumes that the dump to be processed is in the data set described by the SYSUT1 DD statement. AMDPRDMP attempts dump processing on this data set.

Programmer Response: None.

AMD177I **THESE MODULES NOT FOUND**
AMD177I **mod**
AMD177I **mod**

Explanation: AMDPRDMP was unable to locate any **mod** in either SYS1.LINKLIB or in a private library.

System Action: AMDPRDMP suppresses the function that required the use of **mod**, issues message AMD180I to indicate which function cannot be used, and continues processing with one of the following actions:

- If the module named in message AMD180I is required for execution of a control statement, that control statement is not executed, and AMDPRDMP proceeds with the next control statements.
- If the module is a user exit program for the EDIT function of AMDPRDMP, the system issues message AMD214I to indicate that EDIT processing will terminate. AMDPRDMP continues with the next control statement.
- If the module is a format appendage for the EDIT function then trace records requiring this appendage will be printed in the hexadecimal dump format.

Operator Response: Check the module names **mod** identified in message AMD177I and inform the system programmer that the system issued message AMD177I.

Programmer Response: If the module name **mod** in message AMD177I has the format AMDUSRxx, xx being a hexadecimal number in the range 1-50, the module is a user format appendage for the EDIT function. Trace records requiring this module are user trace records for which the hexadecimal dump may be desired. If this is the case, no programmer action is required.

If the module name **mod** is not of the format AMDUSRxx, or if the user format appendage is desired, message AMD177I indicates a probable user error. The required AMDPRDMP module must be link edited into a private library or the SYS1.LINKLIB data set.

Problem Determination: Table I, items 2, 4, 29.

AMD178I **I/O ERROR ON BLDL**

Explanation: AMDPRDMP issued a BLDL macro instruction to locate a required module. The BLDL macro instruction encountered an input/output error.

System Action: AMDPRDMP suppresses the function that required use of the module, issues message AMD180I to indicate which function cannot be used, and continues processing with one of the following actions:

- If the module named in message AMD180I is required for execution of a control statement, then the system does not execute the control statement, and AMDPRDMP proceeds with the next control statement.
- If the module named in message AMD180I is a user exit program or a format appendage for the EDIT function of AMDPRDMP, message AMD214I is issued to indicate that EDIT processing will terminate. AMDPRDMP proceeds with the next control statement.

Operator Response: Inform the system programmer that the system issued message AMD214I.

Programmer Response: Verify that the AMDPRDMP modules have been link edited correctly into the private library or SYS1.LINKLIB data set.

Problem Determination: Table I, items 1, 4, 29.

AMD180I **mod FUNCTION INOPERATIVE**

Explanation: The **mod** function of AMDPRDMP is inoperative. **mod** can be:

- required for the execution of an AMDPRDMP control statement. In this case, the module name will have the format AMDPRxxx where xxx is one of:

  PCR - PRINT CURRENT function
  FXT - FORMAT function
  PJB - PRINT JOBNAME= (print storage by specific jobname)
  LPA - LINK PACK AREA map function
  SCN - EDIT GTF trace data (control card scan phase)
  XED - EDIT GTF trace data (processing phase)
  NUC - PRINT NUCLEUS, SQA, and CSA functions
  PMS - PRINT STORAGE= or PRINT REAL

- a system or subsystem format appendage routine for the EDIT function of AMDPRDMP. In this case, the module name has the format AMDSYSxx where xx is a hexadecimal number, or AMDUSRyy where yy is a hexadecimal number in the range 51-FF.

- a user exit program for the EDIT function of AMDPRDMP as specified by the EXIT parameter of the EDIT control statement.

System Action: AMDPRDMP may take a variety of actions depending on which function became inoperative. The system issues either message AMD177I or AMD178I prior to issuing message AMD180I. Both of these message indicate the reason for the function becoming inoperative and the resulting system action.

Operator Response: Follow the recommended programmer response indicated for either message AMD177I or AMD178I.

Problem Determination: Table I, items 25c, 29.

AMD181I **DELETE ERROR - mod**

Explanation: During execution of the EDIT function, AMDPRDMP attempted to acquire storage for the load of a program segment by deleting the loaded module **mod**., but **mod** had already been deleted.

System Action: EDIT execution terminates; AMDPRDMP processing proceeds with the next control statement.

Operator Response: Report the occurrence of message AMD181I to the programmer.

Programmer Response: Probable user error. User exit programs and user format appendages for the EDIT function must not issue the DELETE macro instruction specifying modules of AMDPRDMP. Verify that this is not done. Rerun the job including a SYSPRINT DD statement in the AMDPRDMP JCL.

Problem Determination: Table I, items 1, 2, 4, 29.

AMD187I **INVALID EXEC CARD PARAMETER**

Explanation: The AMDPRDMP program detected a syntax error in the value of the PARM= parameter of its EXEC JCL statement.

System Action: AMDPRDMP processing continues; the value assumed for this parameter may be in error.

Programmer Response: Probable user error. Correct the value of the PARM= parameter, and make sure that a SYSPRINT DD statement has been included with the AMDPRDMP JCL.

Problem Determination: Table I, items 1, 4, 29.

AMD201I **INVALID KEYWORD BEGINNING WITH xxx**

Explanation: While scanning a control statement, EDIT has encountered an invalid keyword. The first three characters of that keyword are indicated by xxx.

System Action: If control statements are being provided by the SYSIN data set, the function requested by the control statement in error will not be executed. Further action depends on the type of data set being processed:

1. If a dump data set is being processed, AMDPRDMP will syntax-check the remaining control statements for that dump without executing them.

2. If an external trace data set is being processed, AMDPRDMP will resume processing with the next control statement.

If user control statements are being entered from the primary system console, AMDPRDMP issues message AMD210D to allow the user to enter EDIT keywords.

Operator Response: If the control statements are being entered by way of the system console, reenter the EDIT keywords with correct syntax.

Programmer Response: Probable user error. If the control statements are being provided by the SYSIN data set, rerun the job using valid keywords and abbreviations.

Problem Determination: Table I, items 2, 4, 29.

AMD202I **INVALID PARENTHESES**

Explanation: While scanning a control statement, EDIT encountered either unbalanced parentheses or parentheses around keyword values for which only one value may be specified.

System Action: If control statements are being provided by the SYSIN data set, the function requested by the control statement in error will not be executed. Further action depends on the type of data set being processed:

1. If a dump data set is being processed, AMDPRDMP will syntax-check the remaining control statements for that dump without executing them.

2. If an external trace data set is being processed, AMDPRDMP will resume processing with the next control statement.

If user control statements are being entered from the primary system console, AMDPRDMP issues message AMD210D to allow the operator to enter EDIT keywords.

Operator Response: If entering the control statements from the system console, reenter the EDIT keywords correctly.

Programmer Response: Probable user error. If the control statements were entered by way of the SYSIN data set, rerun the job, making sure that all parentheses are paired and that no parentheses are used with the DDNAME and EXIT keywords.

Problem Determination: Table I, items 2, 4, 29.

AMD203I **INVALID PARM VALUE FOR KEYWORD keywd**

Explanation: While scanning the keyword parameter - keywd, EDIT encountered a value that contains other than valid alphameric values or a value that falls outside the range of values allowed for that keyword.

System Action: If control statements are being provided by the SYSIN data set, the function requested by the control statement in error will not be executed. Further action depends on the type of data set being processed:

1. If a dump data set is being processed, AMDPRDMP will syntax-check the remaining control statements for that dump without executing them.

2. If an external trace data set is being processed, AMDPRDMP will resume processing with the next control statement.

If user control statements are being entered from the primary system console, AMDPRDMP issues message AMD210D to allow the operator to enter EDIT keywords.

Operator Response: If the control statements are being entered by way of the system console, reenter the EDIT keywords, making sure that all the errors mentioned above have been corrected.

Programmer Response: Probable user error. If the control statements are being provided by the SYSIN data set, rerun the job, making sure that alphabetic and numeric characters are used correctly, and that all parameters fall within the range of values allowed.

Problem Determination: Table I, items 2, 4, 29.

AMD204I **LENGTH OF PARM INVALID FOR KEYWORD keywd**

Explanation: While scanning parameter values associated with keyword (keywd), EDIT encountered a parameter value that exceeds the maximum length allowed for parameters of that keyword.

System Action: If control statements are being provided by the SYSIN data set, the function requested by the control statement in error will not be executed. Further action depends on the type of data set being processed:

1. If a dump data set is being processed, AMDPRDMP will sntax-check the remaining control statements for that dump without executing them.

2. If an external trace data set is being processed, AMDPRDMP will resume processing with the next control statement.

If user control statements are being entered from the primary system console, AMDPRDMP issues message AMD210D to allow the operator to enter EDIT keywords.

Operator Response: If the control statements are being entered from the system console, reenter the EDIT keywords, making sure that the error mentioned above has been corrected.

Programmer Response: Probable user error. If the control statements are being provided by the SYSIN data set, rerun the job, making sure that all parameter values conform to length requirements.

Problem Determination: Table I, items 2, 4, 29.

AMD205I **DUPLICATE KEYWORD-keywd**

Explanation: While scanning a control statement, EDIT encountered the EXIT or DDNAME keyword after it had already been specified with a different value.

System Action: If control statements are being provided by the SYSIN data set, the function requested by the control statement in error will not be executed.

Further action depends on the type of data set being processed:

1. If a dump data set is being processed, AMDPRDMP will syntax-check the remaining control statements for that dump without executing them.

2. If an external trace data set is being processed, AMDPRDMP will resume processing with the next control statement.

If user control statements are being entered from the primary system console, AMDPRDMP issues message AMD210D to allow the operator to enter EDIT keywords.

Operator Response: If the control statements were entered from the system console, reenter the EDIT keywords, making sure that the error mentioned above has been corrected.

Programmer Response: Probable user error. If the control statements are being provided by the SYSIN data set, rerun the job, making sure that the EXIT or DDNAME keyword is specified only once per EDIT control statement.

Problem Determination: Table I, items 2, 4, 29.

AMD206I **EXCESSIVE NO. PARM VALUES FOR KEYWORD keywd**

Explanation: While scanning multiple parameter values associated with keyword (keywd), EDIT has encountered a greater number of unique parameter values than is allowed for this keyword.

System Action: If control statements are being provided by the SYSIN data set, the function requested by the control statement in error will not be executed. Further action depends on the type of data set being processed:

1. If a dump data set is being processed, AMDPRDMP will syntax-check the remaining control statements for that dump without executing them.

2. If an external trace data set is being processed, AMDPRDMP will resume processing with the next control statement.

If user control statements are being entered from the primary system console, AMDPRDMP issues message AMD210D to allow the operator to enter EDIT keywords.

Operator Response: If entering the control statements from the system console, reenter the EDIT keywords, making sure that the error mentioned above has been corrected.

Programmer Response: Probable user error. If the control statements were entered from the SYSIN data set, rerun the job, making sure that the number of unique parameter values does not exceed the maximum number allowed for this keyword.

Problem Determination: Table I, items 2, 4, 29.

AMD207I **INVALID DELIMITER FOR KEYWORD keywd**

Explanation: While scanning values for keyword (keywd) EDIT has encountered either a delimiter in the place of a value or an unexpected type of delimiter.

System Action: If control statements are being provided by the SYSIN data set, the function requested by the control statement in error will not be executed. Further action depends on the type of data set being processed:

1. If a dump data set is being processed, AMDPRDMP will syntax-check the remaining control statements for that dump without executing them.

2. If an external trace data set is being processed, AMDPRDMP will resume processing with the next control statement.

If user control statements are being entered from the primary system console, AMDPRDMP issues message AMD210D to allow the operator to enter EDIT keywords.

Operator Response: If the control statements are being entered from the system console, reenter the EDIT keywords with the proper delimiters.

Programmer Response: Probable user error. If the control statements are being provided by SYSIN data set, check all delimiters and rerun the job.

Problem Determination: Table I, items 2, 4, 29.

AMD208I **START VALUE EXCEEDS STOP VALUE IN STMTS ABOVE**

Explanation: The START parameter value is larger than the STOP parameter value.

System Action: If control statements are being provided by the SYSIN data set, the function requested by the control statement in error will not be executed. Further action depends on the type of data set being processed:

1. If a dump data set is being processed, AMDPRDMP will syntax-check the remaining control statements for that dump without executing them.

2. If an external trace data set is being processed, AMDPRDMP will resume processing with the next control statement.

If user control statements are being entered from the primary system console, AMDPRDMP issues message AMD210D to allow the user to enter EDIT keywords.

Operator Response: If the control statements are being entered from the system console, reenter the EDIT keywords, making sure that the error mentioned above has been corrected.

Programmer Response: Probable user error. If the control statements are being provided by the SYSIN data set, rerun the job, ensuring that the STOP parameter value is greater than the START parameter value.

Problem Determination: Table I, items 2, 4, 29.

AMD209I INVALID USER EID OR RANGE

Explanation: While scanning the parameter values associated with the USR keyword, EDIT has encountered one of the following conditions:

- An invalid symbolic EID.
- An EID range in which the left (lower) value exceeds the right (upper) value.

System Action: If control statements are being provided by the SYSIN data set, the function requested by the control statement in error will not be executed. Further action depends on the type of data set being processed:

1. If a dump data set is being processed, AMDPRDMP will syntax-check the remaining control statements for that dump without executing them.
2. If an external trace data set is being processed, AMDPRDMP will resume processing with the next control statement.

If user control statements are being entered from the primary system console, AMDPRDMP issues message AMD210D to allow the operator to enter EDIT keywords.

Operator Response: If entering the control statements from the system console, reenter the EDIT keywords, making sure that all of the errors mentioned above have been corrected.

Programmer Response: Probable user error. If the control statements were entered by way of the SYSIN data set, rerun the job, making sure that any symbolic EIDs used are valid and that the right (upper) value in an EID range is equal to or greater than the left (lower) value.

Problem Determination: Table I, items 2, 4, 29.

AMD211I EDIT OPTIONS IN EFFECT - option list

Explanation: This message is issued to inform the user what keyword options have been accepted for EDIT's data reduction process.

If more than one option has been accepted, they are separated by commas in the order specified below:

EXIT = exitname
DDNAME = ddname
START = (day,hh.mm.ss)
STOP = (day,hh.mm.ss)
JOBNAMES = (jjj,jjj,...jjj)
ASCB = (ascbaddr,ascbaddr,...ascbaddr)
either of the following:
IO = SIO = parm
or
SIO = parm,IO = parm
SVC = parm
PI = parm
USR = parm
EXT
RNIO
SRM
RR
DSP

In the options listed, parm may be either ALL, or SEL, where ALL indicates that all events within that event class will be edited, and SEL indicates that only events selected by the user supplying the keyword parameter values for that event class will be edited.

System Action: EDIT processing continues with the above data reduction options in effect.

Operator Response: None.

Programmer Response: None.

AMD212I RCD ON PG nnnnnnnn. RETCD rc RCVD FROM MODULE mod

Explanation: The record currently being processed by EDIT has been dumped in hexadecimal on page number nnnn of the output data set. Module (mod) attempted to format that record, but EDIT was unable to process it because module (mod) returned an invalid return code (rc).

System Action: EDIT will display, in hexadecimal, the record associated with the error.

EDIT takes action based on the value of the "ER" parameter, specified in the EXEC statement of the AMDPRDMP JCL, as follows:

0 If the error was in a format appendage module, as soon as EDIT recognizes that a subsequent record requires that module, processing will terminate for that record and EDIT will select another. If the error was in a user exit module, records will continue to be processed by the format appendages. Message AMD213I will then be issued by EDIT.

1 If the error was in a format appendage module, all subsequent records requiring the same format module will be dumped in hexadecimal format. Message AMD215I will then be issued by EDIT. If the error was in a user exit module, formatting of records will continue. Message AMD213I will then be issued by EDIT.

2 Processing of the current EDIT function will terminate. Message AMD214I is issued by EDIT, and processing will continue with the next control statement.

If ER= is not specified on the EXEC statement, a value of ER= 2 will be assumed.

Operator Response: Report this message to the programmer.

Programmer Response: Probable user error if the module name is either:

1. AMDUSRxx where xx is a hexadecimal number in the range 01-50.

2. A user exit name.

Verify that the module sets a valid return code, and correct it, if necessary.

Problem Determination: If the module name is neither case, (1) nor (2), see Table I, items 1, 2, 4, 29. Make sure that a SYSPRINT DD statement is included in the AMDPRDMP JCL. Save a listing of the SYSPRINT data set and the GTF input trace data set or the dump data set being processed.

AMD213I PROCESSING CONTINUES - BYPASSING MODULE mod

Explanation: This message is issued following message AMD212I and message AMD216I if the user has specified "0" as the value of the ER parameter on the AMDPRDMP EXEC statement. EDIT continues processing, bypassing format appendage or user exit module (mod).

System Action: EDIT execution continues. If the error noted in message AMD212I or AMD216I occurred in a format appendage module, further records requiring that module will not be processed. If the error noted in message AMD212I or AMD216I occurred in a user exit module, processing of records will continue without passing control to the failing user exit.

Operator Response: None.

Programmer Response: None.

AMD214I CURRENT EDIT FUNCTION TERMINATED

Explanation: This message is issued, during EDIT execution, when one of the following occurs:

1. A user exit module could not be found or loaded for execution.

2. A format appendage module, required for processing an external data set, existed in the correct library but could not be loaded for execution.

3. During an attempt to load a user exit module or a format appendage module, an I/O error occurred during execution of a BLDL macro instruction.

4. EDIT attempted to acquire virtual storage space for the load of a module needed to process the current input record by deleting a previously loaded user exit or format appendage module which is no longer in use. EDIT found, however, that the previously loaded format appendage or user exit module had already been deleted.

5. The user has specified '2' or '3' as the value of the ER= parameter on the AMDPRDMP EXEC statement, and an error, identified by message AMD212I, and message AMD216I occurs. When the user has specified 1 as the value of the ER= parameter on the AMDPRDMP EXEC statement, EDIT continues processing, dumping in hexadecimal any record that requires format appendage module (mod) for editing.

System Action: The current EDIT function terminates. Processing continues with the next control statement.

Operator Response: None.

Programmer Response: None.

AMD215I FURTHER RCDS REQUIRING mod WILL BE DUMPED IN HEX

Explanation: This message is issued following message AMD212I and message AMD216I when the user has specified 1 as the value of the ER= parameter on the AMDPRDMP EXEC statement. EDIT continues processing, dumping in hexadecimal any record that requires format appendage module (mod) for editing.

System Action: EDIT continues processing, having deleted format appendage module (mod). Any subsequent records requiring mod for editing will be dumped in hexadecimal.

Operator Response: None.

Programmer Response: None.

AMD216I **RCD AND OTHER INFO ON PG nnnnnnnn. PGM CHECK IN MODULE mod**

Explanation: A program check interrupt has occurred during execution of the format appendage or user exit module (mod). The current input record will be dumped in hexadecimal, along with information pertaining to the program check, on page nnnn of the AMDPRDMP data set.

System Action: EDIT will display in hexadecimal the record associated with the error. EDIT continues execution based on the value of the ER parameter specified on the EXEC statement:

0 If the error was in a format appendage module, as soon as EDIT recognizes that a subsequent record requires that module, processing will terminate for that record an EDIT will select another. If the error was in a user exit module, records will continue to be processed by the format appendages. EDIT will then issue message AMD213I.

1 If the error was in a format appendage module, all subsequent records which require processing by the same format module will be dumped in hexadecimal format. Message AMD215I is then issued by EDIT. If the error was in a user exit module, the resultant action is the same as if 'ER=0' had been specified.

2 Processing of the current EDIT function will terminate. Message AMD214I is issued by EDIT. Processing continues with the next control statement.

If ER= is not specified on the EXEC statement, a value of ER=2 is assumed.

Operator Response: Report this message to the programmer.

Programmer Response: Probable user error if the module name is either:

1. AMDUSRxx where xx is a hexadecimal number in the range 01-50.

2. A user exit name.

Verify that the module in error has been thoroughly tested, using the 'ER=3' parameter value on the AMDPRDMP EXEC statement and including a SYSABEND DD statement in the AMDPRDMP JCL if a dump of the module is desired.

Problem Determination: If the module name is neither case (1) or (2), see Table I, items 1, 2, 4, 13, 29.

AMD217I **NO SYS DATA, JOB SELECTION NOT ALLOWED**

Explanation: The EDIT function of AMDPRDMP is being used and the JOBNAME keyword parameter was specified on the EDIT control statement. The trace data set being processed is in the SYSM format; therefore, editing of trace records by specific jobname is not possible.

System Action: If control statements are being provided by the SYSIN data set, EDIT processing terminates and AMDPRDMP execution continues with the next user control statement. If control statements are being entered from the system console, message AMD218D will be issued allowing the operator to decide if EDIT processing is to continue.

Operator Response: Message AMD218D will be issued following message AMD217I. Respond to message AMD218D.

Programmer Response: Probable user error. If control statements are being provided by the SYSIN data set, do not use the JOBNAME parameters of the EDIT control statement. Make sure that a SYSPRINT DD statement is included in the AMDPRDMP JCL. Rerun the job.

If control statements are being entered from the system console, respond to message AMD218D when it is issued.

Problem Determination: Table I, items 1, 3, 4, 29. Save the GTF trace data set or the dump data set being processed.

AMD220I **NO EDIT DD CARD - ddname**

Explanation: The EDIT function of AMDPRDMP is being used. The DD statement specified by the EDIT keyword parameter DDNAME has been omitted from the AMDPRDMP JCL. In the message text, ddname is the name specified by this parameter.

System Action: EDIT processing terminates. AMDPRDMP execution continues with the next control statement.

Operator Response: Report this message to the programmer.

Programmer Response: Probable user error. Supply the necessary DD statement, or correct the DDNAME parameter by specifying the correct ddname. Make sure that a SYSPRINT DD statement has been included in the AMDPRDMP JCL.

Problem Determination: Table I, items 1, 4, 29.

AMD225I to AMD252I

AMD225I **REGION TOO SMALL FOR EDIT BUFFERS**

Explanation: AMDPRDMP's region is too small to contain the trace data set buffers. The amount of storage required for the buffer is three times the BLKSIZE. A BLKSIZE of 4096 bytes is always used.

System Action: EDIT processing terminates. AMDPRDMP execution continues with the next user control statement.

Operator Response: Report this message to the programmer.

Programmer Response: Probable user error. Increase the region size.

Problem Determination: Table I, items 1, 4, 29.

AMD226I **NO RECORDS IN REQUESTED INTERVAL**

Explanation: EDIT did not find any records in the requested interval for one of the following reasons:

- The time interval specified by the START/STOP keywords is within the time interval covered by the trace data set, but GTF did not generate any records during that time.
- The entire trace data set was generated before the START= time indicated by the EDIT control statement.

System Action: EDIT processing terminates. AMDPRDMP execution continues with the next user control statement.

Operator Response: None.

Programmer Response: None.

AMD227I **DATA SET CREATED AFTER STOP TIME**

Explanation: The EDIT function of AMDPRDMP is being used for an external trace data set. The user specified a STOP= value in his EDIT control statement that is earlier than the value of any time stamp record in the data set.

System Action: Current EDIT processing terminates and processing resumes with the next control statement.

Operator Response: Report this message to the programmer.

Programmer Response: Probable user error. Make sure that the STOP= time is within the time-range of this data set (this can be determined by executing AMDPRDMP EDIT with the same data set, specifying the SYS and USR=ALL options, and examining the block time stamps). Rerun the job with the correct STOP= value.

Problem Determination: Table I, items 4, 29.

AMD228I **TRACE INPUT TO EDIT NOT FROM MVS/XA SYSTEM OR AN EMPTY DATA SET**

Explanation: The trace tape mounted as input for AMDPRDMP EDIT was created on an operating system other than MVS/XA or is an empty data set.

System Action: AMDPRDMP EDIT processing terminates.

Programmer Response: Probable user error. Make sure that the output tape was created by AHLGTF or AMDSADMP service aids on an MVS/XA System.

Problem Determination: Table I, items 2, 13, 29.

AMD251I **INPUT FILE DOES NOT CONTAIN A REAL OR VIRTUAL DUMP**

Explanation: AMDPRDMP has determined that the input data set does not contain a real or virtual dump and therefore cannot print storage for a real or virtual request.

System Action: AMDPRDMP execution continues, but virtual and real requests cannot be printed.

Operator Response: Probable user error. If control statements are being entered through the console, the current tape volume should be demounted by entering the NEWDUMP or NEWTAPE control statement. Otherwise, execute AMDPRDMP again, making sure that the correct dump tape is mounted.

Problem Determination: Table I, items 2, 13, 28, 29.

AMD252I **INPUT FILE DOES NOT CONTAIN A REAL DUMP**

Explanation: AMDPRDMP has determined that the input data set does not contain a real storage dump for a PRINT REAL operation.

System Action: AMDPRDMP execution continues without satisfying PRINT REAL requests.

Operator Response: Probable user error. If control statements are being entered by way of the console, the current tape volume should be demounted by entering the NEWDUMP or NEWTAPE control statement. Otherwise, execute AMDPRDMP again, making sure that the correct dump tape is mounted.

Problem Determination: Table I, items 2, 13, 28, 29.

AMD254I **SYSUT1 D.A. FILE NOT DEFINED - EXECUTION TERMINATED**

Explanation: Because there is no TAPE DD statement included in the AMDPRDMP JCL, the dump information is assumed to be on the direct access file, SYSUT1. However, either the SYSUT1 DD is into defined or else the file described by that DD statement is not direct access storage.

System Action: Execution terminates.

Programmer Response: Probable user error. Either supply a TAPE DD statement defining a tape containing dump data sets, or supply a SYSUT1 DD statement describing a direct access data set that contains dump information.

Problem Determination: Table I, items 1, 2, 3, 4, 13, 29.

AMD258I **UNABLE TO SATISFY A REQUEST FOR COMMON DATA FOR A SWAPPED OUT ADDRESS SPACE**

Explanation: Print Dump could not access common data for a swapped out address space because Print Dump could not determine the bounds of the address space's private area.

System Action: Print Dump processing continues but is limited.

Programmer Response: None.

AMD260I **UNABLE TO ACCESS [PAGE|SEGMENT] TABLE AT adr ID=id**

Explanation: An attempt to verify that the segment table can be accessed from dumped storage failed, adr is the 3-byte real storage address used to access the segment table.

id	Meaning
1	The address was specified by the user with the SEGTAB= control statement.
2	The address was found in the dumped storage at location X'31C'.

System Action: Processing continues.

Programmer Response: If id is 1, verify that the address specified in the SEGTAB= control statement is in dumped real storage. If id is 2, verify that the address found at location X'31C' is the address of an area in dumped real storage.

Problem Determination: Table I, items 1, 2, 3, 4, 29.

AMD261I **UNABLE TO ACCESS CVT**

Explanation: AMDPRDMP was unable to locate the CVT in the dump data set. If input is an SVC dump, or a DSS dump then selected portions of storage were dumped that did not include the CVT. If input was created by AMDSADMP, then there was a probable I/O error either creating or reading the data set.

System Action: Processing continues for as many verbs as possible. Error messages will be put out for any verbs that cannot be processed.

Programmer Response: None.

AMD263I **LEVEL OF DUMP DOES NOT MATCH THE VERSION OF PRINT DUMP**

Explanation: AMDPRDMP can only successfully process dumps from MVS/XA. Checks are made in the dumped system's CVT option field (CVTDCB) for the correct bit configurations for an MVS/XA system. One of these checks failed indicating that the dump might not be from an MVS/XA system. AMDPRDMP only processes dumps taken by MVS/XA dump programs that are at the same release level as AMDPRDMP.

System Action: Processing terminates.

Programmer Response: Probable user error. Verify that the dump was taken on an MVS/XA system. Dumps created from other systems or from Release 1 or 2 of OS/VS2 must be processed by the version of PRDMP for that system.

Problem Determination: Table I, items 1, 2, 3, 4, 29.

AMD264I **PAGE TABLE FAILED VALIDITY CHECK - PROCESSING TERMINATED ID=id**

Explanation: An attempt to verify that the segment table (and hence page tables) is valid has failed. The id indicates where the segment table address was obtained.

id	Meaning
1	The address was specified by the user with the SEGTAB= control statement.
2	The address was found in the dumped storage in the field FLCCVT2 of the PSA.

System Action: Processing continues.

Programmer Response: If id is 2, the SEGTAB= control statement can be used to specify the valid segment table address.

Problem Determination: Table I, items 1, 2, 3, 4, 29.

AMD267I to AMD276I

AMD267I **FOLLOWING INVALID OPERAND ENCOUNTERED: opr**

Explanation: Print Dump encountered invalid operand opr on the FORMAT statement.

System Action: Print Dump continues checking the syntax of operands.

Programmer Response: None.

AMD268I **PROCESSING CONTINUING WITH NEXT OPERAND IF SPECIFIED**

Explanation: Print Dump continues to syntax check the remaining operands, if any, after finding an invalid operand on a FORMAT statement.

System Action: Print Dump continues checking the syntax of operands.

Programmer Response: Correct the invalid operand and reenter the FORMAT statement.

AMD269I **OPERAND SYNTAX ERROR**

Explanation: Print Dump encountered an error while checking the syntax of operands included with either the ASID or JOBNAME verb.

System Action: Print Dump continues checking the syntax of verbs.

Programmer Response: Correct the invalid operands for the ASID or JOBNAME verb and reenter the FORMAT statement. You need not reenter valid operands.

AMD270I **INVALID RANGE SPECIFIED: aaaa-bbbb**

Explanation: The ASID verb on a FORMAT statement lists invalid range aaaa-bbbb.

System Action: Print Dump ignores the invalid ASID range request but continues checking the syntax of other operands.

Programmer Response: Correct the invalid ASID range and reenter the FORMAT statement.

AMD271I **PROCESSING TERMINATED FOR THE RANGE SPECIFICATION OF ASIDS**

Explanation: Print Dump encountered an invalid range for the ASID verb on a FORMAT statement.

System Action: Print Dump ignores the invalid ASID range request but continues checking the syntax of other operands.

Programmer Response: Reenter the FORMAT statement with a valid ASID range. You need not reenter valid operands.

AMD272I **PROCESSING CONTINUING FOR OTHER ASIDS, IF SPECIFIED**

Explanation: Print Dump encountered an invalid ASID value. If other ASID values are specified, processing continues.

System Action: Print Dump continues checking the syntax of the remaining operands.

Programmer Response: Reenter the FORMAT statement with valid ASID values.

AMD273I **INVALID ASID SPECIFIED: aaaa**

Explanation: Print Dump encountered invalid ASID value aaaa on a FORMAT statement.

System Action: Print Dump continues checking the syntax of the remaining operands.

Programmer Response: Reenter the FORMAT statement with valid ASID values.

AMD274I **FOLLOWING DUPLICATE ASID REQUESTED WITH THE ASID OPERAND: aaaa**

Explanation: Duplicate ASID values are specified for the ASID operand on the FORMAT statement.

System Action: Print Dump ignores the duplicate request and processes ASID aaaa only once.

Programmer Response: None.

AMD275I **FOLLOWING DUPLICATE JOBNAME REQUESTED: jjj**

Explanation: Duplicate jobnames are specified for the jobname operand on the FORMAT statement.

System Action: Print Dump ignores the duplicate request and processes JOBNAME jjj only once.

Programmer Response: None.

AMD276I **FOLLOWING INVALID REQUESTED JOBNAME IGNORED: jjj**

Explanation: Print Dump encountered invalid JOBNAME jjj on for the JOBNAME operand on a FORMAT statement.

System Action: Print Dump ignores invalid JOBNAME jjj and continues checking the syntax of other operands.

Programmer Response: Reenter the format statement with a valid JOBNAME. You need not reenter valid operands.

AMD277I **FOLLOWING EXTRANEOUS INFORMATION ON INPUT RECORD IGNORED: x-info**

Explanation: Print Dump found extraneous information, x-info, after a valid operand or delimiter on a FORMAT statement.

System Action: Print Dump ignores the extraneous information and continues processing.

Programmer Response: Verify input.

AMD278I **SEGMENT TABLE FAILED VALIDITY CHECK - PROCESSING TERMINATED ID=id**

Explanation: The bit settings in the segment table flag bytes indicate that the segment table for the dumped system is invalid. ID indicates where the segment table address was obtained.

id	Meaning
1	The address was specified by the user with the SEGTAB= control statement.
2	The address was found in the dumped storage in the field FLCCVT2 of the PSA.

System Action: Processing continues.

Programmer Response: If id is 2, the SEGTAB= control statement can be used to specify the valid segment table address.

Problem Determination: Table I, items 1, 2, 3, 4, 29.

AMD280I **INSUFFICIENT SYSUT1 SPACE PROCESSING CONTINUES**

Explanation: The data set corresponding to the SYSUT1 DD statement does not have enough space allocated to it or space was not available to be allocated. A B37, D37, or E37 ABEND was intercepted which is explained by a write-to-programmer message issued by the system.

System Action: Processing continues. When dump data is requested which was successfully written to SYSUT1 it is obtained from the SYSUT1 data set. However, when dump data is requested which could not be written to SYSUT1, it is obtained from the original input tape.

Programmer Response: If the job needs to be rerun for any other reason, more space may be provided for the SYSUT1 DD statement.

Problem Determination: Table I, items 1, 4, 25a, 29.

AMD281I **AMDPRDMP EXIT INTERFACE INOPERATIVE mod, cde [-rc]**

Explanation: AMDPRDMP was unable to load a required exit interface module (mod) and consequently, the exit function of AMDPRDMP is inoperative. The cde is the ABEND completion code return from this LOAD. The return code, (-rc), from this LOAD is also returned.

System Action: AMDPRDMP continues processing without the exit interface. Exit verbs will be ignored as indicated in message AMD289I. Exit modules defined in AMDPRECT will not obtain control on the various AMDPRDMP exits.

Operator Response: Notify the system programmer of this message.

Programmer Response: Follow the directions for the system completion code (cde) in *System Codes*.

AMD282I **SOME ASIDS COULD NOT BE FOUND**

Explanation: Not all of the ASIDs specified on the control statement could be processed. The omitted ASIDs were either not active or the storage associated with them was not dumped.

System Action: The specified ASIDs which were found are processed.

Operator Response: None.

AMD284I **VIRTUAL PREFIX VALUE NOT AVAILABLE**

Explanation: The value of the PSA prefix register could not be found in the dump data set. This value is used for accessing real locations 0-4K when an AMDSADMP is taken.

System Action: AMDPRDMP continues processing, but no prefixing is done. When information is requested from the real block 0-4K, the information that was in the absolute block 0-4K at the time of the dump is used. This may or may not be the correct information.

Programmer Response: None.

AMD285I **DUMP DEFINED ON DIRECT ACCESS-SYSUT1 IGNORED**

Explanation: A SYSUT1 file was provided but AMDPRDMP determined that the dump data set was already defined on a direct access device.

System Action: AMDPRDMP will process the dump on the direct access device defined by TAPE (or anyname) DD statement without loading the SYSUT1 work file.

Programmer Response: None.

AMD286I to AMD290I

AMD286I **CVT AT xxxxxxxx NOT VALID ID=n**

Explanation: AMDPRDMP was unable to verify that the CVT address (xxxxxxxx) did not pass the validity checks:

- n=1 - The CVT address (xxxxxxxx) was supplied by the user.
- n=2 - The CVT address (xxxxxxxx) was supplied in a DSS or SVC DMP header.
- n=3 - The CVT address (xxxxxxxx) was at loc X'4C' in the dump.

System Action: AMDPRDMP attempts to process the dump data set using the CVT address specified by the user. It is possible that just part of the CVT is invalid and that some parts of it are valid.

Programmer Response: Verify that the address specified in the CVT= control statement really points to the CVT.

AMD287I **UNABLE TO LOAD EXIT MODULE - mod,cde [-rc]**

Explanation: The exit control table (AMDPRECT) indicated that module (mod) should be loaded to process a AMDPRDMP exit, but AMDPRDMP was unable to load the exit module. In the message text, cde is the ABEND completion code returned from this LOAD. The return code (-rc) from this LOAD is also returned.

System Action: Processing continues. AMDPRDMP will not attempt to invoke this exit module for AMDPRDMP exits during the remainder of this execution. If the exit has its own verb, however, AMDPRDMP will attempt to invoke the exit routine each time the verb is encountered.

Operator Response: Notify the system programmer of this message.

Programmer Response: Follow the directions for system completion code (cde) in *System Codes*.

AMD288I **NO OUTPUT PRODUCED BY verb**

Explanation: A AMDPRDMP exit module was processing in conjunction with its own verb (verb) and produced no output to the PRINTER data set.

System Action: Processing continues with the next control statement.

Operator Response: None.

Programmer Response: If the module that processes verb is an IBM-supplied AMDPRDMP exit module, this message indicates a probable system error. Make sure that the exit verb and its associated operands, if any, were specified correctly. If the module is a user-supplied AMDPRDMP exit module, this message indicates a probable user error, unless the exit routine intentionally produces no output under certain conditions. Make sure that the exit routine is coded correctly if some output is expected under all conditions.

Problem Determination: Table I, items 1, 2, 3, 13, 22, 29.

AMD289I **VERB IGNORED - verb**

Explanation: The AMDPRDMP exit interface function is inactive as indicated by message AMD281I. AMDPRDMP determined that verb was probably an exit verb defined in AMDPRECT, but the inoperative exit interface could not determine if it was a valid exit verb.

System Action: The verb is ignored. AMDPRDMP processing continues normally for all verbs not defined in AMDPRECT.

Programmer Response: Correct the problem that caused the exit interface to be inactive, as specified by message AMD281I.

AMD290I **ERROR IN VIRTUAL PREFIX VALUE REAL = yyyyyyyy, VIRTUAL = xxxxxxxx**

Explanation: The values that AMDPRDMP found for the PSA prefix register failed the validity checks of being non-zero and on a 4K boundary. These values are used for accessing logical locations 0-4K when an AMDSADMP dump is taken on a multiprocessing system. Particular circumstances in which this message is issued are:

- One address was zero but the other was not. (Both addresses equal zero is an indication of a dump from a uniprocessing system.)
- At least one address was not on a 4K boundary.
- One of the addresses could not be read from the dump. In this case, the address that could not be read is set to zero in the message.

The addresses in the message are the addresses that were found either in the PCCA for the IPLed CPU in the dumped system or in the CPU status record (real address only).

System Action: AMDPRDMP continues processing after substituting a zero for the invalid address. Prefixing is not done when this happens.

Programmer Response: None.

AMD291I PERMANENT I/O ERROR ON SYSUT1

Explanation: An unrecoverable I/O error occurred while writing the SYSUT1 work file.

System Action: Processing continues. When dump data, which was successfully written to SYSUT1, is requested, it is obtained from the SYSUT1 data set. However, when dump data, which could not be written to SYSUT1, is requested, it is obtained from the original input tape.

Programmer Response: None.

Problem Determination: Table I, items 1, 2, 13, 18, 29.

AMD292I ERROR IN ASCB DISPATCHING CHAIN

Explanation: One of the following errors occurred while AMDPRDMP was reading the ASCB dispatching queue during dump initialization:

- The block of dump data containing one or more of the following pointers was not found in the dump data set:
  - The pointer in the CVT to the head ASCB on the dispatching chain.
  - The pointer in the CVT to the last ASCB on the dispatching chain.
  - The forward or backward chain pointer in any of the ASCBs on the dispatching chain.
- An I/O error occurred while reading the block of dump data that contained one or more of the preceding pointers.

System Action: Processing continues. If any part of the dispatching chain was read, it is saved for later use by the read control module for translating virtual address requests to real addresses when required. Only those virtual read requests whose ASCBs were read will be satisfied. If no part of the dispatching chain could be read, then only requests for virtual storage from common memory will be satisfied.

Operator Response: None.

Programmer Response: None.

AMD293I NUMBER OF ASCBs EXCEEDED DEFAULT LIMIT OF xxx

Explanation: AMDPRDMP found more than xxx ASCBs on the ASCB dispatching chain during dump initialization. This limit is used by AMDPRDMP to avoid a possible looping condition. It is also used in the determination of the ratio of buffer space to workspace required to format address spaces in excess of 200 in a dumped system.

System Action: Processing continues. Only the first xxx ASCBs are saved for later use by AMDPRDMPs read control module for translating virtual addresses to real addresses when required.

Operator Response: None.

Programmer Response: If the number of address spaces in the dumped system is greater than xxx, increase the limit to contain the actual number. This limit is initially established at 200 but can be changed by the user if too small. Care should be taken not to change this limit unless the dumped system actually contains more than 200 address spaces.

Problem Determination: Table I, items 1, 2, 3, 13, 16, 29.

AMD294I UNABLE TO ACCESS SEGMENT TABLE ADDRESS AT PSA LOCATION PSASTOR

Explanation: The user did not include a SEGTAB= verb on a PRDMP control statement, so PRDMP attempted to get the address of the segment table from PSA location PSASTOR. in the dumped system. However, the read for this location failed, so no segment table address is available.

System Action: PRDMP attempts further processing. Since no segment table address could be found, no translation can take place. Hence, only requests for a real storage print will be successful.

Operator Response: None.

Programmer Response: Rerun the PRDMP job being sure to include a SEGTAB= verb as the first PRDMP verb.

AMD295I NONE OF THE SPECIFIED ASIDS COULD BE FOUND

Explanation: None of the ASIDs specified on the control statement could be processed. The ASIDs were either not active or the storage associated with them was not dumped.

System Action: AMDPRDMP processing continues with the next control statement.

AMD296I UNABLE TO OBTAIN SUFFICIENT STORAGE

Explanation: While processing a PRINT STORAGE request, AMDPRDMP attempted to GETMAIN storage for a work area and the GETMAIN failed.

System Action: AMDPRDMP processing continues, but PRINT STORAGE requests are not processed.

Programmer Response: Increase region size.

Problem Determination: Table I, items 1, 2, 3, 4, 29.

AMD297I to AMD303I

AMD297I **CONTROL NOT PASSED TO EXIT MODULE mod - ATTACHED FAILED**

Explanation: AMDPRDMP could not attach user exit module mod.

System Action: Processing continues.

Programmer Response: None.

Problem Determination: Table I, item 29.

AMD298I **DUMP DATA SET READ FACILITY LIMITED DUE TO THE INABILITY TO DETERMINE PRIVATE BOUNDS**

Explanation: Print Dump could not gather information related to the boundaries that describe the various areas of the storage layout.

System Action: Print Dump continues processing, but the processing is limited because of the lack of information.

Programmer Response: None.

AMD299I **UNABLE TO ACCESS GDA**

Explanation: Print Dump is virtually ineffective at this time because it cannot access the GDA (global data area). Without the GDA, Print Dump cannot format related information.

System Action: Print Dump continues processing, but the processing is limited because the GDA could not be accessed.

Programmer Response: None.

AMD300I **EXIT MODULE mod ABNORMALLY TERMINATED xcde**

Explanation: x is U for user ABEND, s for system ABEND. cde is the ABEND code.

AMDPRDMP attached user exit module mod which abnormally terminated.

System Action: Processing continues with the next control statement.

Operator Response: None.

Programmer Response: If module mod is an IBM-supplied AMDPRDMP exit module, this message indicates a probable system error. Make sure that the exit verb and its associated operands, if any, are specified correctly. If module mod is a user-supplied AMDPRDMP exit module, this message indicates a probable user error.

Problem Determination: Table I, items 1, 2, 3, 13, 22, 29.

AMD301I **INVALID REQUEST FOR DUMP DATA DURING PROCESSING OF USER EXIT MODULE mod**

Explanation: A user exit module called the storage access service routine which found more than one bit on in the parameter list field, ADPLPRDP. This is an invalid parameter list request.

Programmer Response: If module mod is an IBM-supplied AMDPRDMP exit module, this message indicates a probable system error. Make sure that the exit verb and its associated operands, if any, are specified correctly. If module mod is a user-supplied AMDPRDMP exit module, this message indicates a probable user error.

Problem Determination: Table I, items 1, 2, 3, 13, 22, 29.

AMD302I **WORK AREA FOR DUMP HEADER RECORD COULD NOT BE OBTAINED - HEADER RECORD ACCESS REQUESTS CANNOT BE SATISFIED**

Explanation: PRINT DUMP tried to GETMAIN storage for a work area and the GETMAIN failed.

System Action: Processing continues but the header record is not available for subsequent access requests.

Programmer Response: Increase the region size on the JOB statement.

Problem Determination: Table I, items 1, 2, 3, 4, 29.

AMD303I **- ERROR ACCESSING CVT. TIME VALUE ON ABSTRACT PAGE NOT AVAILABLE**

Explanation: AMDPRDMP was unable to locate the CVT (communications vector table) in the dump data set. If the input is an SVC dump or a DSS dump, selected portions of storage that do not include the CVT were dumped. If input was created by AMDSADMP, there is a probable I/O error when creating or reading the data set.

System Action: Processing continues without time values.

Programmer Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 29.

AMD304I **- STORAGE ACCESS ERROR. PSW DATA ON ABSTRACT PAGE NOT AVAILABLE**

Explanation: AMDPRDMP is unable to access the data in the dump PSW.

System Action: Processing continues without listing dump PSW data.

Programmer Response: None.

Problem Determination: Table I, items 1, 2, 3, 4, 29.

AMD305I **AMDPRDMP UNABLE TO USE REQUIRED TSO SERVICES**

Explanation: AMDPRDMP requires time sharing option (TSO) services to process SYSIN and SYSPRINT files and to provide parsing services to dump formatting programs. But, TSO is not available.

System Action: AMDPRDMP terminates its execution.

Programmer Response: Make sure that AMDPRDMP has access to TSO services:

- A suitable SYSPRINT file must be available for a message log.
- The TSO terminal monitor program (IKJEFT01) or an equivalent terminal monitor program must have prepared the execution environment for the use of TSO service routines.

Problem Determination: Table I, items 1, 3, 4, 7, 29.

Symptom Record Messages (ASR)

Component Name	ASR
Program Producing Message	Input Output Supervisor (ASR)
Audience and Where Produced	For the user of IPCS: at the terminal and at the IPCS print file for some designated messages.
Message Format	ASRnnnnnt text ASR Component identifier nnnnn Message serial number. t Type code: A Action; the user must perform a specific action. D Decision; the user must choose an alternative. E Error; the user must correct the error before continuing. I Information; no user action is required. text Message text.
Associated and Referenced Publications	• *MVS/XA Data Administration: Macro Instruction Reference*, GC26-4014 • *MVS/XA Integrated Catalog Administration: Access Method Services Reference*, GC26-4019 • *MVS/XA Interactive Problem Control System (IPCS) Planning and Customization*, GC28-1406 • *MVS/XA Interactive Problem Control System (IPCS) User's Guide*, GC28-1407 • *MVS/XA Interactive Problem Control System (IPCS) Command Reference*, GC28-1408 • *MVS/XA Message Library: System Codes*, GC28-1157 • *MVS/XA SPL: Service Aids*, GC28-1159 • *MVS/XA SPL: System Macros and Facilities*, Volumes 1 and 2, GC28-1150, GC28-1151 • *MVS/XA SPL: System Modifications*, GC28-1152 • *MVS/XA TSO Guide to Writing a Terminal Monitor Program (TMP) or a Command Processor*, GC28-1295 • *MVS/XA TSO Terminal Messages*, GC38-1046 • *MVS/XA VSAM Catalog Administration: Access Method Services Reference*, GC26-4075

ASR10001I **The dump does not contain a primary symptom string.**

Explanation: The symptom area of the dump header record does not contain a primary symptom string. For SVC dumps and SYSMDUMPS, the Dump Analysis and Elimination (DAE) function generates the primary string. If the primary symptom is not present in an SVC Dump or SYSMDUMP, this generally indicates that an SWDA was not available for DAE symptom extraction. This will be the case for any dumps taken in a non-recovery environment, such as operator requested dumps or dumps requested via SLIP/PER. Stand Alone dumps do not contain a primary symptom string.

System Action: The system does not display any primary symptom information.

Programmer Response: None.

ASR10002I to ASR10003I

ASR10002I **The dump does not contain a secondary symptom string.**

Explanation: The symptom area of the dump header record does not contain a secondary symptom string. Dump analysis exits that execute under IPCS generate the secondary symptom string.

System Action: The system does not display any secondary symptom information.

Programmer Response: You can create secondary string information by executing in IPCS:

- the BLSCBSVC CLIST for SVC dumps.
- the BLSCBSAD CLIST for Stand Alone dumps.

You can then re-issue the SYMPTOM subcommand to view the secondary symptoms.

ASR10003I **Symptom information could not be accessed for this dump.**

Explanation: The dump header record could not be retrieved from the dump data set.

System Action: The system does not display any symptom information.

Programmer Response: None.

Interactive Problem Control System (IPCS) Messages (BLS)

Component Name	BLS
Program Producing Message	Interactive Problem Control System (IPCS)
Audience and Where Produced	For the user of IPCS: at the terminal and at the IPCS print file for some designated messages.
Message Format	BLSnnnnnt text BLS Component identifier nnnnn Message serial number. t Type code: A Action; the user must perform a specific action. D Decision; the user must choose an alternative. E Error; the user must correct the error before continuing. I Information; no user action is required. text Message text.
Comments	For an explanation of the IPCS User Completion ABEND Codes see *MVS/XA IPCS Logic and Diagnosis*, LY28-1298.
Associated and Referenced Publications	• *MVS/XA Data Administration: Macro Instruction Reference*, GC26-4014 • *MVS/XA Integrated Catalog Administration: Access Method Services Reference*, GC26-4019 • *MVS/XA Interactive Problem Control System (IPCS) Planning and Customization*, GC28-1406 • *MVS/XA Interactive Problem Control System (IPCS) User's Guide*, GC28-1407 • *MVS/XA Interactive Problem Control System (IPCS) Command Reference*, GC28-1408 • *MVS/XA Message Library: System Codes*, GC28-1157 • *MVS/XA SPL: Service Aids*, GC28-1159 • *MVS/XA SPL: System Macros and Facilities*, Volumes 1 and 2, GC28-1150, GC28-1151 • *MVS/XA SPL: System Modifications*, GC28-1152 • *MVS/XA TSO Guide to Writing a Terminal Monitor Program (TMP) or a Command Processor*, GC28-1295 • *MVS/XA TSO Terminal Messages*, GC38-1046 • *MVS/XA VSAM Catalog Administration: Access Method Services Reference*, GC26-4075

BLS01 Messages

BLS01000I **Contention data initialization is in progress**

Explanation: Resource managers are examining the dumped data in order to identify current resource contention at the time of the dump.

System Action: The system continues processing.

Programmer Response: If processing of the command takes longer than you want to wait, you can attention out of the command and run the ANALYZE subcommand in the background.

Problem Determination: Table I, items 5, 16, 29.

BLS01001I **Contention analysis will be incomplete. PCQE passed by exit *xxxxxxxx* is not valid.** *reason*

Explanation: The Contention Queue Element (CQE) create service was called with a parameter list (mapped by BLSAPCQE) that failed a validity check. *xxxxxxxx* passed the bad PCQE. *reason* may be one of the following error conditions:

No PCQE acronym.
The passed parameter list did not contain the PCQE identifier. Verify that the exit has generated a good CQE create parameter list and that it is initialized properly. This error condition usually occurs if the wrong storage was passed.

No resource name provided.
The address of the resource name or the length of the resource name is zero. These two fields must be non-zero.

Resource name exceeds 2800 bytes.
The caller specified a resource name length greater than the maximum size allowed (2800).

Additional data exceeds 2600 bytes.
The caller specified the additional data length greater than the maximum size allowed (2800).

Control block is not a STRUCTURE in virtual storage.
The data description for the control block that represents a unit of work must be identified as a STRUCTURE in virtual storage.

Inconsistent data description.
The caller specified an inconsistent or invalid description for the unit of work which owns or is waiting for the resource. The control block must be identified as a STRUCTURE in virtual storage in a specified ASID.

System Action: The system terminates the CQE create request.

Programmer Response: Correct the code that calls the CQE create service so that it specifies a valid request.

Problem Determination: Try using the TRAPON function to view the parameter list on entry to the CQE create service.

BLS01002I **No resource contention detected. Undetected contention is possible.**

Explanation: The ANALYZE exits produced data with no identifiable resource contention. However, Contention may still exist for resources which the ANALYZE exits have not yet examined.

System Action: The system does not produce a contention report and the subcommand terminates.

Programmer Response: Look at the full ANALYZE report by ASID or RESOURCE. Contention may exist but the system could not identify it from the data it had received.

If this occurs on a virtual dump, the ANALYZE subcommand should not be used due to changing storage during the dumping process. If this occurs on a SADMP and the system did not issue any error messages, then ANALYZE did not detect any contention. However ANALYZE exits do not exist for all possible resources, and some form of contention might still be the cause of the problem.

Problem Determination: If the user believes that contention should have been reported, see Table I, items 5, 16, 29.

BLS01003I **No resources meet the EXCEPTION criteria**

Explanation: The EXCEPTION criteria is that a unit of work owns a resource in contention and is not waiting for another resource. No resources or units of work met this criteria.

System Action: None

Programmer Response: If this condition occurs, run ANALYZE ALL to see all of the resource contention data. It is still possible that unidentified contention exists.

Problem Determination: If the user believes that the analysis is incorrect, see Table I, items 5, 16, 29.

BLS01004I **ANALYZE exit list in PARMLIB member BLSCECT has changed. Correct BLSCECT member or issue DROPDUMP RECORDS(TRANSLATION).**

Explanation: The list of ANALYZE exits specified in PARMLIB member BLSCECT has changed since the last invocation of the ANALYZE or STATUS subcommand. The contention information in the dump directory is inconsistent with the current exit list.

System Action: The system terminates subcommand processing.

Programmer Response: You may take either one of the following actions:

- Check which data set is allocated to file IPCSPARM and examine the contents of member BLSCECT. If BLSCECT does not contain the correct ANALYZE exits first, correct BLSCECT, then terminate and restart the IPCS session to pick up the new BLSCECT contents.
- If the BLSCECT member is correct, issue the IPCS subcommand DROPDUMP RECORDS(TRANSLATION) before re-issuing the ANALYZE subcommand.

Problem Determination: If the problem persists, see table I, items 5, 16, 29.

BLS01005I **No resource Lockouts were detected for this dump.**

Explanation: During processing of the ANALYZE command, no circular resource contention chains were found.

System Action: The system continues subcommand processing.

Programmer Response: None.

Problem Determination: None.

BLS01006I **No ANALYZE exits are defined in PARMLIB member BLSCECT**

Explanation: While processing the ANALYZE subcommand, the system determined that none of the exits defined in SYS1.PARMLIB(BLSCECT) had the ANALYZE option specified.

System Action: The system terminates subcommand processing.

Programmer Response: FILE(IPCSPARM) is used to allocate SYS1.PARMLIB. You can override the IPCSPARM allocation with your own data set. Check the IPCSPARM allocation to determine whether a user data set is allocated or whether the allocation has defaulted to SYS1.PARMLIB. Look at the BLSCECT member in the allocated data set to determine if the ANALYZE exits have been properly specified.

Problem Determination: None.

BLS01040I **No errors were detected by the CBSTAT exits**

Explanation: Control block status exits were invoked for the specified control block but no output was produced.

System Action: The system continues processing.

Programmer Response: While none of the exits which perform analysis for the specific components detected any errors, you should be aware of the possibility that an error in control block or unit may still exist.

Problem Determination: None.

BLS01041I **The CBSTAT exits defined in BLSCECT do not process: STRUCTURE(*controlblk*)**

Explanation: There are no CBSTAT exits defined in parmlib member BLSCECT, which recognize the control block *controlblk* being processed.

System Action: The system continues processing.

Programmer Response: Re-issue the subcommand with a recognizable name. Note that the data description must be for a virtual storage address. CBSTAT does not support data-descriptions for parameters such as CPU, ABSOLUTE and HEADER. Also check the allocation of file IPCSPARM to make sure the allocated BLSCECT member contains the correct CBSTAT definitions.

Problem Determination: None.

BLS01042I **CBSTAT output cannot be generated. CBSP passed by exit *exitname* is not valid.** *reason*

Explanation: The CBSTAT service was called with a parameter list (mapped by BLSACBSP) that failed a validity check. *exitname* is the name of the exit that passed the bad CBSP. *reason* may be one of the following error conditions:

No CBSP acronym.
The passed parameter list did not contain the CBSP identifier. Verify that the exit has generated a good CBSTAT parameter list and that it is initialized properly.

STRUCTURE not specified.
The control block must be identified as a STRUCTURE . Field CBSPDTY must contain an "M" to indicate that the data description is for a STRUCTURE.

Inconsistent data description.
The caller specified an inconsistent or invalid description for the control block which is to be analyzed.

System Action: The system terminates the CBSTAT service request.

Programmer Response: Correct the code that calls the CBSTAT service so that it specifies a valid request.

Problem Determination: When running in an IPCS environment, try using the TRAPON function to view the parameter list on entry to the CBSTAT service.

BLS01043I **CBSTAT requires the specification of a STRUCTURE in virtual storage**

Explanation: The Control Block Status (CBSTAT) subcommand was invoked with a data description that does not describe a STRUCTURE in virtual storage. The CBSTAT service and the CBSTAT exits in IPCS support only control blocks in virtual storage.

System Action: The system terminates the CBSTAT subcommand.

Programmer Response: Correct the input parameters specified on the CBSTAT subcommand.

Problem Determination: None

BLS01060I **A symptom was not added to the dump header. The ADSY parameter passed by exit *exitname* is not valid.**

{ **No ADSY acronym.**
No symptom address.
Symptom length equals zero.
Symptom exceeds 15 characters. }

Explanation: The Add Symptom service was called with a parameter list (mapped by BLSADSY) that failed a validity check. The message identifies the exit *exitname* passing the ADSY. In the event that there is data in either ADSYMP2 or ADSYML2, the system also checks these fields to make sure that a second symptom is also valid. The message text also contains the reason for the failure:

No ADSY acronym
The ADSYID field of the passed parameter list did not contain the ADSY identifier.

No symptom address
The ADSYMP (or ADSYMP2) field of the passed parameter list contains a field of zeros.

Symptom length equals zero
The ADSYML or ADSYML2 field has a zero length value.

Symptom exceeds 15 characters
The symptom is limited to 15 characters, including the "/" delimiter. The ADSYML or ADSYML2 field contains more than the limit.

System Action: The system does not process the request to add a symptom.

Programmer Response: If a user exit is generating the symptom, try correcting the symptom that the dump analysis exit routine is passing.

Problem Determination: When running in an IPCS environment, try using the TRAPON function to view the parameter list on entry to the Add Symptom service.

BLS01061I **Symptom generated by exit *exitname* not added: 'keyword/data'. Symptom not valid:**

{ **'/' separator missing**
data missing
character not valid
keyword exceeds 8 characters
keyword missing
duplicate paired symptoms }

Explanation: The Add Symptom service detected a validity check error in the symptom. The symptom is in the form of *keyword/data*. The message identifies the the exit *exitname* passing the ADSY and the symptom. The message text also contains the cause for the failure:

'/' separator missing.
The symptom does not have a '/' separator to indicate the boundary between the keyword and data fields.

data missing.
The symptom does not have any data following the "/" delimiter.

character not valid.
Symptom data may contain only alphanumeric or national characters (@, #, $).

keyword exceeds 8 characters.
The keyword section of the symptom is limited to 8 characters.

keyword missing.
The symptom does not have a keyword preceding the "/" delimiter.

duplicate paired symptoms.
The second symptom is the same as the first symptom in the passed parameter list (mapped by BLSADSY).

System Action: The system does not process the request to add a symptom.

Programmer Response: If a user exit is generating the symptom, you should correct the user exit routine being passed.

Problem Determination: None.

BLS01062I **Unable to add symptom(s): 'keyword/data' 'keyword/data' Insufficient space in the dump header.**

Explanation: The Add Symptom service attempted to write to section 4 of the symptom record in the dump header record. There was insufficient space to satisfy the update request.

System Action: The system terminates processing.

Programmer Response: None.

Problem Determination: It is possible that an exit has produced a large, non-selective, number of symptoms. Although this may be the desired result, it may also indicate a logic error. You can run the SYMPTOM verbexit to try and identify the exit producing the large number of symptoms. Once identified, the dump could be restored from a backup copy that does not have an out of space condition. To get a fresh copy of the symptoms, you can run all exits except the exit in question.

BLS01063I **Unable to add symptom:**

{ target dump not on a DASD device. / dump header record is not available. / unable to establish ESTAE routine. / RACF authorization failure. }

Explanation: The system cannot add the symptom for one of the following reasons:

- The Add Symptom service is available only to dumps on DASD devices.
- The dump header record is not available.
- The system was unable to establish an ESTAE routine
- RACF authorization failed.

System Action: The system terminates processing.

Programmer Response: If you want the current symptoms, move the dump to a DASD device and rerun the subcommand to obtain RACF authorization.

Problem Determination: None.

BLS01064I **Unable to add symptom(s): 'keyword/data' 'keyword/data' Unable to open the dump data set for update.**

Explanation: A request was made for an OPEN for Update, but the request was not satisfied.

System Action: The system terminates processing.

Programmer Response: This situation can occur when a second IPCS user is running an analysis routine that is also adding symptoms to the dump header record. If this is true, run the routine again, and this should resolve the update conflict.

Problem Determination: Table I, items 16 and 29.

BLS01065I **Unable to add symptom(s): 'keyword/data' 'keyword/data' I/O error (reading | writing) the dump header record.**

Explanation: A request was made to perform either a BDAM READ or WRITE update of the dump header record. The request was not satisfied.

System Action: The system terminates processing.

Programmer Response: This occurs as a result of the DECB having been returned from either the READ or the WRITE attempt with flag fields DECCC2 or DECCC3 set to other than a 0 value.

Problem Determination: Table I, items 16 and 29.

BLS01066I **Unable to add symptom(s): 'keyword/data' 'keyword/data' Unexpected error.**

Explanation: The Add Symptom service encountered an abend during processing and its ESTAE routine was entered.

System Action: The system terminates processing.

Programmer Response: None.

Problem Determination: Table I, items 5, 16 and 29.

BLS01067I **Unable to add symptom(s): 'keyword/data' 'keyword/data' Unable to prevent ATTENTION interrupt.**

Explanation: The Add Symptom service issued the STAX macro to prevent the header record from being tied up during ATTENTION processing. The STAX macro gave a non-zero return code.

System Action: The system terminates processing.

Programmer Response: None.

Problem Determination: Table I, items 16 and 29.

BLS03 Messages

BLS03100I **INVALID ALLOCATION MODEL NAME POINTER. POINTER&RBL. = &RBL.0**

Explanation: An internal error; the internal pointer (the address) to the allocation model is invalid and an allocation model name was not supplied.

System Action: The data set is not allocated.

Programmer Response: None.

Problem Determination: Table I, items 5, 16, and 29.

BLS03101I **CANNOT FIND REQUESTED ALLOCATION MODEL. MODEL NAME = modelname**

Explanation: An internal error; the calling routine supplied allocation model name *modelname* that did not match any allocation model in the BLSCAMOD module.

System Action: The data set is not allocated.

Programmer Response: None.

Problem Determination: Table I, items 5, 16, and 29.

BLS03102I **INVALID ALLOCATION MODEL OVERRIDE PARAMETER KEY CODE. MODEL NAME = modelname, KEY CODE = cde**

Explanation: An internal error; allocation override key code cde did not match any key code value recognized by the BLSCABLD module.

System Action: The data set is not allocated.

Programmer Response: None.

Problem Determination: Table I, items 5, 16, and 29.

BLS03103I **INVALID ALLOCATION MODEL OVERRIDE keyname PARAMETER VALUE. MODEL NAME = modelname, keyname VALUE = val**

Explanation: An internal error; a calling module supplied keyname value val that was not recognized by the BLSCABLD module.

System Action: The data set is not allocated.

Programmer Response: None.

Problem Determination: Table I, items 5, 16, and 29.

BLS03104I **INVALID ALLOCATION MODEL OVERRIDE PARAMETER LENGTH. MODEL NAME = modelname, KEY CODE = cde, PARM VALUE = prm, LENGTH = len**

Explanation: An internal error; a calling module supplied an invalid length **len** for the indicated override parameter **prm**.

System Action: The data set is not allocated.

Programmer Response: None.

Problem Determination: Table I, items 5, 16, and 29.

BLS03105I **INVALID ALLOCATION MODEL NAME POINTER (INTERNAL ERROR). MODEL NAME = modelname**

Explanation: An internal error; the internal pointer (the address) to allocation model **modelname** is invalid.

System Action: The data set is not allocated.

Programmer Response: None.

Problem Determination: Table I, items 5, 16, and 29.

BLS03106I **TOO MANY ALLOCATION PARAMETERS. MAX 32. MODULE NAME = mod, MODEL NAME = modelname**

Explanation: An internal error; the allocation parameter list exceeded the internal dimension allowed. Module **mod** issued the message.

System Action: The data set is not allocated.

Programmer Response: None.

Problem Determination: Table I, items 5, 16, and 29.

BLS03107I **ALLOCATION PARAMETER TEXT UNIT SPACE EXCEEDED. MAX 1024 BYTES. MODULE NAME = mod, MODEL NAME = modelname**

Explanation: An internal error; the internal work area assigned to contain allocation text units is depleted. Module **mod** issued the message.

System Action: The data set is not allocated.

Programmer Response: None.

Problem Determination: Table I, items 5, 16, and 29.

BLS03108I **INVALID IDCAMS DEFINE CLUSTER SYSIN STREAM POINTER. MODEL NAME = modelname**

Explanation: An internal error; the internal pointer (the address) to the access method services DEFINE CLUSTER request stream is invalid.

System Action: The data set is not allocated.

Programmer Response: None.

Problem Determination: Table I, items 5, 16, and 29.

BLS03109I **GETMAIN FAILED. MODULE NAME = mod**

Explanation: Enough storage could not be obtained to process a data access services request. Module **mod** detected the message.

System Action: The data access services request terminates. Control returns to the caller with a return code of 12.

Programmer Response: Ensure that the region size is large enough for the execution of an IPCS session. If the problem persists, issue the END subcommand to terminate the session and restart IPCS.

Problem Determination: Table I, items 5, 16, and 29.

BLS03110I **LINK/LOAD SVC TO mod1 FROM mod2 FAILED. ABEND CODE = xxx, REASON CODE = yyy**

Explanation: Module mod2 issued a LINK or LOAD SVC to module mod1. The SVC failed with an ABEND code of **xxx** and a reason code of **yyy**. The module names are respectively:

mod1	**mod2**
BLSCAAMS	IDCAMS
BLSCANAL	IKJEFF18
BLSCANLE	IKJEFF19
BLSCANLI	IKJEFF02
BLSCFAMS	IDCAMS

System Action: The data access services request terminates. Control returns to the caller.

Programmer Response: Ensure that the requested modules are available for use. Respond as if IPCS had ABENDed with a code of **xxx**. See *System Codes* for an explanation of the system completion (ABEND) codes.

Problem Determination: Table I, items 5, 16, and 29.

BLS03111I **mod INVOKED IDCAMS WHICH RETURNED INFO/ERROR MESSAGES. MODEL NAME = modelname**

Explanation: Module **mod** invoked access method services and received messages indicating an error or condition that needs further explanation. The access method services messages are displayed following this message.

System Action: The data access services request might not have been performed, depending on the nature of the BLS03nnnt messages that follow.

Programmer Response: If required to do so, respond to the messages that follow. See message prefix IDC in this manual for an explanation of the access method services messages.

Problem Determination: Table I, items 1, 3, 4, 25a, and 34a.

BLS03112I **DYNALLOC INFO/ERROR MESSAGE. MODEL NAME = modelname, OPCODE = cde, RETCODE = rc, INFO/REASON CODE = yyy, DSN = dsn**

Explanation: Dynamic allocation finished with return code **rc**, and reason or information code **yyy**. The model name used is **modelname**, the operation code to dynamic allocation is **cde**, and the data set name is **dsn**.

The dynamic allocation operation codes are:

```
01    Allocation
02    Unallocation
```

See *System Macros and Facilities* for an explanation of the dynamic allocation return codes and their information/reason codes.

The model is the name of the default allocation parameters block that is contained in the module BLSCAMOD. The model name is copied into the data access services control block during dynamic allocation.

System Action: The data access services request either completes or terminates depending upon the nature of the error. Any messages returned by IKJEFF18 are displayed following this message.

Programmer Response: Take action indicated by the IKJEFF18 messages that follow this message. See *TSO Terminal Messages* for an explanation of the IKJEFF18 messages.

Problem Determination: Table I, items 1, 3, 4, 25a, 29, and 34a.

BLS03113I **INVALID ALLOCATION MODEL TEXT UNIT KEY CODE. MODEL NAME = modelname, KEY CODE = cde**

Explanation: An internal error; allocation model modelname has an invalid allocation text-unit key-code of **cde**.

System Action: The data set was not allocated.

Programmer Response: None.

Problem Determination: Table I, items 5, 16, and 29.

BLS03114I **VSAM DATA ACCESS ERROR. DDNAME = ddn, OPCODE = cde, IPCS RETCODE = rc1, IPCS REJECT CODE = xxx, VSAM RETCODE = rc2, ERROR/FEEDBACK CODE = yyy, DSN = dsn**

Explanation: Data access services detected a VSAM logical or physical I/O error while executing operation code cde for data set name **dsn**, ddname **ddn** IPCS issued return code **rc1** and command reject code **xxx**. VSAM issued return code **rc2** with error/feedback code **yyy**.

Data access services operations, except OPEN and CLOSE, issue command reject reason codes with a return code of 4 (invalid request). The command reject reason codes are:

Reason	Explanation
0	DMCB (data access services control block) is not open.
2	Ineligible for keyed or relative record access, or wrong access type.
3	Buffer length supplied too small.
4	Invalid relative record number.
5	VSAM positioning error.
6	Updated record has changed key.
7	Invalid record length change.
8	Invalid record length.
9	Invalid key length.
11	VSAM MODCB verb failed.
255	Undefined request reject-reason.

System Action: The requested data access services operation was not performed. Additional messages produced by VSAMFAIL might follow this message. These messages give additional information about the error.

Programmer Response: Respond appropriately to the VSAMFAIL messages that follow. See *TSO Terminal Messages* for an explanation of the VSAMFAIL messages.

Problem Determination: Table I, items 1, 3, 4, 25a, 29, and 34a.

BLS03115I **SYNAD ERROR. DSN = dsn, VOLSER = volser, STEP = sss, UADR = ddd, DEVTY = devtyp, DDNAME = ddn, OPN = opr, RRE = err, ACSMTH = QSAM, ADR/BLK = bbbb**

Explanation: QSAM detected a physical I/O error while executing operation **opr** for data set **dsn**, ddname **ddn**. The data set resided on volume serial **vol** that was mounted on unit **ddd** (where ddd is displayed in hexadecimal). Other inserted values for this message that are not described here are obtained from the SYNAD message buffer. For the meanings of these values, see the SYNADAF macro description in *Data Administration: Macro Instruction Reference*.

System Action: The requested operation is not performed.

Programmer Response: Examine the error data, correct the error if possible, and retry the operation.

Problem Determination: Table I, items 1, 3, 4, 25a, 29, and 34a.

BLS03116I **NON-VSAM DATA ACCESS ERROR. DDNAME = ddn, OPCODE = opr, IPCS RETCODE = rc, IPCS REJECT CODE = xxx, ABEND CODE = yyy, ABEND REASON CODE = zzz, DSN = dsn**

Explanation: Data access services detected a logical or physical I/O error while executing operation **opr** for data set name **dsn**, ddname **ddn**. IPCS issued return code **rc** and command reject code **xxx**. If the ABEND code is nonzero, ABEND **yyy** occurred with an ABEND reason code (in general purpose register 15) **zzz**.

Data access services operations, except OPEN and CLOSE, issue command reject reason codes with a return code of 4 (invalid request). The command reject reason codes are:

Reason	Explanation
0	The DMCB (data access services control block) is not open.
1	Invalid request, GET for update not allowed.
2	Ineligible for keyed or relative record access, or wrong access type.
3	Buffer length too small.
8	Invalid record length.
10	Invalid non-VSAM command.

System Action: The requested data access services operation was not performed. Message BLS03115I will follow this message if the data access services return code is X'0C'. This message gives additional information about the error.

Programmer Response: Respond as appropriate to the message that follows.

Problem Determination: Table I, items 1, 3, 4, 25a, 29, and 34a.

BLS03117I **INVALID FREE PARAMETER KEY CODE. MODULE NAME = mod, KEY CODE = cde**

Explanation: An internal error; a data access services request to free a data set contained keyword parameter code **cde** that did not match any key code value recognized by module **mod**.

System Action: If **mod** is BLSCFDYN, the data set may not be freed. If **mod** is BLSCFAMS, the data set is freed but not scratched. The subcommand terminates and the executing module issues an IPCS user completion ABEND code of X'06E' (decimal 0110).

Programmer Response: None.

Problem Determination: Table I, items 5, 16, and 29.

BLS03118I **INVALID FREE keyname PARAMETER VALUE. MODULE NAME = mod, keyname VALUE = val**

Explanation: An internal error; a calling module supplied keyname parameter value **val** that was not recognized by module **mod**.

System Action: The data set may not be freed. The subcommand terminates and the executing module issues an IPCS user completion ABEND code of X'06E' (decimal 0110).

Programmer Response: None.

Problem Determination: Table I, items 5, 16, and 29.

BLS03119I **INVALID FREE prm PARAMETER LENGTH. MODULE NAME = mod, prm = val, LENGTH = len**

Explanation: An internal error; calling module **mod** supplied an invalid length **len** for parameter **prm**, which specified value **val**.

System Action: The data set is not scratched. The subcommand terminates and the executing module issues an IPCS user completion ABEND code of X'06E' (decimal 0110).

Programmer Response: None.

Problem Determination: Table I, items 5, 16, and 29.

BLS04 Messages

BLS04000I **PROBLEM IDENTIFIER prob-id DOES NOT EXIST**

Explanation: Either problem identifier **prob-id** specified with the subcommand or, if omitted, the default problem identifier could not be found in the problem directory.

System Action: The subcommand processing terminates.

Programmer Response: Verify the specified problem identifier or use the SETDEF LIST subcommand to verify the default problem identifier.

Problem Determination: Table I, items 1, 3, 4, 26c (for IPCSPRnn), and 29. Use IDCAMS PRINT to dump the appropriate records from the problem directory; see Table IV for details.

BLS04001I **SPECIFIED PROBLEM IDENTIFIER prob-id DOES NOT EXIST**

Explanation: Problem identifier **prob-id** specified with the subcommand could not be found in the problem directory.

System Action: Except for the LISTPROB subcommand, all subcommands terminate. The LISTPROB processing continues with the next problem identifier or range in the specified problem list.

Programmer Response: Verify the specified problem identifier.

Problem Determination: Table I, items 1, 3, 4, 26c (for IPCSPRnn), and 29. Use IDCAMS PRINT to dump the appropriate records from the problem directory; see Table IV for details.

BLS04002I **DEFAULT PROBLEM IDENTIFIER prob-id DOES NOT EXIST**

Explanation: The subcommand did not specify a problem identifier and the default problem identifier **prob-id** could not be found in the problem directory.

System Action: The subcommand processing terminates.

Programmer Response: Use the SETDEF LIST subcommand to verify the default problem identifier.

Problem Determination: Table I, items 1, 3, 4, 26c (for IPCSPRnn), and 29. Use IDCAMS PRINT to dump the appropriate records from the problem directory; see Table IV for details.

BLS04003I **PROBLEM IDENTIFIER OMITTED AND THERE IS NO DEFAULT IN EFFECT**

Explanation: The subcommand did not specify a problem identifier and the SETDEF subcommand did not specify a default problem identifier.

System Action: The subcommand processing terminates.

Programmer Response: Use the SETDEF LIST subcommand to verify that there is no default in effect. If there is no default in effect specify the problem identifier with the subcommand, or use the SETDEF subcommand to establish a default problem identifier.

BLS04004I **USER IS NOT OWNER OF PROBLEM prob-id**

Explanation: A request by a TSO userid to modify or delete a problem was not satisfied. The TSO userid was either not the owner of problem **prob-id** or not specified on the ADMINAUTHORITY keyword in the IPCSPRnn member of SYS1.PARMLIB.

System Action: The subcommand processing terminates.

Programmer Response: Ensure that the appropriate IPCSPRnn member is being used. If it is the correct IPCSPRnn member, perform the following checks to verify that the TSO userid is, in fact, not authorized to delete or modify the problem.

1. Issue the LISTPROB subcommand. Is the TSO userid the same as prob-id? If it is, see the Problem Determination explanation.

2. Check the IPCSPRnn member of SYS1.PARMLIB. Does the ADMINAUTHORITY parameter specify the TSO userid? If it does, see the Problem Determination explanation.

If both of the preceding checks result in responses of "No", the TSO userid is not authorized to modify or delete the problem.

Problem Determination: Table I, items 1, 3, 4, 26c, (for IPCSPRnn), and 29. Use IDCAMS PRINT to dump the appropriate records from the problem directory; see Table IV for details.

BLS04005I DATA SET NAME dsn NOT KNOWN TO IPCS

Explanation: Data set name **dsn** specified with the subcommand or, if omitted, the default data set name could not be found in the data set directory.

System Action: The subcommand processing terminates.

Programmer Response: Using the LISTDSN subcommand, verify that the data set name has been added to the data set directory. Use the SETDEF LIST subcommand to verify the default data set name.

Problem Determination: Table I, items 1, 3, 4, 26c (for IPCSPRnn), and 29. Use IDCAMS PRINT to dump the appropriate records from the data set directory; see Table IV for details.

BLS04008I DATA SET NAME OMITTED AND THERE IS NO DEFAULT IN EFFECT

Explanation: The subcommand did not specify a data set name and the SETDEF subcommand did not specify a default data set name.

System Action: The subcommand processing terminates.

Programmer Response: Use the SETDEF LIST subcommand to verify that there is no default data set name. If there is no default data set name, specify the data set name with the subcommand or use the SETDEF subcommand to establish a default data set name.

BLS04009I DATA SET dsn IS NOT ASSOCIATED WITH PROBLEM prob-id

Explanation: A MODDSN subcommand requested data set name **dsn**. However, this data set name was not found to be associated with the specified or default problem **prob-id**.

System Action: The subcommand processing terminates.

Programmer Response: Ensure that the data set name is associated with the appropriate problem identifier by using the LISTDSN PROBLEMS or LISTPROB DSNAMES subcommand.

Problem Determination: Table I, items 1, 3, 4, 26c (for IPCSPRnn), and 29. Use IDCAMS PRINT to dump the appropriate records from the problem and data set directories; see Table IV for details.

BLS04010I A MEMBER OF A DATA SET CANNOT BE MANAGED

Explanation: A subcommand attempted to apply the attribute MANAGED to a data set member. If the attribute MANAGED is required, the entire data set must have this attribute. If only the member of the data set is required, it must have the attribute UNMANAGED.

System Action: The management attribute is forced to UNMANAGED and the subcommand processing continues. If this is the only error detected by the subcommand, it will complete with a return code of 8.

Programmer Response: Determine whether to have the entire data set MANAGED or to have the member UNMANAGED. Issue the ADDDSN subcommand if you want the entire data set managed, and DELDSN to dissociate the UNMANAGED member from the problem.

Note: IPCS does not recognize the fact that an UNMANAGED member may exist in a MANAGED data set. When the data set is scratched, UNMANAGED members may still be associated with problems, but the data set will no longer exist.

BLS04011I PROBLEM DESCRIPTION DATA SET NOT FOUND

Explanation: An ADDPROB or MODPROB subcommand used the DSDESCRIPTION keyword to specify the name of a data set containing a problem description. This data set could not be allocated.

System Action: If the subcommand was ADDPROB, no description is added to the problem. If the subcommand was MODPROB, the current description for that problem is deleted.

Programmer Response: Verify the data set name used. Ensure that the data set is cataloged.

BLS04012I ERROR ACCESSING THE PROBLEM DESCRIPTION DATA SET

Explanation: An ADDPROB or MODPROB subcommand used the DSDESCRIPTION keyword to specify the name of a data set containing a problem description. An error occurred while accessing this data set.

System Action: The portion of the problem description that was read from the data set before the error occurred becomes the current problem description. All of the problem description that had been with the problem is deleted.

Programmer Response: Ensure that the data set specified by the DSDESCRIPTION keyword had the appropriate format (RECFM = F, FB, FS, or FBS; and LRECL = 80). Retry the operation using the MODPROB subcommand.

BLS04014I INTERNAL ERROR; FREEMAIN WAS UNABLE TO FREE ALL REQUESTED STORAGE

Explanation: A FREEMAIN macro instruction returned with a nonzero return code. The requested storage could not be released.

System Action: The subcommand terminates and the executing module issues an IPCS user completion ABEND code of X'097' (decimal 0151).

Programmer Response: None.

Problem Determination: Table I, items 5, 16, and 29.

BLS04015I NOT ENOUGH VIRTUAL STORAGE COULD BE OBTAINED

Explanation: A GETMAIN macro instruction returned with a nonzero return code. Enough virtual storage could not be obtained for continued execution.

System Action: The subcommand terminates and the executing module issues an IPCS user completion ABEND code of X'07A' (decimal 0122).

Programmer Response: Ensure that the region size is large enough for the execution of an IPCS session.

Problem Determination: Table I, items 5, 16, and 29.

BLS04016I UNABLE TO OBTAIN STORAGE FOR FPBLOK

Explanation: A GETMAIN macro instruction requested storage for control block FPBLOK. IPCS initialization could not complete because sufficient storage was not available.

System Action: The IPCS session terminates.

Programmer Response: Ensure that the region size is large enough for the execution of an IPCS session.

BLS04017I {FILE(IPCSPARM)|SYS1.PARMLIB} member *mem* not found

Explanation: Member *mem*, specified by the PARM operand on the IPCS command or the IPCS command default, could not be found. IPCS initialization could not complete.

System Action: The IPCS session terminates.

Programmer Response: Specify an existing member, create the specified member, or allocate FILE(IPCSPARM) to a concatenation of libraries, one of which contains the member that you wish to use.

Problem Determination: None.

BLS04018I Error while processing {FILE(IPCSPARM)|'SYS1.PARMLIB'} member *mem*

Explanation: During IPCS initialization, an error occurred while parsing the content of member *mem*. IPCS initialization could not be completed after encountering a nonzero return code.

System Action: The IPCS session terminates.

Programmer Response: Verify the validity of member *mem* specified during IPCS invocation. Determine the type of error that occurred:

- One or more syntax errors in the member, as indicated by associated parse error messages.
- An I/O error accessing the member, as indicated by associated diagnostic messages.
- An internal error associated with the parse interface.

Correct the syntax error(s) and/or conditions causing an I/O error.

Problem Determination: If you suspect an internal error, perform Table I, item 29.

BLS04019I to BLS04045I

BLS04019I **Missing or invalid keyword** *keywd*

Explanation: The required *keywd* is either missing or the value specified is invalid in the active SYS1.PARMLIB member being used for IPCS initialization. IPCS initialization did not complete.

System Action: The IPCS session terminates.

Programmer Response: Supply valid forms of all required keywords in the SYS1.PARMLIB member to be used.

BLS04040I **ERROR ALLOCATING THE PROBLEM DIRECTORY**

Explanation: The problem directory could not be allocated. The messages following this message indicate in more detail the nature of the error.

System Action: If this condition occurs during initialization of the IPCS session, the session terminates. If this condition occurs during execution of a subcommand, the subcommand terminates.

Programmer Response: Ensure that the appropriate IPCSPRnn member is being used. Respond appropriately to the messages that follow.

Problem Determination: Table I, items 1, 3, 4, 25a, 26c (for IPCSPRnn), 29, and 34a. If this message is followed by an IPCS user completion ABEND code, perform Table I, items 5 and 16 in addition.

BLS04041I **ERROR ACCESSING THE PROBLEM DIRECTORY**

Explanation: An error occurred while opening, reading, writing, or closing the problem directory. The messages following this message indicate in more detail the nature of the error.

System Action: The subcommand processing terminates.

Programmer Response: Respond appropriately to the messages that follow.

Problem Determination: Table I, items 1, 3, 4, 25a, 26c (for IPCSPRnn), 29, and 34a. If this message is followed by an IPCS user completion ABEND code, perform Table I, items 5 and 16 in addition.

BLS04042I **ERROR OPENING THE PROBLEM DIRECTORY**

Explanation: The problem directory could not be opened. The messages following this message indicate in more detail the nature of the error.

System Action: The subcommand processing terminates.

Programmer Response: Respond appropriately to the messages that follow.

Problem Determination: Table I, items 1, 3, 4, 25a, 26c (for IPCSPRnn), 29, and 34a. If this message is followed by an IPCS user completion ABEND code, perform Table I, items 5 and 16 in addition.

BLS04043I **ERROR READING THE PROBLEM DIRECTORY**

Explanation: An error occurred while reading the problem directory. The messages following this message indicate in more detail the nature of the error.

System Action: The subcommand processing terminates.

Programmer Response: Respond appropriately to the messages that follow.

Problem Determination: Table I, items 1, 3, 4, 25a, 26c (for IPCSPRnn), 29, and 34a. If this message is followed by an IPCS user completion ABEND code, perform Table I, items 5 and 16 in addition.

BLS04044I **ERROR WRITING THE PROBLEM DIRECTORY**

Explanation: An error occurred while writing into the problem directory. The messages following this message indicate in more detail the nature of the error.

System Action: The subcommand processing terminates.

Programmer Response: Respond appropriately to the messages that follow.

Problem Determination: Table I, items 1, 3, 4, 25a, 26c (for IPCSPRnn), 29, and 34a. If this message is followed by an IPCS user completion ABEND code, perform Table I, items 5 and 16 in addition.

BLS04045I **ERROR CLOSING THE PROBLEM DIRECTORY**

Explanation: An error occurred while closing the problem directory. The messages following this message indicate in more detail the nature of the error.

System Action: The subcommand processing terminates.

Programmer Response: Respond appropriately to the messages that follow.

Problem Determination: Table I, items 1, 3, 4, 25a, 26c (for IPCSPRnn), 29, and 34a. If this message is followed by an IPCS user completion ABEND code, perform Table I, items 5 and 16 in addition.

BLS04046I ERROR FREEING dsn

Explanation: An error occurred while trying to free data set **dsn** during a normal IPCS session termination or an abnormal IPCS subtask termination. The messages following this message indicate in more detail the nature of the error.

System Action: Processing continues to close and free any additional data sets represented on the DMCB (data access services control block) chain.

Programmer Response: Respond appropriately to the messages that follow.

Problem Determination: Table I, items 1, 3, 4, 25a, 26c (for IPCSPRnn), 29, and 34a. If this message is followed by an IPCS user completion ABEND code, perform Table I, items 5 and 16 in addition.

BLS04047I ERROR CLOSING dsn

Explanation: An error occurred while trying to close data set **dsn** during a normal IPCS session termination or an abnormal IPCS subtask termination. The messages that follow this message indicate in more detail the nature of the error.

System Action: Processing continues to close and free any additional data sets represented on the DMCB (data access services control block) chain.

Programmer Response: Respond appropriately to messages that follow.

Problem Determination: Table I, items 1, 3, 4, 25a, 26c (for IPCSPRnn), 29, and 34a. If this message is followed by an IPCS user completion ABEND code, perform Table I, items 5 and 16 in addition.

BLS04050I ERROR ALLOCATING THE DATA SET DIRECTORY

Explanation: An error occurred while allocating the data set directory. The messages that follow this message indicate in more detail the nature of the error.

System Action: If this condition occurs during initialization of the IPCS session, the session terminates. If this condition occurs during execution of a subcommand, the subcommand terminates.

Programmer Response: Ensure that the appropriate IPCSPRnn member is being used.

Problem Determination: Table I, items 1, 3, 4, 25a, 26c (for IPCSPRnn), 29, and 34a. If this message is followed by an IPCS user completion ABEND code, perform Table I, items 5 and 16 in addition.

BLS04051I ERROR ACCESSING THE DATA SET DIRECTORY

Explanation: An error occurred while opening, reading, writing, or closing the data set directory. The messages following this message indicate in more detail the nature of the error.

System Action: The subcommand processing terminates.

Programmer Response: Respond appropriately to messages that follow.

Problem Determination: Table I, items 1, 3, 4, 25a, 26c (for IPCSPRnn), 29, and 34a. If this message is followed by an IPCS user completion ABEND code, perform Table I, items 5 and 16 in addition.

BLS04052I ERROR OPENING THE DATA SET DIRECTORY

Explanation: An error occurred while opening the data set directory. The messages following this message indicate in more detail the nature of the error.

System Action: The subcommand processing terminates.

Programmer Response: Respond appropriately to the messages that follow.

Problem Determination: Table I, items 1, 3, 4, 25a, 26c (for IPCSPRnn), 29, and 34a. If this message is followed by an IPCS user completion ABEND code, perform Table I, items 5 and 16 in addition.

BLS04053I ERROR READING THE DATA SET DIRECTORY

Explanation: An error occurred while reading the data set directory. The messages following this message indicate in more detail the nature of the error.

System Action: The subcommand processing terminates.

Programmer Response: Respond appropriately to messages that follow.

Problem Determination: Table I, items 1, 3, 4, 25a, 26c (for IPCSPRnn), 29, and 34a. If this message is followed by an IPCS user completion ABEND code, perform Table I, items 5 and 16 in addition.

BLS04054I to BLS04063D

BLS04054I ERROR WRITING THE DATA SET DIRECTORY

Explanation: An error occurred while writing to the data set directory. The messages following this message indicate in more detail the nature of the error.

System Action: The subcommand processing terminates.

Programmer Response: Respond appropriately to the messages that follow.

Problem Determination: Table I, items 1, 3, 4, 25a, 26c (for IPCSPRnn), 29, and 34a. If this message is followed by an IPCS user completion ABEND code, perform Table I, items 5 and 16 in addition.

BLS04055I ERROR CLOSING THE DATA SET DIRECTORY

Explanation: An error occurred while closing the data set directory. The messages following this message indicate in more detail the nature of the error.

System Action: The subcommand processing terminates.

Programmer Response: Respond appropriately to the messages that follow.

Problem Determination: Table I, items 1, 3, 4, 25a, 26c (for IPCSPRnn), 29, and 34a. If this message is followed by an IPCS user completion ABEND code, perform Table I, items 5 and 16 in addition.

BLS04056I ERROR FREEING THE DATA SET DIRECTORY

Explanation: An error occurred while freeing the data set directory. The messages following this message indicate in more detail the nature of the error.

System Action: The subcommand processing terminates.

Programmer Response: Respond appropriately to the messages that follow.

Problem Determination: Table I, items 1, 3, 4, 25a, 26c (for IPCSPRnn), 29, and 34a. If this message is followed by an IPCS user completion ABEND code, perform Table I, items 5 and 16 in addition.

BLS04060I THE DEFAULT KEYWORD WAS SPECIFIED BUT NO DATA SET NAME WAS SPECIFIED. THE DEFAULT KEYWORD IS IGNORED

Explanation: The subcommand specified the DEFAULT keyword to change the default data set name, but no data set name was supplied.

System Action: The subcommand processing ignores the DEFAULT keyword and the default data set name remains unchanged.

Programmer Response: Use the SETDEF subcommand to change the default data set name.

BLS04061I INTERNAL ERROR; RECORD PASSED TO BLSFOD00 WAS NOT A BASE RECORD

Explanation: The record passed to module BLSFOD00 for comparison was not a data set directory base record.

System Action: The subcommand terminates and the executing module issues an IPCS user completion ABEND code of X'070' (decimal 0112).

Programmer Response: Reissue the subcommand.

Problem Determination: Table I, items 5, 16, and 29.

BLS04062I DATA SET ATTRIBUTE CONFLICT

Explanation: The subcommand specified the NOCONFIRM keyword or the NOCONFIRM keyword was in effect through the use of a previous SETDEF subcommand. A data set attribute conflict occurred between the attributes specified with the subcommand and the current values in the data set directory.

System Action: If the subcommand issued was ADDDSN, the current values in the data set directory remain unchanged. If the subcommand issued was MODDSN, the new attributes replace the corresponding old attribute values in the data set directory.

Programmer Response: Ensure that the desired system action was taken. Use the MODDSN subcommand to reset any attributes incorrectly altered. Specify SETDEF CONFIRM to receive future confirmation messages.

BLS04063D DATA SET ATTRIBUTE CONFLICT. ENTER Y TO CONTINUE, N TO TERMINATE

Explanation: This message requests the user to indicate whether the subcommand should continue or terminate. A data set attribute conflict occurred between the attributes specified with the subcommand and the current values in the data set directory. One of the following might have caused the conflict:

- The subcommand specified the CONFIRM keyword.

- A previous SETDEF subcommand specified the CONFIRM keyword.
- The subcommand used the default.

The informational messages preceding this message indicate the current attribute values in the data set directory.

System Action: If the user enters Y, the subcommand should continue. If the subcommand issued was:

- ADDDSN, the data set will be associated with the problem, but the current values in the data set directory are left intact.
- MODDSN, the new attributes specified with the subcommand will replace the corresponding old attribute values in the data set directory.

If the user enters N, the subcommand terminates and no changes will be made to the attribute values in the data set directory. If the subcommand issued was ADDDSN, the data set will not be associated with the problem.

Programmer Response: Reply Y to continue or N to terminate the subcommand.

BLS04064I INTERNAL ERROR; MODULE BLSFOD00 FOUND UNMATCHED DSD KEYS

Explanation: The data set directory records passed to module BLSFOD00 for comparison had different keys.

System Action: The subcommand terminates and the executing module issues an IPCS user completion ABEND code of X'06F' decimal (0111).

Programmer Response: Reissue the subcommand.

Problem Determination: Table I, items 5, 16, and 29.

BLS04065I DEFAULT PROBLEM IDENTIFIER prob-id SUFFIX HAS INVALID FORMAT

Explanation: The default problem identifier **prob-id** suffix does not consist of numeric characters.

System Action: The subcommand processing terminates.

Programmer Response: Verify the default problem identifier using the SETDEF LIST subcommand. Reset the default problem identifier using the SETDEF subcommand, or specify the problem identifier with the subcommand.

BLS04066I ERROR WHILE REQUESTING CONFIRMATION, RC=rc

Explanation: The TSO PUTGET service routine issued a nonzero return code **rc**.

System Action: The subcommand processing terminates.

Programmer Response: Reissue the subcommand. See the PUTGET section in the *TSO Guide to Writing a TMP or a Command Processor* publication for the meaning of a nonzero return code.

BLS04067D INVALID REPLY - ENTER Y CONTINUE, N TO TERMINATE

Explanation: The reply received to a previous confirmation message was not Y or N. The user is requested to enter the appropriate response.

System Action: If a valid response is not received after five additional attempts, the system terminates the subcommand.

Programmer Response: Supply the appropriate response (either Y or N).

BLS04068I INTERNAL ERROR; THE STATUS RECORD FOR PROBLEM IDENTIFIER prob-id IS INVALID, SEQUENCE NUMBER = xxx

Explanation: While trying to add a data set to problem **prob-id**, the sequence number **xxx** (to be used for the next data set association record) contained data other than numeric characters.

System Action: The subcommand processing terminates.

Programmer Response: Follow steps in problem determination. If possible, correct the error in the problem directory and reissue the subcommand.

Problem Determination: Table I, items 1, 3, 4, and 29. Use IDCAMS PRINT to dump the appropriate records from the problem directory; see Table IV for details.

BLS04069I DEFAULT KEYWORD IGNORED; DEFAULT VALUE REMAINS UNCHANGED

Explanation: The subcommand specified the DEFAULT keyword, but it was ignored due to the occurrence of an error.

System Action: The default problem identifier or default data set name remains unchanged.

Programmer Response: Correct the error and reissue the subcommand.

BLS04070I to BLS04077I

BLS04070I RECORD SEQUENCE NUMBER OVERFLOW

Explanation: While writing the problem description into the problem directory, the next key field sequence number to be used was 10000, which will not fit into the sequence number field. The problem description for the problem is too large.

System Action: The remainder of the problem description is not transcribed into the problem directory for this problem.

Programmer Response: Decrease the size of the problem description to be used for this problem. Use a description containing fewer than 130,001 logical lines/records.

BLS04071I MODULE mod DETECTED AN INTERNAL ERROR

Explanation: An internal error occurred during execution of module **mod**. The messages following this message indicate in more detail the nature of the internal error.

System Action: The execution of the module terminates.

Programmer Response: Respond appropriately to the messages that follow.

Problem Determination: Table I, items 5, 16, and 29.

BLS04072I THE cbname POINTER DOES NOT POINT TO A VALID cbname CONTROL BLOCK. POINTER VALUE = adr

Explanation: The area pointed to by address adr does not contain a valid control block **cbname**.

System Action: The execution of the module terminates with the module issuing an IPCS user completion ABEND code of X'091' (decimal 0145).

Programmer Response: Retry the subcommand. If the error persists, perform problem determination.

Problem Determination: Table I, items 5, 16, and 29.

BLS04073I INVALID REQUEST TO DELETE MEMBER mem FROM DATA SET dsn

Explanation: A DELDSN or DELPROB subcommand invalidly requested deletion of member name **mem** in data set **dsn**. The deletion did not occur because IPCS does not allow the attribute MANAGED to be applied to a member of a data set. This condition may be the result of a programming error.

System Action: The dissociate process completes without the delete being performed. The subcommand processing continues. The data set is freed with a disposition of KEEP.

Programmer Response: None.

Problem Determination: Probable software error. Table I, item 29.

BLS04074I DATA SET dsn NOT SCRATCHED. DATA SET NAME IS OF FORM SYSN.

Explanation: A DELDSN or DELPROB subcommand issued a request to scratch data set **dsn**. However, the data set name is of the form SYSn. n is between 0 and 9 and is not a SYS1.DUMPnn data set.

System Action: The dissociate process continues without scratching the data set. The data set is freed with a disposition of KEEP.

Programmer Response: None.

BLS04075I DUMP DATA SET dsn NOT INITIALIZED; INVALID DATA SET ORGANIZATION

Explanation: A DELDSN or DELPROB subcommand attempted to initialize dump data set **dsn**. However, the data set organization was invalid. That is, it was not physical sequential or physical sequential unmovable.

System Action: The dissociate process continues without initializing the SYS1.DUMPnn data set. The data set is freed with a disposition of KEEP.

Programmer Response: None.

BLS04076I DUMP DATA SET dsn NOT INITIALIZED; DEVICE TYPE NOT DIRECT ACCESS

Explanation: A DELDSN or DELPROB subcommand attempted to initialize dump data set **dsn**. However, IPCS cannot initialize the dump data set because **dsn** does not reside on a direct access storage device (DASD).

System Action: The dissociate process continues without initializing the SYS1.DUMPnn data set. The data set is freed with a disposition of KEEP.

Programmer Response: None.

BLS04077I DEVICE TYPE MACRO FAILED FOR DUMP DATA SET dsn

Explanation: A DELDSN or DELPROB subcommand attempted to initialize dump data set **dsn**. The DEVTYPE macro issued an error code.

System Action: The dissociate process continues without initializing the SYS1.DUMPnn data set. The data set is freed with a disposition of KEEP.

Programmer Response: None.

BLS04078I ERROR INITIALIZING DUMP DATA SET dsn

Explanation: An error occurred while attempting to initialize dump data set **dsn**. The messages that follow indicate the cause of the error.

System Action: The dissociate process continues without initializing the SYS1.DUMPnn data set. An attempt is made to free the data set with a disposition of KEEP.

Programmer Response: Respond appropriately to the messages that follow.

BLS04079I ERROR SCRATCHING DATA SET dsn

Explanation: An error occurred while attempting to scratch data set **dsn**. The messages following this message indicate the cause of the error.

System Action: The dissociate process continues without the scratch being performed. The subcommand continues processing. The data set is freed with a disposition of KEEP.

Programmer Response: Respond appropriately to the messages that follow.

BLS04080I TEXT EXCEEDS MAXIMUM FIELD LENGTH. nnn CHARACTERS ALLOWED

Explanation: The value specified for a keyword contained more characters than are permitted; the maximum number of characters allowed is **nnn**.

System Action: The IKJPARS module will reprompt for the keyword's value.

Programmer Response: Reenter the value using a valid text length.

BLS04081I INVALID TEXT. MUST BE nnn CHARACTERS IN LENGTH

Explanation: The value specified for a keyword contained a number of characters other than **nnn**.

System Action: The IKJPARS module will reprompt for the keyword's value.

Programmer Response: Reenter the value using the correct text length.

BLS04082I UPDATE TERMINATED DUE TO USER'S REQUEST

Explanation: The user requested termination of the subcommand by using the IPCS attention exit. The subcommand could terminate without destroying the integrity of the IPCS problem or data set directory.

System Action: The subcommand processing terminates.

Programmer Response: To complete the update process, reissue the subcommand.

BLS04083I DATA SET dsn ALREADY ASSOCIATED WITH PROBLEM prob-id

Explanation: An attempt was made to add data set **dsn** to problem **prob-id**. **dsn** is already associated with that problem.

System Action: The subcommand completes processing normally. If the DEFAULT keyword was specified, it is established. If the CONFIRM keyword was specified, confirmation processing is bypassed.

Programmer Response: Ensure that the problem identifier is correct. Use the LISTPROB or LISTDSN subcommand to determine the status of the data set. If desired, use the MODDSN subcommand to change the existing attributes.

Problem Determination: Use IDCAMS PRINT to dump the appropriate records from the problem and data set directories; see Table IV for details.

BLS04084I PROBLEM ASSOCIATION RECORD MISSING FROM THE DATA SET DIRECTORY

Explanation: The correct problem association record for the data set identified in the previous BLS04089I message could not be found in the data set directory by a DELPROB or DELDSN subcommand.

- If this message is preceded by message BLS06402I and followed by message BLS04086I, both the problem identifier and the data set name (specified with the subcommand or defaulted) are unknown to IPCS.
- If this message is only preceded by message BLS06402I, the specified or default problem identifier is unknown to IPCS.
- If this message is only followed by message BLS04086I, the specified or default data set name is unknown to IPCS.
- If this message is not preceded by message BLS06402I and not followed by message BLS04086I, the problem and the data set name (specified or defaulted) are not associated.

System Action: The subcommand continues processing in order to complete the dissociate process. Because the problem association record is missing from the data set directory, the system assumes that the data set association record is also missing from the problem directory. Therefore, no attempt is made to erase the data set association record. If this assumption is correct, the problem directory and the data set directory will be synchronized at the end of the subcommand.

Programmer Response: Verify that the correct data set name and problem identifier have been

specified in the subcommand, or verify the defaults using the SETDEF LIST subcommand.

Problem Determination: Table I, item 29. Use IDCAMS PRINT to dump the appropriate records from the problem directory and data set directory; see Table IV for details.

BLS04085I DATA SET ASSOCIATION RECORD MISSING FROM THE PROBLEM DIRECTORY

Explanation: The correct data set association record for the data set identified in the previous BLS04089I message could not be found in the problem directory by a DELDSN DSNAME subcommand.

System Action: The subcommand continues processing in order to complete the dissociate process. The problem directory and the data set directory will be synchronized at the end of the subcommand.

Programmer Response: None.

Problem Determination: Table I, item 29. Use IDCAMS PRINT to dump the appropriate records from the problem directory and data set directory; see Table IV for details.

BLS04086I DATA SET BASE RECORD MISSING FROM THE DATA SET DIRECTORY

Explanation: The base record for the data set identified in the preceding BLS04089I message could not be found in the data set directory by a DELPROB or DELDSN subcommand.

System Action: The subcommand continues processing to complete the dissociate process. The data set is assumed to have the attribute UNMANAGED, and is therefore not scratched or initialized.

Programmer Response: If this message is preceded by message BLS04084I, verify that the correct data set name was specified with the subcommand, or verify the default data set name using the SETDEF LIST subcommand.

Problem Determination: Table I, item 29. Use IDCAMS PRINT to dump the appropriate records from the problem directory and data set directory; see Table IV for details.

BLS04087I DATA SET dsn SCRATCHED

Explanation: A DELDSN or DELPROB subcommand dissociated and scratched data set **dsn**. The data set had a management attribute of MANAGED, and was not associated with any other problem.

System Action: The subcommand processing continues.

Programmer Response: None.

Problem Determination: None.

BLS04088I DATA SET dsn INITIALIZED

Explanation: A DELDSN or DELPROB subcommand dissociated and initialized dump data set **dsn**. The dump data set had a management attribute of MANAGED, and was not associated with any other problem.

System Action: The SYS1.DUMPnn data set is freed with a disposition of KEEP. The subcommand continues processing.

Programmer Response: None.

Problem Determination: None.

BLS04089I DISSOCIATING DATA SET dsn

Explanation: A DELPROB or DELDSN subcommand began the dissociate process for data set **dsn**.

System Action: The subcommand continues the dissociate process.

Programmer Response: None.

BLS04090I NO DATA SETS ARE ASSOCIATED WITH PROBLEM prob-id

Explanation: There were no data sets found to be associated with problem **prob-id**. If this message is preceded by message BLS04001I, BLS04002I, or BLS06402I, **prob-id** is unknown to IPCS.

System Action: If this message is issued from a DELDSN subcommand, the subcommand terminates because there are no dissociations to perform. If this message is issued from a DELPROB subcommand, processing continues to delete the remaining information about the problem from the problem directory.

Programmer Response: If this message is preceded by message BLS04001I, BLS04002I, or BLS06402I, use the SETDEF LIST subcommand to verify the specified problem identifier or to verify the default problem identifier.

Problem Determination: Table I, item 29. Use IDCAMS PRINT to dump the appropriate records from the problem directory and data set directory; see Table IV for details.

BLS04091I BASE RECORD MISSING FOR DATA SET dsn

Explanation: The base record for data set **dsn** could not be found in the data set directory. (Each data set has a base record that contains attributes for that data set.) This message normally occurs as the result of a misspelled data set name with the DELDSN subcommand. However, it can also be issued if an out-of-synchronization condition exists between the problem directory and the data set directory.

This condition can be detected whenever a list of data sets and their attributes is being built based upon the information in the problem directory, such as during the processing of a LISTPROB DSNAMES, a DELDSN ALL CONFIRM or a DELPROB CONFIRM subcommand.

System Action: The subcommand processing continues. In no case is the named data set scratched.

Programmer Response: If the subcommand being executed is DELPROB or DELDSN ALL, a reply of Y to the confirmation message will attempt to re-synchronize the problem directory and the data set directory. Otherwise, the two directories will be left in their out-of-synchronization condition. If this problem continually recurs for different data sets, or continues to occur after a resynchronization attempt has been made, follow the problem determination procedures.

Problem Determination: Table I, items 1, 3, 4, 13, 25a, 26c (for IPCSPRnn) 29, and 34a. Use IDCAMS PRINT to dump the appropriate records from the problem and data set directories; see Table IV for details.

BLS05 Messages

BLS05100I PROBLEM prob-id HAS BEEN ADDED TO THE PROBLEM DIRECTORY

Explanation: The ADDPROB subcommand added new problem **prob-id** to the problem directory.

System Action: None.

Programmer Response: None.

BLS05101I PROBLEM NUMBERS ARE EXHAUSTED

Explanation: The next problem number to be assigned is 100,000, which is too large to fit into the problem number field.

System Action: The ADDPROB subcommand terminates.

Programmer Response: Define a new IPCS data base consisting of problem and data set directories. These data sets should be defined and initialized according to IPCS installation instructions.

BLS05104I SPECIFIED PROBLEM IDENTIFIER prob-id ALREADY EXISTS

Explanation: Problem identifier **prob-id** specified with the PROBLEM keyword is currently in use.

System Action: The subcommand processing terminates.

Programmer Response: Select another problem identifier or allow ADDPROB to do so by omitting the PROBLEM keyword.

BLS05301I NO PROBLEMS EXIST FOR RANGE prob-id1:prob-id2

Explanation: No problem identifiers exist in the problem directory in the range **prob-id1** to **prob-id2**.

System Action: The LISTPROB subcommand continues processing with the next problem identifier or range specified in the problem list.

Programmer Response: None.

Problem Determination: Table I, item 29. Verify that problems do exist within the range specified by using IDCAMS PRINT to dump the appropriate records for the problem range from the problem directory; see Table IV for details.

BLS05400I USER NOT AUTHORIZED TO DELETE PROBLEM (prob-id)

Explanation: The user attempted to delete problem **prob-id** and either:

- A TSO userid was defined in the parmlib member IPCSPRnn for the DELETEAUTHORITY keyword, and the user issuing this subcommand is not the authorized user.
- No TSO userid was defined in the parmlib member IPCSPRnn for the DELETEAUTHORITY keyword. However, the user's TSO userid does not match the userid stored in the problem directory as the owner of **prob-id**, nor does it match the TSO userid specified in the ADMINAUTHORITY keyword in the parmlib member IPCSPRnn.

System Action: The subcommand processing terminates. The problem is not deleted.

Programmer Response: Either obtain the required authority to delete the problem, or have the problem deleted by an IPCS user who has the proper authority.

BLS05401I **DELPROB TERMINATED**

Explanation: The user replied N to message BLS05402D.

System Action: The subcommand processing terminates. The problem is not deleted.

Programmer Response: None.

BLS05402D **ENTER Y TO CONFIRM DELETION, N TO TERMINATE**

Explanation: A DELPROB or DELDSN subcommand specified the CONFIRM option. The user is requested to verify that the item(s) should be deleted. The item(s) will be deleted if the user replies Y. If the user replies N, the subcommand will be terminated either by message BLS05401I or BLS06401I, and the item(s) will not be deleted.

System Action: If the response is Y, the subcommand deletes the item(s). If the response is N, the subcommand terminates without deleting the item(s).

Programmer Response: Enter Y or N as appropriate.

BLS06 Messages

BLS06401I **DELDSN TERMINATED**

Explanation: The user replied N to message BLS05402D.

System Action: The subcommand processing terminates. The data set is not dissociated.

Programmer Response: None.

BLS06402I **PROBLEM STATUS RECORD MISSING FOR PROBLEM prob-id**

Explanation: The status record for the problem identifier **prob-id** could not be found in the problem directory. If this message is followed, after confirmation processing is complete, either by message BLS04084I or BLS04090I, **prob-id** was not known to IPCS.

System Action: The subcommand continues processing in order to complete the dissociate process. Since the status record is missing, the problem is assumed to be owned by the user, so that any data set association records found may be dissociated.

Programmer Response: If this message is followed either by message BLS04084I or BLS04090I, verify the specified problem identifier, or use the SETDEF LIST subcommand to verify the default problem identifier.

Problem Determination: Table I, item 29. Use IDCAMS PRINT to dump the appropriate records from the problem directory and data set directory; see Table IV for details.

BLS07 Messages

BLS07001I **Invalid environment for ISPEXEC**

Explanation: During IPCS processing, the user issued the ISPEXEC subcommand but ISPF was not active.

System Action: Processing of the ISPEXEC subcommand terminates.

Programmer Response: Having started your IPCS session, activate ISPF and then reenter the ISPEXEC subcommand.

BLS07002I **ISPEXEC needs further qualification**

Explanation: When the ISPEXEC subcommand was entered, the qualifier, which indicates the requested ISPF service, was omitted.

System Action: Processing of the ISPEXEC subcommand terminates.

Programmer Response: Supply adequate qualifiers on the ISPEXEC subcommand to indicate the requested service to be performed.

BLS17 Messages

BLS17001I **LOAD FAILED FOR modname. PROCESSING CONTINUES.**

Explanation: During initialization of the common IPCS/PRDMP/SNAP exit interface, IPCS processing was unable to load the specified module **modname**.

System Action: Error condition; IPCS, PRDMP, or SNAP processing continues.

Programmer Response: Report the error to the system programmer.

BLS17002I **LOAD FAILED FOR MODEL/FORMATTER modname**

Explanation: The control block formatter could not load the formatting model or formatting module **modname** that was specified in the control block acronym table (CBAT).

System Action: IPCS, PRDMP, or SNAP does not format the specified control block; returns to the caller.

Programmer Response: Report the error to the system programmer.

BLS17003I **acronym NOT FOUND IN CONTROL BLOCK ACRONYM TABLE**

Explanation: The control block formatter could not find the requested **acronym** (passed in the ADPLPFMT parameter) in the control block acronym table (CBAT).

System Action: IPCS, PRDMP, or SNAP does not format the specified control block; returns to the caller.

Programmer Response: Verify that the acronym being passed is in the CBAT.

BLS17004I **acronym ENTRY IN CONTROL BLOCK ACRONYM TABLE IS NULL**

Explanation: The control block formatter found the **acronym** (passed in the ADPLPFMT field) in the control block acronym table (CBAT). However, the entry in the CBAT did not have a valid address to the formatting module or the formatting model.

System Action: IPCS, PRDMP, or SNAP does not format the specified control block; returns to the caller.

Programmer Response: Report the error to the system programmer.

BLS17005I **UNABLE TO OBTAIN STORAGE FOR CONTROL BLOCK ACRONYM TABLE PROCESSING**

Explanation: IPCS, PRDMP, or SNAP processing issued a GETMAIN macro instruction to obtain storage to be used for the processing of the control block acronym table (CBAT) but the storage was not available.

System Action: IPCS, PRDMP, or SNAP issues a return code of 16 indicating its inability to obtain storage for the CBAT.

Programmer Response: None.

BLS17006I **INSUFFICIENT STORAGE FOR EXIT**

Explanation: A module issued a GETMAIN macro instruction to obtain storage for its exit processing but storage was not available.

System Action: IPCS processes the next command or subcommand. PRDMP processes the next control statement.

Programmer Response: Increase the region size by either:

- Logging on again specifying the SIZE keyword in line mode.
- Indicating a larger size on the full screen LOGON menu.

BLS17007I **EXIT modname ABENDED WITH CODE xxx**

Explanation: Exit module **modname** abnormally ended with system completion code **scde**.

System Action: IPCS processes the next command or subcommand. PRDMP processes the next control statement.

Programmer Response: Respond appropriately to the system completion code.

BLS17009I **UNRECOGNIZED VERB**

Explanation: IPCS or PRDMP did not find the entered subcommand or command in the exit control table (ECT) or in the IPCS build directory list (BLDL) lookup.

System Action: IPCS processes the next command or subcommand. PRDMP processes the next control statement.

Programmer Response: Enter a valid command or subcommand.

BLS17012I **LINK TO MODULE mod FAILED {FOR VERB verb}**

Explanation: IPCS failed in its attempt to pass control to entry point **mod**. When the subcommand VERB is used, message BLS17012I also identifies the **verb**.

System Action: IPCS processes the next command or subcommand, and PRDMP processes the next control statement.

Programmer Response: Ensure that **mod** resides in either a library in the link list or in a data set that is in a STEPLIB. Using the write-to-programmer service the contents supervisor makes more detailed information regarding this situation available. To have write-to-programmer messages written at the TSO terminal enter the command

PROFILE WTPMSG MSGID

Follow the steps described in the contents supervisor message.

BLS17013I **NO OUTPUT PRODUCED BY xxxxxxxx**

Explanation: IPCS processing passed control to the specified exit module but no lines (output) were written to the printer or the terminal.

System Action: IPCS processes the next command or subcommand. PRDMP processes the next control statement.

Programmer Response: None.

BLS17014I **Trap of INPUT/OUTPUT(trap) is not supported for service** *SSS*

Explanation: The user has requested a trap action for a service *SSS*. *sss* is not supported.

System Action: Processing continues, but the trap is not set.

Programmer Response: None.

Problem Determination: None.

BLS17015I **{FILE(IPCSPARM)|SYS1.PARMLIB} must have partitioned organization**

Explanation: The information required to build a list of system-supplied exits that perform dump analysis and formatting could not be processed.

System Action: Processing continues.

Programmer Response: None.

Problem Determination: None.

BLS17016I **No traps are set, INPUT and/or OUTPUT keywords are needed.**

Explanation: The user entered a TRAPON subcommand that equates to NOINPUT and NOOUTPUT.

System Action: Processing continues, but no traps are set.

Programmer Response: Reenter the TRAPON subcommand with the INPUT or OUTPUT keywords.

Problem Determination: None.

BLS17017I **{FILE(IPCSPARM)|SYS1.PARMLIB} member BLSCECT not found**

Explanation: The information required to build a list of system-supplied exits that perform dump analysis and formatting could not be located.

System Action: Processing continues, but no system-supplied exits may be used.

Programmer Response: None.

Problem Determination: None.

BLS17018I **Error while processing {FILE(IPCSPARM)|SYS1.PARMLIB}**

Explanation: The information required to build a list of system-supplied exits that perform dump analysis and formatting could not be processed.

System Action: Processing continues, but no system-supplied exits may be used.

Programmer Response: None.

Problem Determination: None.

BLS17019I **No exit type specified for EP** *exitname*

Explanation: The information required to build a list of system-supplied exits that perform dump analysis and formatting is not properly specified.

System Action: Processing continues, but no system-supplied exits may be used.

Programmer Response: None.

Problem Determination: None.

BLS17040I **Array specification error:** *reason*

Explanation: The model being processed by the format model processor contains an error in a two-dimensional array specification. The *reason* may be one of the following:

1. more than two dimensions
2. orders out of range
3. upper bound < lower bound
4. will not fit on one line
5. the order is duplicated

When array dimensions are supplied by the calling exit program, the program could be at fault.

System Action: Processing of the array is bypassed. Any fields following the array will be processed.

Programmer Response: Correct the model and/or the exit program.

Problem Determination: Use the debug tool to display both the FMT input parameter and data (model) to determine which one is wrong when reason 3 occurs.

BLS18 Messages

BLS18000I *cbname* **AT** *adr*

Explanation: The ASMCHECK subcommand located control block *cbname* at address *adr*.

System Action: None.

Programmer Response: None.

BLS18001I **nnn I/O REQUESTS RECEIVED, xxx I/O REQUESTS COMPLETED BY ASM**

Explanation: The auxiliary storage manager (ASM) received **nnn** requests for paging I/O, and completed **xxx** requests. If **xxx** is substantially less than **nnn**, the system might have been experiencing paging problems.

System Action: None.

Programmer Response: None.

BLS18002I PAGE DATA SET nnn IS ON UNIT ddd

Explanation: The location of paging data set **nnn** is on hexadecimal unit number **ddd**.

System Action: None.

Programmer Response: None.

BLS18003I I/O REQUEST ACTIVE FOR ABOVE DATA SET

Explanation: IPCS processing scanned the I/O request blocks (IORBs) for the paging data set (identified by the previous message BLS18002I) and found active I/O requests.

System Action: None.

Programmer Response: None.

BLS18004I IOSB FOR ABOVE HAD ABNORMAL IOSCOD VALUE X'xxx'

Explanation: The I/O supervisor block (IOSB) associated with the paging data set (identified by the previous message BLS18002I) had an abnormal completion code, which is indicated in the IOSCOD field. The value in the IOSNRMC field in the IOSB mapping macro indicates normal completion and is not necessarily zero.

System Action: None.

Programmer Response: None.

BLS18005I INVALID cbname AT ADDRESS adr, ASID nnn

Explanation: Retrieved control block **cbname** did not meet validation criteria. Either the control block was overwritten or the pointer used to find the block was bad. Validation criteria include:

- Testing the retrieved block for constants that must be present.
- Testing for boundary alignments.

System Action: None.

Programmer Response: None.

BLS18006I NUMBER OF MESSAGES QUEUED (UCMWQNR) IS nnn. LIMIT (UCMWQLM) IS xxx

Explanation: IPCS processing obtained the unit control module (UCM). The fields in the message text (displayed in decimal) and their meanings follow:

nnn indicates the total queued messages (according to the UCMWQNR field).

xxx indicates the limit of queued messages (according to the UCMWQLM field).

System Action: None.

Programmer Response: None.

BLS18007I nnn MAJOR WQES CHAINED FROM UCM

Explanation: IPCS processing counted the write-to-operator queue elements (WQEs) chained via their WQELKPA fields. The UCMWTOQ field of the unit control module (UCM) established addressability to the first WQE on the chain. **nnn** WQEs were counted.

System Action: None.

Programmer Response: None.

BLS18008I UCMSTS STATUS FLAG BYTE IS X'xxx' FOR FOLLOWING CONSOLE

Explanation: In examining the unit control module device entry (UCME), IPCS processing encountered a nonzero UCMSTS status byte value of xxx. Message BLS18009I displays the address of the console associated with this UCME.

System Action: None.

Programmer Response: None.

BLS18009I nnn WQES FOUND FOR CONSOLE adr

Explanation: IPCS performed the following processing:

- Searched the console queue element (CQE) chain from a given unit control module device entry (UCME).
- Counted **nnn** WQE(s).
- Obtained the unit control block (UCB) for the console at address **adr**.

System Action: None.

Programmer Response: None.

BLS18010I OUTSTANDING REPLY nnn

Explanation: IPCS processing encountered an operator reply element (ORE) with reply identifier nnn.

System Action: None.

Programmer Response: None.

BLS18011I to BLS18020I

BLS18011I **SYS1.DUMPnn--mm/dd/yy hh:mm:ss--xxx**

Explanation: For SYS1.DUMPnn (where nn is the data set number) the time-of-day clock indicates that the dump was created on the date mm/dd/yy and at the time hh:mm:ss. The dump title xxx is also shown.

System Action: None.

Programmer Response: None.

BLS18012I **UNABLE TO OPEN SYS1.DUMPnn**

Explanation: IPCS processing allocated SYS1.DUMPnn (where **nn** is the data set number), but the OPEN subcommand failed.

System Action: Terminates operation for this dump data set. Processing continues with the next dump data set.

Programmer Response: None.

BLS18013I **NO TITLE RECORD FOUND FOR SYS1.DUMPnn**

Explanation: IPCS processing read ten records from SYS1.DUMPnn (where nn is the data set number), and found no header record.

System Action: Terminates operation for this dump data set. Processing continues with the next dump data set.

Programmer Response: None.

BLS18014I **SYS1.DUMPnn IS EMPTY**

Explanation: SYS1.DUMPnn (where **nn** is the data set number) is empty.

System Action: Terminates operation for this dump data set. Processing continues with the next dump data set.

Programmer Response: None.

BLS18015I **Requested module not found**

Explanation: The FINDMOD subcommand could not locate the requested module after checking the following:

- The symbol table
- The active link pack area CDE chain
- The link pack directory entries (LPDEs)

System Action: None.

Programmer Response: None.

BLS18016I **Major name for this module is** *mod*

Explanation: IPCS processing found the requested module *mod* by checking a minor link pack directory entry (LPDE) or a contents directory entry (CDE). The system fetched the major LPDE or CDE that contained *mod*.

System Action: None.

Programmer Response: None.

BLS18017I **JOB jjj, ASCBnnn AT adr**

Explanation: For the address space identified by decimal **nnn**, IPCS processing found job name **jjj** at address **adr**. Either the ASCBJBNI field or the ASCBJBNS field of the address space control block (ASCB) established addressability to job name **jjj**. IPCS processing entered the ASCB with the name **ASCBnnn** in the symbol table.

System Action: None.

Programmer Response: None.

BLS18018I **SUMMARY DID NOT FIND JOB jjj**

Explanation: During SUMMARY subcommand processing, the user requested job name **jjj** but IPCS processing did not find any corresponding valid address space control blocks (ASCBs).

System Action: None.

Programmer Response: None.

BLS18019I **TCBnnnxx AT adr**

Explanation: IPCS processing found a task control block (TCB) at address **adr**. The fields in the message text and meanings follow:

nnn is the ASID (in decimal).

xx is an indicator of the TCB's position on the chain from the address space extension block (ASXB). For example, the first TCB on the chain is indicated by AA, the second TCB by AB, and so forth.

IPCS processing places an entry for this TCB using the name **TCBnnnxx** in the symbol table.

System Action: None.

Programmer Response: None.

BLS18020I **xxx AT adr**

Explanation: IPCS processing located a request block (RB) at address **adr** in the ASID being displayed. **xxx** might be a PRB, a TIRB, an IRB, an SIRB, an SVRB, or a ??RB depending upon the RB. ("??" indicates an unknown RBFTP field and possibly an invalid RB).

System Action: None.

Programmer Response: None.

BLS18021I **THE ABOVE ADDRESS SPACE IS SWAPPED OUT**

Explanation: For the address space control block (ASCB) being displayed, the ASCBNOQ field contains B'1'. Therefore, the address space extension block (ASXB) and subsequent control blocks might not be available.

System Action: None.

Programmer Response: None.

BLS18022I **TCBCMP FIELD IS NONZERO FOR cbname**

Explanation: The TCBCMP field for TCB **cbname** contains a nonzero value.

System Action: None.

Programmer Response: None.

BLS18023I **TCBRTWA FIELD IS NONZERO FOR cbname**

Explanation: The TCBRTWA field for the TCB **cbname** contains a nonzero value.

System Action: None.

Programmer Response: None.

BLS18025I **FIND MASK is not valid**

Explanation: The AND mask entered with the FIND subcommand was invalid for one of the following:

- Not the same length as the search argument.
- A value resulting in more than 8 bytes (16 hexadecimal digits).
- A value not of the form X'xxxx'.

System Action: The subcommand processing terminates.

Programmer Response: Reenter the subcommand with the proper mask.

BLS18026I **FIND terminated due to discontinuity near** *adr*

Explanation: The FIND subcommand specified the BREAK option. At or beyond address *adr*, storage necessary for the search was not available in the dump.

System Action: The subcommand processing terminates.

Programmer Response: None.

BLS18027I **FIND not previously entered**

Explanation: A FIND subcommand specified no search argument. The FIND subcommand had not been entered earlier in the session.

System Action: The subcommand processing terminates.

Programmer Response: Enter the FIND subcommand with a search argument.

BLS18028I **Argument not found, searched to end of FINDAREA**

Explanation: The FIND subcommand accessed all the dump data required through the upper limit of the FINDAREA. However, the FIND subcommand did not find a match. The FINDAREA symbol describes the area explicitly (by the ADDRESS keyword) or implicitly (FIND without the ADDRESS keyword) in effect for the FIND subcommand.

System Action: None.

Programmer Response: None.

BLS18029I **Comparison discontinued after** *nnn* **bytes**

Explanation: A discontinuity in storage was encountered before all bytes of the COMPARE operands were examined. *nnn* bytes were examined.

System Action: None.

Programmer Response: None.

BLS18030I **Operand 1 is {greater than | less than | equal to | not equal to } operand 2**

Explanation: The data represented by the first operand was logically compared to the data represented by the second operand (subfield of WITH keyword) and found to be greater than, less than, equal to, or not equal to it.

System Action: None.

Programmer Response: None.

BLS18031I **Operand lengths unequal, shorter length** *nnn* **used for comparison**

Explanation: The entered or implied lengths of COMPARE operands were not equal. The comparison was performed using the shorter length.

System Action: None.

Programmer Response: None.

BLS18038I to BLS18048I

BLS18038I NO EXISTING RECORDS FOUND

Explanation: The SCAN subcommand did not locate any entries in the specified map range.

System Action: Error condition; subcommand processing completes.

Programmer Response: None.

BLS18039I NO NEW RESULTS PRODUCED

Explanation: IPCS processing scanned the map in the specified range, and produced no new map records or scan results.

System Action: Warning condition; subcommand processing completes.

Programmer Response: None.

BLS18040I nnn RESOURCES, [xxx SELECTED OF WHICH] yyy SHOW CONTENTION

Explanation: The ENQCHECK subcommand processed nnn global resource serialization queue control blocks (ISGQCBs). If a major resource name was designated as a selection criterion, the subcommand displays the number of resources selected, **xxx**. Furthermore, the subcommand displays the number of resources selected and subject to contention, **yyy**, which is determined by locating the ISGQCBs satisfying one of the following two criteria:

1. Use of the associated resource is prohibited.
2. Both exclusive and shared resource requests were detected during the examination of the global resource sharing queue elements (ISGQELs) associated with a resource.

System Action: None.

Programmer Response: None.

BLS18041I ISGQCB AT adr FOR THE FOLLOWING nnn-BYTE [GLOBAL] {SYSTEM/SYSTEMS/ASID nn} RESOURCE NAME:

Explanation: IPCS processing found a global resource serialization queue control block (ISGQCB) at address **adr**. The name that is printed on the following line (in dump format) is **nnn** bytes long. The ISGQCB represents a system, systems, or ASID **nn** resource. If a resource is known throughout a global resource serialization ring, this message displays the word GLOBAL.

System Action: None.

Programmer Response: None.

BLS18042I {SHARED|EXCLUSIVE} USE ISGQEL AT adr FOR [SYSID xxx,|SYSTEM yyy,]ASID nn

Explanation: IPCS processing encountered a shared or exclusive use of the global resource serialization queue element (ISGQEL) at address **adr**. If the name can be obtained from the global resource serialization ring status vector (ISGRSV), this message displays the name of the system, **yyy**, requesting the global resource. If the name cannot be obtained, this message displays the numeric value associated with the requesting system, **xxx**. This message also displays the address space, **nn**, requesting the resource for both local and global resources.

System Action: None.

Programmer Response: None.

BLS18043I BEGINNING address outside of FINDAREA

Explanation: The FIND subcommand did not specify a beginning address. Using the symbol X, the resulting aligned address is outside the FINDAREA. If the LISTSYM subcommand is issued, IPCS processing will display this address.

System Action: The subcommand processing terminates.

Programmer Response: Reenter the subcommand with a beginning address.

BLS18046I ERROR LINKING EXIT pgm

Explanation: An error occurred while linking to the exit routine specified by an ASCBEXIT, a TCBEXIT or a VERBEXIT subcommand. Program name **pgm** might have been misspelled.

System Action: The subcommand processing terminates.

Programmer Response: To have write-to-programmer messages written at the TSO terminal enter the command

```
PROFILE WTPMSG MSGID
```

and then re-execute the subcommand. If the problem recurs, the system will produce a diagnostic message for guidance in problem determination.

BLS18048I nnn RECORDS, xxx RESCANNED, yyy VERIFIED, zzz UNSCANNED

Explanation: This message is a LISTMAP summary. The subcommand examined **nnn** entries in the map and found the following:

xxx entries were completely validated, and found to contain errors greater than or equal to the flag setting. The subcommand rescanned the entries to produce the error diagnostics.

yyy entries were displayed according to the display options in effect because the entries were at least partially validated.

zzz entries were ignored because no validation had been done.

System Action: None.

Programmer Response: None.

BLS18050I INVALID RANGE SPECIFIED

Explanation: The user specified an address range in which the second address was less than the first address.

System Action: The subcommand processing terminates.

Programmer Response: Reissue the subcommand with a correct address range specification.

BLS18051I adr = > UNALIGNED INDIRECT-ADDRESS-POINTER

Explanation: The user specified address adr through indirect addressing. That address was not on a fullword boundary.

System Action: Severe condition; the subcommand, for which the indirect address was specified, terminates.

Programmer Response: Reenter the subcommand with a valid indirect address.

BLS18052I Double precision register substituted

Explanation: The user specified a single-precision floating-point register but the program substituted a double-precision floating-point register data type. If informational messages are being received, IPCS displays this message.

System Action: Performs the specified operation as though a double-precision floating-point register was requested.

Programmer Response: None.

BLS18053I Invalid storage description

Explanation: The user entered an invalid combination of storage description operands in an address expression.

System Action: Severe condition; the subcommand, for which the address was specified, terminates.

Programmer Response: Respecify the subcommand with a valid description of storage.

BLS18054I ARRAY ATTRIBUTES IGNORED

Explanation: The user specified an array through the use of the DIMENSION, ENTRIES, or MULTIPLE operand that exceeds the allowable upper limit for that request. The largest address values supported are 4095 for CPU status, 4103 for header information, and 2^{24}-1 for absolute, real, or virtual storage. IPCS processing converts the request to a SCALAR request with the length adjusted to the upper boundary.

System Action: Warning condition; the subcommand continues processing as described above.

Programmer Response: Respecify the subcommand with a valid array dimension.

BLS18058I Errors detected in *xxx* at *adr*:

Explanation: IPCS processing found damaged storage in the dump for area *xxx* at address *adr*. The SCAN messages follow this message.

System Action: The system continues processing.

Programmer Response: None.

BLS18059I Located via *xxx* at *adr*

Explanation: This message identified address *adr* of the locating area *xxx*.

System Action: None.

Programmer Response: None.

BLS18060I Invalid data in summary dump header record

Explanation: During dump initialization, IPCS processing detected invalid data in the summary dump header record.

System Action: Some summary dump information can be ignored as dump initialization continues.

Programmer Response: None.

BLS18061I Could not create an RPL

Explanation: The IPCSDDIR TSO command received a non-zero return code from the GENCB macro when it attempted to build a VSAM RPL.

System Action: IPCSDDIR terminates with a return code of 16. The dump directory specified on the command is not loaded.

Programmer Response: None.

BLS18062I to BLS18070I

BLS18062I **Could not create an ACB**

Explanation: The IPCSDDIR TSO command received a non-zero return code from the GENCB macro when it attempted to build an ACB.

System Action: IPCSDDIR terminates with a return code of 16. The dump directory specified on the command is not loaded.

Programmer Response: None.

BLS18063I **MAJOR RESOURCE NOT FOUND**

Explanation: The ENQCHECK subcommand specified a major resource name, but no major resource names in the dump began with that name.

System Action: Warning condition; subcommand processing completes.

Programmer Response: None.

BLS18064I **NO SYMBOLS FOUND**

Explanation: The LISTSYM or the DROPSYM subcommand specified symbols, but the subcommand did not find the requested symbols in the symbol table.

System Action: Warning condition; subcommand processing completes.

Programmer Response: None.

BLS18065I **NO SYMBOLS SELECTED**

Explanation: The LISTSYM or the DROPSYM subcommand specified some symbols that were found in the symbol table, but the subcommand did not select any of the symbols for processing.

System Action: Error condition; subcommand processing completes.

Programmer Response: Use the PURGE keyword if you want the symbols dropped.

BLS18066I **SYMBOL xxx NOT FOUND**

Explanation: The LISTSYM or the DROPSYM subcommand specified symbol **xxx**, which was not in the symbol table.

System Action: Warning condition; subcommand processing completes.

Programmer Response: None.

BLS18067I **SYMBOL xxx NOT SELECTED**

Explanation: The LISTSYM or the DROPSYM subcommand specified symbol **xxx**, which was found in the symbol table, but was not eligible to be selected.

System Action: Error condition; subcommand processing completes.

Programmer Response: Use the PURGE keyword if you want the symbol dropped.

BLS18068I **NO SYMBOLS FOUND IN RANGE xxx:yyy**

Explanation: The LISTSYM or the DROPSYM subcommand specified a range of **xxx** to **yyy** to locate symbols in the symbol table. The subcommand did not find any symbols in the specified range.

System Action: Warning condition; subcommand processing completes.

Programmer Response: None.

BLS18069I **NO SYMBOLS SELECTED IN RANGE xxx:yyy**

Explanation: The LISTSYM or the DROPSYM subcommand found symbols in the symbol table in the range **xxx** to **yyy**. The subcommand did not select any symbols.

System Action: Error condition; subcommand processing completes.

Programmer Response: Use the PURGE keyword if you want the symbols in the range dropped.

BLS18070I **Storage description adjusted**

Explanation: IPCS processing found an invalid description of storage, which resulted from any of the following conditions:

- Merging the address space associated with a symbol or register and the address specified on the subcommand
- Using attributes previously associated with a symbol or register
- Using default attributes

IPCS adjusted the attributes to form a valid set.

System Action: The subcommand processing continues.

Programmer Response: None.

BLS18071I DELIMITER ERROR IN OPERAND FIELD

Explanation: IPCS processing detected a delimiter error in the operand field. Upon return from a verb exit routine, the additional return code in the PRDMP parameter list extension was 4. This return code does not affect the return code from the VERBEXIT subcommand.

System Action: Processes the next IPCS subcommand or PRDMP control statement.

Programmer Response: Correct the delimiter error and reenter the command.

BLS18072I INVALID KEYWORD IN OPERAND FIELD

Explanation: IPCS processing detected an invalid keyword in the operand field. Upon return from a verb exit routine, the additional return code in the PRDMP parameter list extension was 8. This return code does not affect the return code from the VERBEXIT subcommand.

System Action: Processes the next IPCS subcommand or PRDMP control statement.

Programmer Response: Reenter the command specifying a valid keyword.

BLS18073I SYNTAX ERROR IN OPERAND FIELD

Explanation: IPCS processing detected a syntax error in the operand field. Upon return from a verb exit routine, the additional return code in the PRDMP parameter list extension was 12. This return code does not affect the return code from the VERBEXIT subcommand.

System Action: Processes the next IPCS subcommand or PRDMP control statement.

Programmer Response: Correct the syntax error and reenter the command.

BLS18074I ENTRY(IES) KEYWORD IGNORED

Explanation: The user specified both the ENTRIES and the SCALAR keywords. The SCALAR keyword takes precedence and IPCS does not process the ENTRIES keyword.

System Action: The subcommand continues processing as described above.

Programmer Response: If the data is to be considered as an array do not use the SCALAR keyword.

BLS18075I Indirect address could not be resolved

Explanation: The user specified an address with one or more levels of indirection that could not be resolved. This may be caused by the address being outside of the valid limits or the storage not being available.

System Action: Severe condition; the subcommand, for which the indirect address was specified, terminates.

Programmer Response: Reenter the subcommand with a valid indirect address.

BLS18076I nnn PROBES WERE EFFECTIVE

Explanation: nnn scan probes, which were initiated by the scan routine, produced new results.

System Action: None.

Programmer Response: None.

BLS18077I *subcommand* **processing may not be valid for a VIRTUAL dump**

Explanation: The entered *subcommand* used information that might have been dynamically changed during SDUMP processing. This might prevent satisfactory subcommand processing.

System Action: Subcommand processing continues as if the data provided in the dump is valid.

Programmer Response: If the dump can be processed without the use of the suspect data, continue dump analysis. If not, be aware that the use of such data can produce misleading results.

BLS18078I *subcommand* **processing is not valid for ACTIVE storage**

Explanation: The specified *subcommand* does not process active storage because it would require access to storage in another private address space.

System Action: Processing terminates.

Programmer Response: None.

Problem Determination: None.

BLS18079I *subcommand* **processing only supports MVS/XA dumps**

Explanation: The specified *subcommand* does not process sequential data sets, partitioned data set directories, or partitioned data set members.

System Action: Processing terminates.

Programmer Response: None.

Problem Determination: None.

BLS18081I to BLS18102I

BLS18081I **IMPROPER ENVIRONMENT FOR DSPL3270**

Explanation: DSPL3270 supports a primary screen size (24 x 80).

System Action: The DSPL3270 subcommand processing terminates.

Programmer Response: Restrict the use of the DSPL3270 subcommand to supported facilities.

BLS18082I **DSPL3270 READY TO REWRITE THE SCREEN--DEPRESS ENTER WHEN READY**

Explanation: The DSPL3270 subcommand specified the PAUSE operand. This message indicates that the DSPL3270 function is available.

System Action: None.

Programmer Response: If full-screen mode is desired, press the ENTER key.

BLS18086I **MAJOR NAME FOR THIS MODULE IS mod**

Explanation: The requested module **mod** was found via a minor link pack directory entry (LPDE) or contents directory entry (CDE). The major LPDE or CDE that contained the major name mod was fetched.

System Action: None.

Programmer Response: None.

BLS18087I **Comparison mask is not valid**

Explanation: The AND mask entered with the COMPARE subcommand was either:

- Not the same length as the object to which it was to be applied.
- A picture or text string.

System Action: Subcommand execution is terminated.

Programmer Response: Reenter the subcommand with the proper mask.

BLS18090I **ASID must be in the range 1:65534**

Explanation: All address space identifiers must be in the range 1 to 65534.

System Action: Prompts for correct values.

Programmer Response: Enter a valid ASID.

BLS18092I **CPU ADDRESS MUST BE IN THE RANGE 0:15**

Explanation: Only CPU addresses 0 to 15 are valid.

System Action: Prompts for correct values.

Programmer Response: Enter a valid CPU address.

BLS18094I **nnn BLOCKS PROCESSED**

Explanation: The RUNCHAIN subcommand processed **nnn** control blocks before terminating.

System Action: None.

Programmer Response: None.

BLS18095I **Invalid module name**

Explanation: The FINDMOD subcommand specified an invalid module name because of one of the following:

- The CHARACTER representation for the module name was specified or defaulted, but more than eight characters were entered.
- The HEXADECIMAL representation for the module name was specified, but invalid characters were entered.

System Action: The subcommand processing terminates.

Programmer Response: Reenter the subcommand with a valid module name.

BLS18100I **adr-space adr NOT AVAILABLE [FOR symbol]**

Explanation: A request for storage from the dump could not be satisfied. The first byte that could not be retrieved was in address space **adr-space** at location **adr**. If a symbol was given, this message displays **symbol**

System Action: Warning condition; the subcommand processing completes for this control block.

Programmer Response: None.

BLS18102I **Invalid Equate Symbol record filed at** *offset*

Explanation: Prior to the use of an equate symbol record (commonly passed as a parameter between IPCS subroutines), IPCS attempted to validate the record and determined that the record was in error.

System Action: If running with "SETDEF TEST," a user abend (X'06D') is generated. If running with "SETDEF NOTEST," the request involving the equate symbol record fails. This may cause the current subcommand to terminate or to produce partial results.

Programmer Response: If the error is in a user written exit, determine which module created the bad equate symbol record and correct the problem.

Problem Determination: Table I, items 5, and 16.

BLS18104I SYMBOL xxx NOT FOUND

Explanation: IPCS searched the symbol table and did not find symbol **xxx**.

System Action: Severe condition; the service routine processing terminates.

Programmer Response: None.

BLS18114I nnn RECORDS ERASED

Explanation: The DROPMAP subcommand erased **nnn** map records in the specified range.

System Action: None.

Programmer Response: None.

BLS18116I RESERVED UNIT ddd, UCB AT adr

Explanation: The hexadecimal unit device number **ddd** has a unit control block (UCB) at address **adr**.

System Action: None.

Programmer Response: None.

BLS18122I Initialization in progress for *dsn*

Explanation: FILE(IPCSDDIR) contains no description of the content for data set *dsn*. IPCS requires access to the data set as a dump. IPCS begins to generate a description of the data set.

System Action: IPCS begins dump initialization.

Programmer Response: None.

BLS18123I nnn BLOCK(S), xxx BYTES, IN dsn

Explanation: Message BLS18122I indicated that the system completed dump initialization. FILE(IPCSDDIR) now contains a description of the content of data set **dsn**. There were **nnn** records and **xxx** bytes present in the data set.

System Action: IPCS begins processing of the subcommand that caused dump initialization to be performed.

Programmer Response: None.

BLS18124I TITLE=xxx

Explanation: During dump initialization, IPCS processing read a nonblank dump title **xxx**.

System Action: IPCS continues dump initialization.

Programmer Response: Use the title to obtain a better understanding of the context in which the dump was produced. Use the symbol TITLE to recall the information later.

BLS18125I CPU(adr) STATUS AVAILABLE

Explanation: During dump initialization, IPCS processing read the CPU registers for the processor at address **adr**.

System Action: IPCS continues dump initialization.

Programmer Response: None.

BLS18126I CPU(adr) STATUS MAY BE INVALID

Explanation: During dump initialization, IPCS processing read the CPU registers for the processor at address **adr**. The information provided may be inaccurate.

System Action: IPCS continues dump initialization.

Programmer Response: Display the registers using IPCS. If possible, employ debugging methods that do not require CPU registers.

BLS18127I No dump title

Explanation: During dump initialization, IPCS processing read a blank title.

System Action: IPCS continues dump initialization.

Programmer Response: Use other sources of information to obtain an understanding of the context in which the dump was produced.

BLS18128I CPU(adr)-ONLY GPR STATUS IS VALID

Explanation: During dump initialization, IPCS processing read the CPU registers for the processor at address **adr**. Only the general purpose registers should be considered accurate.

System Action: IPCS continues dump initialization.

Programmer Response: Display suspected registers using IPCS. If possible, employ debugging methods that do not require those registers.

BLS18130I to BLS18154I

BLS18130I NO DUMP DATA SET

Explanation: The entered subcommand requires the use of a data set as a dump, but no data set name has been supplied.

System Action: The subcommand processing terminates.

Programmer Response: Reenter the subcommand specifying a dump data set name. You can also accomplish this by using a separate SETDEF subcommand.

BLS18132I DSORG for *dsn* is not supported

Explanation: The entered subcommand requires the use of default data set *dsn* as a dump. The default data set does not have a data set organization (DSORG) that is valid for dump data sets.

System Action: The subcommand processing terminates.

Programmer Response: Establish a default dump data set and reenter the subcommand.

BLS18133I INITIALIZATION UNABLE TO OPEN MVS DUMP DATA SET

Explanation: The entered subcommand requires the use of the default data set as a dump. The default data set cannot be opened.

System Action: The subcommand processing terminates.

Programmer Response: Establish a default dump data set and reenter the subcommand.

BLS18134I SYS1.DUMPnn--INVALID DSORG

Explanation: SYS1.DUMP**nn** (where nn is the data set number) does not have a data set organization (DSORG) that is valid for dump data sets.

System Action: Terminates operation for this dump data set. Processing continues with the next dump data set.

Programmer Response: None.

BLS18135I No [translation] records {deleted|to delete} for {ACTIVE|DSNAME(*dsname*)| FILE(*ddname*)}

Explanation: The DROPDUMP subcommand did not perform the action requested.

System Action: Error condition; subcommand processing completes.

Programmer Response: None.

BLS18136I Description of {ACTIVE|DSNAME(*dsname*)| FILE(*ddname*)} is in use

Explanation: DROPDUMP RECORDS(ALL) processing was requested for a dump while the description of that dump was in use.

System Action: This is a serious condition and the system terminates dump processing.

Programmer Response: This should only happen when you request DROPDUMP processing while using the IPCS dialog:

- If you asked the IPCS command to process the dump before beginning your ISPF session, end the ISPF session and request DROPDUMP processing there.
- If you asked the IPCS dialog to process the dump from another logical screen, request the DROPDUMP processing from that screen.

BLS18150I NAME VALUE AND CHAIN VALUE RESULT IN INVALID NAME LENGTH

Explanation: On the RUNCHAIN subcommand the length of the NAME operand plus the digits for the CHAIN value created a symbol name that is invalid because it exceeds 31 characters.

System Action: The subcommand processing terminates.

Programmer Response: Reenter the subcommand specifying the NAME operand with a valid length.

BLS18151I CHAIN POINTER POINTS PAST END OF STORAGE

Explanation: The RUNCHAIN subcommand detected an invalid chain pointer.

System Action: The subcommand processing terminates.

Programmer Response: None.

BLS18154I Unable to open *dsn* for yyy

Explanation: A data set *dsn* required for subcommand processing could not be opened.

System Action: The subcommand processing terminates.

Programmer Response: None.

BLS18155I DSORG OF TARGET DATA SET IS NOT SUPPORTED

Explanation: The target data set of the COPYDUMP subcommand must have one of the following data set organizations (DSORG): physical sequential, direct, or undefined. IPCS does not allow any other type of DSORG.

System Action: The subcommand processing terminates.

Programmer Response: Specify as the target data set one with a DSORG allowed by IPCS.

BLS18156I INCONSISTENT SPECIFICATION OF INPUT DATA SET

Explanation: If the input data set was previously opened for processing (and not closed) using the INDATASET keyword, then INDATASET must be used to resume processing (or the input data set left unspecified). Likewise, if processing was initiated using the INFILE specification, then resumption of processing must be done using the INFILE keyword.

System Action: Processing terminates.

Programmer Response: If the data set was opened using INDATASET, then use INDATASET again; or if INFILE was used, then use INFILE again. Optionally, processing may be resumed by omitting both the INDATASET and INFILE keywords.

BLS18160D MAY SUMMARY DATA BE USED BY DUMP ACCESS? ENTER Y TO USE, N TO BYPASS

Explanation: This message requests the user to verify whether the summary data is to be used for initialization. The summary dump records are contained in the dump data set. The data contained in the summary records, which is taken from a copy of the shared system areas, might not match the actual state of the system at the time the rest of the dump was taken.

System Action: If the user enters Y, initialization processes the summary data. It is important to recognize that if the user selects Y, then some subcommands that use information in the SUMDUMP records may be affected. For example, if a dump is produced as a result of a SDUMP macro being issued with BRANCH = YES specified, that dump may have values for PSAAOLD that differ between the SUMDUMP records and the regular dumped records.

If the user enters N, initialization bypasses processing of all the summary records and produces a dump directory that is equivalent to the output of PRDMP without the SUMDUMP option.

Programmer Response: Reply Y to use the summary data or N to bypass it.

BLS18161I Wrong EPOCH detected in the time of day clock

Explanation: The first bit of a time-of-day clock value should always be set to one for the current epoch. The STATUS subcommand detected a zero value when obtained from the dump being processed.

System Action: IPCS attempts to format the value in the clock. It may not be accurate.

Programmer Response: Use alternate means, if possible, to determine the time the dump was produced.

BLS18162I UNABLE TO OPEN INPUT DATA SET

Explanation: The input data set could not be opened.

System Action: The subcommand processing terminates.

Programmer Response: Check that the input data set was correctly specified.

BLS18163I NO TARGET DATA SET

Explanation: A target data set must be specified on the command line, but was not.

System Action: The subcommand processing terminates.

Programmer Response: Use either the OUTDATASET keyword or the OUTFILE keyword to specify the target data set.

BLS18164I TARGET DATA SET CAN NOT BE MEMBER OF A PDS

Explanation: The COPYDUMP subcommand does not allow a member name to be specified as part of the name of the target data set.

System Action: The subcommand processing terminates.

Programmer Response: Specify a sequential data set as the target data set.

BLS18165I UNABLE TO OPEN TARGET DATA SET

Explanation: The COPYDUMP subcommand could not open the target data set for processing.

System Action: The subcommand processing terminates.

Programmer Response: Check that the target data set was correctly specified.

BLS18166I to BLS18174I

BLS18166I **INVALID DUMP HEADER RECORD**

Explanation: The first record read during the execution of the subcommand is not a valid header record.

System Action: The subcommand processing terminates.

Programmer Response: Reenter the subcommand to cause the dump with the invalid header to be skipped. Normal processing resumes when the next dump in the data set is encountered.

BLS18167I **DUMP nnn - NO TITLE IN DUMP**

Explanation: Dump **nnn**, which is being processed, has a valid header record but does not contain a title.

System Action: Processing continues.

Programmer Response: None.

BLS18168D **PROCEED WITH COPY? ENTER Y TO CONTINUE, N TO TERMINATE**

Explanation: COPYDUMP has verified that a dump is available from the input data set and that the subcommand is prepared to overwrite an existing target data set or to allocate a new target data set and load it with the dump from the input data set. This message requests the user to indicate whether the subcommand should continue or terminate.

System Action: If the user enters Y, the subcommand transcribes the dump. If the user enters N, the contents of an existing target data set remain unchanged and the subcommand terminates. IPCS does not allocate a new target data set.

Programmer Response: Enter Y to continue or N to terminate the subcommand.

BLS18169I **DUMP nnn IS BEING {COPIED/SKIPPED}**

Explanation: IPCS processing is either copying dump nnn from the input data set into the target data set or it is skipping dump **nnn**. (The COPYDUMP subcommand numbers the dumps in the input data set beginning with 1.)

System Action: Processing continues.

Programmer Response: None.

BLS18170I **nnn RECORDS, xxx BYTES, {COPIED/SKIPPED}**

Explanation: A dump from the input data set has been processed. The dump consists of **nnn** records and **xxx** bytes. If the dump was copied, the target data set has been closed and freed.

System Action: None.

Programmer Response: None.

BLS18171I **END OF INPUT DATA SET REACHED**

Explanation: An end-of-file (EOF) condition has occurred on the input data set.

System Action: The input data set is closed.

Programmer Response: None.

BLS18172I **SKIPPING PARTIALLY PROCESSED DUMP**

Explanation: During this IPCS session the user entered the COPYDUMP subcommand. An attention interrupt or an invalid header record being read halted the subcommand's processing.

System Action: IPCS resumes the dump scanning process at the point of interruption. Normal processing resumes when the next header record is encountered in the input data set or when end of file (EOF) is reached.

Programmer Response: None.

BLS18173I **DUMP nnn - TITLE=xxx**

Explanation: This message displays the title **xxx** from dump **nnn** in the input data set.

System Action: None.

Programmer Response: None.

BLS18174I **ASID in current ASCB apparently in error**

Explanation: The definition of the current ASCB (the ASCB addressed by the field PSAAOLD in the PSA) contains an identifier in field ASCBASID. This identifier represents the current address space, which is associated with the current ASCB. However, after tracing through the CVT and the ASVT (checking the symbol table entry for the ASCB), IPCS has determined that the two identifiers for the current address space do **not** match each other. Therefore, what should be the current ASCB may not be the current ASCB.

System Action: Processing of the active subcommand terminates.

Programmer Response: None required. Pursuing this discrepancy may lead to the detection of a valid system problem. It may also lead to the determination that symbol ASCBnnn has been improperly defined.

BLS18175I **Padding byte is not valid**

Explanation: An invalid value was designated for the padding byte to be used by the COMPARE subcommand.

System Action: Subcommand processing terminates.

Programmer Response: Correct the padding value and reenter the COMPARE subcommand.

BLS18176I **BLKSIZE(nnn) used for dsn**

Explanation: The data set control block (DSCB) for a dump data set contains a zero BLKSIZE value. The track capacity for the direct access device containing the data set has been used in an attempt to process the data set.

System Action: Subcommand processing continues.

Programmer Response: None.

BLS18177I **Unable to determine device type on which data set resides**

Explanation: The type of device containing a dump data set could not be determined. The data management DEVTYPE service completed with a non-zero return code.

System Action: Message BLS18182I, which identifies the data set, follows this message. Processing of the current subcommand terminates.

Programmer Response: Check that the data set was correctly specified.

BLS18178I **Dump must reside on direct access or tape**

Explanation: IPCS processes data sets that reside on either direct access storage device (DASD) or tape.

System Action: Message BLS18182I, which identifies the data set, follows this message. Processing of the current subcommand terminates.

Programmer Response: Check that the data set was correctly specified.

BLS18179I **Unable to determine characteristics of data set**

Explanation: The characteristics of the dump data set could not be determined. The data management OBTAIN service completed with a non-zero return code.

System Action: Message BLS18182I, which identifies the data set, follows this message. Processing of the current subcommand terminates.

Programmer Response: Check that the data set was correctly specified.

BLS18180I **Initialization failed for** *dsn*

Explanation: IPCS could not process the dump data.

System Action: Message BLS18182I, which identifies the data set, follows this message. Processing of the current subcommand terminates.

Programmer Response: Check that the data set was correctly specified.

BLS18181I **Unable to open dsn**

Explanation: IPCS could not open dump data set dsn.

System Action: Processing of the active subcommand terminates.

Programmer Response: Check that the data set was correctly specified.

BLS18182I *dsn* **is not usable**

Explanation: IPCS could not process dump data set dsn.

System Action: Processing of the active subcommand terminates. Related messages precede BLS18182I indicating why dsn could not be processed.

Programmer Response: Check that the data set was correctly specified.

BLS18183I **UNABLE TO GET BUFFER POOL FOR DUMP DATA SET**

Explanation: IPCS could not process the dump data set.

System Action: Message BLS18182I, which identifies the data set, follows this message. Processing of the current subcommand terminates.

Programmer Response: If any data sets are open but are not needed during the processing of the data set, which could not be prepared for IPCS processing, use the CLOSE subcommand to release the IPCS storage associated with the dump data sets. If no data sets satisfy this criterion, LOGON in a larger region and resume IPCS processing. When storage has been made available, retry processing of the dump data set.

BLS18185I to BLS18193I

BLS18185I **Dump directory describes some other MVS dump**

Explanation: IPCS could not process the dump data set. Some installations make use of a pool of dump data sets on direct access storage, storing their permanent copies of dumps on mass storage or tape and overlaying a data set in the pool prior to IPCS problem analysis. This action can create a situation where an IPCS dump directory describes dump A after a pool data set has been overlaid with dump B. To assure that this situation does not result in the waste of a significant amount of problem analysis time, IPCS copies some data from the dump data set to the dump directory when a description is created and uses that data to verify the identity of the dump whenever the description in the dump directory is about to be reused. When a mismatch is detected, IPCS produces this message and terminates the processing of the dump data set.

System Action: Processing of the active subcommand terminates.

Programmer Response: Check that the data set was correctly specified. If it was, use the DROPDUMP subcommand to remove the current description from the dump directory. This will remove the description of the other dump and will permit you to begin IPCS problem analysis for the dump now in the data set.

BLS18186I **UNABLE TO OPEN RECFM=F DATA SET FOR PROCESSING**

Explanation: IPCS could not open a dump data set.

System Action: Processing of the active subcommand terminates.

Programmer Response: Check that the data set was correctly specified.

BLS18187I **UNABLE TO OPEN RECFM=U DATA SET FOR PROCESSING**

Explanation: IPCS could not open a dump data set.

System Action: Processing of the active subcommand terminates.

Programmer Response: Check that the data set was correctly specified.

BLS18188I **UNABLE TO OPEN PDS DIRECTORY FOR PROCESSING**

Explanation: IPCS could not open a dump data set.

System Action: Processing of the active subcommand terminates.

Programmer Response: Check that the data set was correctly specified.

BLS18189I **UNABLE TO OPEN PDS MEMBER FOR PROCESSING**

Explanation: IPCS could not open a dump data set.

System Action: Processing of the active subcommand terminates.

Programmer Response: Check that the data set was correctly specified.

BLS18190I **Unable to find PDS member**

Explanation: IPCS could not process the dump data set.

System Action: Processing of the active subcommand terminates.

Programmer Response: Check that the data set was correctly specified.

BLS18191I **Data type in value must be A, C, F, H, OR X**

Explanation: IPCS could not process the value.

System Action: Processing of the active subcommand terminates.

Programmer Response: Reenter the subcommand, correcting the value specified.

BLS18192I **MORE THAN nnn DIGITS IN datatype VALUE**

Explanation: The user entered an excessive number of digits in a value. IPCS could not process the value. The "datatype" must be one of the following: FULLWORD POINTER, FULLWORD SIGNED, HALFWORD SIGNED or HEXADECIMAL.

System Action: Processing of the active subcommand terminates.

Programmer Response: Reenter the subcommand, correcting the value specified.

BLS18193I **INVALID DIGIT IN datatype VALUE**

Explanation: The user entered a digit other than 0-9 for a signed value or the user entered a digit other than 0-9 or A-F for a pointer value or for a hexadecimal value. IPCS could not process the value. The **datatype** must be one of the following: FULLWORD POINTER, FULLWORD SIGNED, HALFWORD SIGNED or HEXADECIMAL.

System Action: Processing of the active subcommand terminates.

Programmer Response: Reenter the subcommand, correcting the value specified.

BLS18194I **Invalid character in EBCDIC value**

Explanation: The user specified an invalid value in a character string. IPCS processing will accept only character codes having associated graphics.

System Action: Processing of the active subcommand terminates.

Programmer Response: Reenter the subcommand, correcting the value specified.

BLS18195I **Unpaired delimiters in EBCDIC value**

Explanation: Each apostrophe designated as part of an EBCDIC character string value must be entered as a pair of apostrophes.

System Action: Processing of the active subcommand terminates.

Programmer Response: Reenter the subcommand, correcting the value specified.

BLS18196I **More than 256 characters in EBCDIC value**

Explanation: The value described more than 256 bytes. The maximum allowed by IPCS is 256.

System Action: Processing of the active subcommand terminates.

Programmer Response: Reenter the subcommand, correcting the value specified.

BLS18197I **Invalid character in picture string**

Explanation: IPCS processing found invalid characters in a picture string operand. When describing a picture string use EBCDIC letters, numbers, and the following special characters:

=	equal to sign
@	at sign
#	pound sign
$	dollar sign
¬	not sign
.	period
-	minus sign
<	less than sign
>	greater than sign

System Action: Processing of the active subcommand terminates.

Programmer Response: Reenter the subcommand, correcting the value specified.

BLS18198I **More than 256 characters in picture string**

Explanation: The picture string value contained more than 256 bytes. The maximum allowed by IPCS is 256.

System Action: Processing of the active subcommand terminates.

Programmer Response: Reenter the subcommand, correcting the value specified.

BLS18199I **precision VALUE MAY NOT BE GREATER THAN nnn**

Explanation: The negative value is less than **nnn** IPCS processing could not process the value. The **precision** must be a FULLWORD or a HALFWORD.

System Action: Processing of the active subcommand terminates.

Programmer Response: Reenter the subcommand, correcting the value specified.

BLS18200I **precision SIGNED VALUE MAY NOT BE GREATER THAN nnn**

Explanation: The positive value is greater than **nnn**. IPCS processing could not process the value. The **precision** must be a FULLWORD or a HALFWORD.

System Action: Processing of the active subcommand terminates.

Programmer Response: Reenter the subcommand, correcting the value specified.

BLS18205I **ODD NUMBER OF DIGITS IN HEXADECIMAL VALUE**

Explanation: IPCS could not process the value.

System Action: Processing of the active subcommand terminates.

Programmer Response: Reenter the subcommand, correcting the value specified.

BLS18206I **{All|translation} records for *nnn* dump[s] dropped**

Explanation: If the message text indicates that ALL RECORDS FOR DUMP DROPPED, the DROPDUMP subcommand deleted all records pertaining to **nnn** dumps from the dump directory. If the message text indicates that TRANSLATION RECORDS FOR DUMP DROPPED, the DROPDUMP subcommand deleted translation records for **nnn** dumps from the dump directory.

System Action: None.

Programmer Response: None.

BLS18210I **Exit request for *reason* rejected**

Explanation: An exit, invoked through the IPCS ASCBEXIT, TCBEXIT, or VERBEXIT subcommand, used invalid parameters when calling a service routine to fulfill a request.

The request was rejected. *reason* may be one of the following:

invalid address space indicates that two or more of the bits (ADPLVIRT, ADPLREAL, ADPLCPU, and ADPLHDR), which define the address space, were on.

CPU(nnnn) STATUS indicates that status data for the designated central processor was requested. However, "nnn" is greater than 15, the limit for a CPU address supported by MVS.

storage LENGTH(nnn) indicates that "nnn" bytes of virtual or real storage were requested. However, "nnn" is greater than 4096, the limit for the storage access service routine.

storage beyond location 7FFFFFFF indicates that virtual or real storage locations beyond location 7FFFFFFF were requested. However, 7FFFFFFF is the largest address supported by IPCS.

System Action: The IPCS service routines reject all further requests from the exit. If the IPCS TEST option is in effect, the executing module issues a user completion ABEND code of X'804' (decimal 2052), requesting a dump.

Programmer Response: None.

Problem Determination: Table I, items 5, and 16.

BLS18211I **AMASK option rejected**

Explanation: An address mask specified on an IPCS ASCBEXIT, TCBEXIT, or VERBEXIT subcommand disagrees with the address mask indicated for the exit in parmlib member BLSCECT.

System Action: The AMASK option is ignored and the subcommand continues.

Programmer Response: None.

BLS18212I **TRKCALC macro failed**

Explanation: During dump initialization, attributes associated with dump data set dsn could not be accessed.

System Action: The system issues message BLS18182I to identify the dump data set being processed. Processing of the subcommand that required access to the data set discontinues.

Programmer Response: None.

BLS18213I **Only EQ and NE searches are supported for {picture|text} strings**

Explanation: The FIND subcommand attempted to locate values in storage using picture or text strings. Only an equal (EQ) or an unequal (NE) condition can occur when processing picture and text strings. When using both upper and lower case letters, IPCS processing can not differentiate picture strings from text strings.

System Action: Processing of the FIND subcommand terminates.

Programmer Response: Reenter the subcommand, specifying a valid combination of operands.

BLS18214I **The MASK option is not supported for {picture|text} strings**

Explanation: The COMPARE or the FIND subcommand cannot process a comparison of picture and text strings in combination with the MASK option.

System Action: Processing of the COMPARE or FIND subcommand terminates.

Programmer Response: Reenter the subcommand, specifying a valid combination of operands.

BLS18215I **STACK request rejected. The stack is full.**

Explanation: The STACK subcommand attempted to enter symbol Z99999 in the stack, but Z99999 was already present in the stack.

System Action: Processing of the STACK subcommand terminates.

Programmer Response: If the entry of new symbols is required, use the DROPSYM subcommand, the RENUM subcommand, or both to make stack entries available.

BLS18216I **MVS/XA IPCS does not support MVS/370 dumps**

Explanation: MVS/XA IPCS detected a MVS/370 dump data set.

System Action: Subcommand processing is terminated.

Programmer Response: Use MVS/370 IPCS to process MVS/370 dumps.

BLS18220I **An I/O error occurred on the dump dataset, a logical EOF will be processed.**

Explanation: The system issues this message when an I/O error was detected on the dump.

System Action: IPCS will process a partial dump.

Programmer Response: Check the system log to see if the error occurred when the the dump was taken.

BLS18300I **Storage not in dump**

Explanation: The image of the control block (identified by the preceding BLS18058I message) could not be retrieved from the dump.

System Action: Warning condition; tests that examine the contents of the control block will not be performed.

Programmer Response: None.

BLS18301I **Not on *alignment* boundary**

Explanation: IPCS processing detected that storage for the control block (identified by the preceding BLS18058I message) was not aligned on the designated boundary. **alignment** may be one of the following:

HALFWORD	An address divisible by 2
FULLWORD	An address divisible by 4
DOUBLEWORD	An address divisible by 8
PAGE (2K)	An address divisible by 2048
PAGE (4K)	An address divisible by 4096
n-BYTE	An address divisible by decimal number n

System Action: Severe condition; tests that examine the contents of the control block will not be performed.

Programmer Response: None.

BLS18302I **Locator *xxx*=*adr*. It may be damaged**

Explanation: Validation failed for the control block at address *adr*. Field *xxx* of the control block (identified by the preceding BLS18058I message) located the address. The field may be invalid.

System Action: Error condition; validity checking continues for this control block.

Programmer Response: None.

BLS18303I ***xxxxxxxx*** **field does not equal** ***yyyy***

Explanation: The *xxxxxxxx* control block identifier field does not contain the expected control block identifier *yyyy*. The control block identifier *yyyy* is from the control block identified in the preceding BLS18058I message.

System Action: Processing terminates and the control block is marked as not valid.

Programmer Response: Determine if the control block has been overlayed or whether the pointer used to access the control block is incorrect.

Problem Determination: None.

BLS18400I **Unable to identify current FRR stack**

Explanation: The current stack address in field PSACSTK does not match any of the PSA FRR stack addresses.

System Action: Processing continues.

Programmer Response: Check for possible overlaying of FRR stack address fields.

Problem Determination: None.

BLS18401I **Unable to complete FRR stack analysis, unknown stack pointer**

Explanation: The address of the stack that was interrupted by the last stack displayed does not match any of the PSA FRR stack addresses.

System Action: Processing continues.

Programmer Response: Check for probable overlaying of FRR stack address fields.

Problem Determination: None.

BLS18402I **FRR stack chain is circular**

Explanation: The address of an FRR stack appears in more than one stack address save field in the PSA.

System Action: The system continues processing.

Programmer Response: Examine the PSA for overlays.

Problem Determination: None.

BLS18405I **CPU *xx* is the recovery CPU that is taking CPU *yy* offline via ACR**

Explanation: Alternate Recovery (ACR) is in progress in the configuration. CPU *xx* periodically pretends to be CPU *yy* in order to run CPU *yy*'s FRR stack and perform general cleanup. Be aware that the information displayed in the diagnostic worksheet does not take into account this switching process. The PSACPUPA field shown in the worksheet indicates if the recovery CPU is currently switched to the dead CPU. This case occurs when PSACPUPA is different from the CPU number in the column heading.

System Action: Processing continues.

Programmer Response: None.

BLS18451I **Unable to identify the area at address:** *adr*

Explanation: IPCS was unable to locate a module, structure or area in the system for the specified address *adr*.

System Action: Processing continues.

Programmer Response: Re-issue the subcommand with a different address.

Problem Determination: The address may represent code that does not reside in a normal system area. Code that has been moved to another location after being loaded will fall into this category. If a module has been loaded into common storage, the CDE entry for the module resides in the address space which did the LOAD. If the ASID on the WHERE subcommand is different from the ASID that did the LOAD to global, the correct CDE will not be found.

IPCS can only identify STRUCTUREs and AREAs that have been previously defined.

BLS18452I **WHERE information is not available. PWHS passed by exit** *xxxxxxxx* **is not valid.** *reason*

Explanation: The WHERE service was called with a parameter list (mapped by BLSRPWHS) that failed a validity check. *xxxxxxxx* is the exit which passed the bad PWHS. *reason* contains one of the following error conditions:

NO PWHS ACRONYM.
The passed parameter list did not contain the PWHS identifier. Verify that the exit has generated a good WHERE parameter list and that it is initialized properly.

VIRTUAL ADDRESS IS REQUIRED.
The address passed to the WHERE service was not a virtual address.

System Action: The WHERE service request is terminated

Programmer Response: Correct the code that calls the WHERE service so that it specifies a valid request.

Problem Determination: When running in an IPCS environment, consider using the TRAPON function to view the parameter list on entry to the WHERE service.

BLS18520I **NOCPU keyword may only be used with standalone dump**

Explanation: NOCPU keyword was incorrectly received for a virtual only dump.

System Action: The system terminates the subcommand.

Programmer Response: None

BLS19 Messages

BLS19000I **ASID(nnn) GREATER THAN MAXUSER**

Explanation: A request has been received to locate the address space control block (ASCB) for ASID **nnn**, but ASID **nnn** exceeds the number of entries in the address space vector table (ASVT).

System Action: Processing related to the address space discontinues.

Programmer Response: None.

BLS19001I **ASID(nnn) NOT ACTIVE**

Explanation: A request has been received to locate the address space control block (ASCB) for address space **nnn**, but the address space vector table (ASVT) indicates that the address space is not active.

System Action: Processing related to the address space discontinues.

Programmer Response: None.

BLS19002I **LCCAVT INDICATES CPU(nn) IS NOT ONLINE**

Explanation: A request has been received to locate the logical configuration communication area (LCCA) for CPU **nn**, but the logical configuration communication area vector table (LCCAVT) indicates that CPU nn is not online.

System Action: None.

Programmer Response: None.

BLS19003I **PCCAVT INDICATES CPU(nn) IS NOT ONLINE**

Explanation: A request has been received to locate the physical configuration communication area (PCCA) for CPU **nn**, but the physical configuration communication area vector table (PCCAVT) indicates that CPU **nn** is not online.

System Action: None.

Programmer Response: None.

BLS19007I **CHANNEL x NOT SYSGENED**

Explanation: IPCS processing received a request to locate the unit control block (UCB) associated with a device on channel path **x**. However, channel path **x** was not defined during system generation.

System Action: Processing, related to the locate request, is discontinued.

Programmer Response: None.

BLS19008I **UNIT ddd NOT SYSGENED**

Explanation: IPCS processing received a request to locate the unit control block (UCB) associated with the device number **ddd**. However, device number **ddd** was not defined during system generation.

System Action: Processing, related to the locate request, discontinues.

Programmer Response: None.

BLS19009I **INVALID CHANNEL LOOKUP TABLE ENTRY**

Explanation: An entry in the channel lookup table (the first part of the table pointed to by field CVTILK1) results in an address beyond the table end.

System Action: Processing, involving the use of the channel lookup table, discontinues.

Programmer Response: None.

BLS19010I **INVALID UCB LOOKUP TABLE ENTRY**

Explanation: An active entry in the UCB lookup table (pointed to by field CVTILK2) is either less than X'1008' or equal to X'FFFF'.

System Action: Processing, involving the use of the UCB lookup table, discontinues.

Programmer Response: None.

BLS19011I **INVALID CONTROL UNIT LOOKUP TABLE ENTRY**

Explanation: An entry in the control unit lookup table (the second part of the channel lookup table pointed to by field CVTILK1) results in a UCB lookup table offset value greater than the length of the UCB lookup table.

System Action: Processing, involving the use of the control unit lookup table, discontinues.

Programmer Response: None.

BLS19012I **RITFPFTE GREATER THAN RITLPFTE**

Explanation: A page frame table (PFT) contained a first PFTE address that was greater than the last PFTE address. This inconsistency indicates that this PFT should not be used.

System Action: The PFT control block fails validity check.

Programmer Response: Investigate the bad PFT to determine if this is the major problem in the dump.

BLS20 Messages

BLS20000I **GETMAIN FAILURE. SCALAR SUBSTITUTED**

Explanation: The user specified an array by using the DIMENSION, ENTRIES, or MULTIPLE operands. The elements contained were so large that virtual storage could not be obtained to display the data in array format.

System Action: Error condition; IPCS converts the request to a SCALAR request for the same data.

Programmer Response: Respecify the subcommand with a smaller array element size or logon with a larger region size.

BLS21 Messages

BLS21002I **COMMAND NOT EXECUTABLE**

Explanation: IPCS could not process the command because the module, which performs the processing of the command, was marked unexecutable.

System Action: IPCS ignores the command.

Programmer Response: None.

Problem Determination: Table I, items 9b, and 13.

BLS21003I **COMMAND NOT FOUND**

Explanation: IPCS processing did not find the entered command name in the list of valid commands or there was an error in the IPCS build directory list (BLDL) lookup.

System Action: None.

Programmer Response: Enter a valid command.

Problem Determination: Table I, items 9b, and 13.

BLS21006I ***xxx*** **command not supported**

Explanation: Command *xxx* is not supported in the current environment. This message can appear when issuing the TEST command from within IPCS or when issuing the GO subcommand when not currently stopped by a trap set by the TRAPON subcommand.

System Action: The system does not process the command.

Programmer Response: None.

BLS21007I to BLS21032I

BLS21007I **Unsupported command type**

Explanation: IPCS processing encountered an error in the IPCS command table.

System Action: None.

Programmer Response: None.

Problem Determination: Table I, items 7c, and 13.

BLS21020I **FILE(IPCSDDIR) NOT ALLOCATED**

Explanation: The user failed to allocate the specified file name to a VSAM cluster prior to entry into IPCS.

System Action: The IPCS command terminates.

Programmer Response: Allocate the IPCSDDIR file name to a VSAM dump directory and reenter the IPCS command.

Problem Determination: Table I, items 1, 3, 4, and 25a.

BLS21021I **DATA SET ORGANIZATION IS NOT VSAM FOR FILE(IPCSDDIR)**

Explanation: The data set allocated to the ddname, IPCSDDIR, is not a VSAM file.

System Action: The IPCS command terminates.

Programmer Response: Allocate the IPCSDDIR file name to a VSAM dump directory and reenter the IPCS command.

Problem Determination: Table I, items 1, 3, 4, and 25a.

BLS21022I **UNABLE TO ACCESS FILE(IPCSDDIR) - mac ERROR**

Explanation: While attempting to add or replace a record in the VSAM dump directory, IPCS detected an error in the use of named macro **mac**.

System Action: The IPCS command or the IPCS subcommand for which this operation is being performed terminates.

Programmer Response: None.

Problem Determination: Table I, items 1, 3, 4, and 25a.

BLS21024I **Unable to OPEN FILE(IPCSDDIR) - GENCB error**

Explanation: During the OPEN subcommand processing of the VSAM dump directory IPCS detected an error in the use of the GENCB macro.

System Action: The IPCS command terminates.

Programmer Response: None.

Problem Determination: Table I, items 1, 3, 4, and 16.

BLS21025D **DSPL3270 suspended--press enter to resume or enter a subcommand**

Explanation: The specified subcommand completed execution. DSPL3270 processing waits for a user response.

System Action: If the ENTER key is pressed, DSPL3270 will resume operation. If a subcommand is entered, it will be processed and DSPL3270 will remain suspended.

Programmer Response: Respond appropriately to the request.

BLS21030I **Print routing rejected--FILE(IPCSPRNT) not available**

Explanation: The user specified the PRINT keyword on an IPCS subcommand but no print file is available for the remainder of this IPCS session.

System Action: No print file routing will be performed.

Programmer Response: If print file output is required, allocate the print file to a suitably sized print data set, and reissue the subcommand requesting printed output.

Otherwise, if print file output is not needed, proceed with the IPCS session.

BLS21031I **Unable to open PRINT/TOC file** *ddname*

Explanation: The OPEN subcommand processing issued a nonzero return code. The most probable reason is a failure to pre-allocate a print file. *ddname* could not be opened.

System Action: The print or TOC (table of contents) file remains closed.

Programmer Response: Allocate a print file prior to invoking IPCS.

BLS21032I **Invalid BLDL parameter list**

Explanation: An IPCS module used an invalid IPCS build directory list (BLDL) service.

System Action: No BLDL service will be performed.

Programmer Response: None.

Problem Determination: Table I, items 9b, and 13.

BLS21033I *service* **return code** *retcode*

Explanation: *service* failed and passed back *retcode*. The system issues this message for BLDL and IDENTIFY services.

System Action: Use of the requested module will be bypassed.

Programmer Response: To have write-to-programmer messages written at the TSO terminal enter the TSO command

```
TSO PROFILE WTPMSG MSGID
```

and then re-execute the subcommand. If the problem recurs, the system will produce a diagnostic message for guidance in problem determination. Since IPCS ALSO HAS A PROFILE subcommand, it is necessary to TYPE TSO in front of PROFILE when you are already running under IPCS.

Problem Determination: Table I, items 9b, and 13.

BLS21034I **Inconsistent BLDL data--BLDL table rebuilt**

Explanation: Separate BLDL operations indicated changes to the contents of the load module libraries since the beginning of the IPCS session.

System Action: IPCS rebuilds the entire IPCS BLDL table to reflect current information.

Programmer Response: None.

BLS21035I **MEMBER NAME NOT SUPPORTED**

Explanation: The name of a task library, designated using the TASKLIB keyword on the IPCS command, may not be entered with a member name.

System Action: If the NOPROMPT option of the TSO PROFILE command is in effect, a correct TASKLIB data set name will be solicited. Otherwise, the IPCS session will be terminated.

Programmer Response: None.

BLS21036I **DATA SET NAME TOO LONG**

Explanation: A task library data set name was entered without enclosing apostrophes, indicating that IPCS should supply the USERID prefix at the beginning of the data set name and the final qualifier "LOAD" at the end of the data set name. However, too many characters were entered to permit the qualifiers to be added.

System Action: If the NOPROMPT option of the TSO PROFILE command is in effect, a correct TASKLIB data set name will be solicited. Otherwise, the IPCS session will be terminated.

Programmer Response: None.

BLS21038I **TASKLIB DATA SETS MUST HAVE PARTITIONED ORGANIZATION**

Explanation: A task library data set name was entered that was the name of a data set whose organization was not partitioned. Task library data sets must be partitioned.

System Action: If the NOPROMPT option of the TSO PROFILE command is in effect, a correct TASKLIB data set name will be solicited. Otherwise, the IPCS session will be terminated.

Programmer Response: None.

BLS21039I **UNABLE TO OPEN TASK LIBRARY**

Explanation: IPCS was unable to prepare the task library designated on the IPCS command for use.

System Action: The IPCS session will be terminated.

Programmer Response: To have write-to-programmer messages written at the TSO terminal enter the command

```
PROFILE WTPMSG MSGID
```

and then re-execute the subcommand. If the problem recurs, the system will produce a diagnostic message for guidance in problem determination.

BLS21043I **INVALID PRINT FILE DCB PARAMETER(S)**

Explanation: IPCS processing encountered invalid or inconsistent data control block (DCB) parameters while attempting to open the print file. Either:

- The logical record size (LRECL) was not within allowable range.
- The blocksize was less than LRECL+4.

System Action: The print file remains closed.

Programmer Response: Determine which DCB parameters are invalid and correct them.

BLS21044I **INVALID PRINT FILE PAGE SIZE**

Explanation: The specified size of the page for the print file is not within the allowed range.

System Action: The print file remains closed.

Programmer Response: Correct the size of the page for the print file, which is specified within the active SYS1.PARMLIB member.

BLS21045I to BLS21056I

BLS21045I *filename* **file is no longer available**

Explanation: The specified file name (PRINT or TOC) is no longer available for use. The most probable reason for this message is the data set is full.

System Action: IPCS ignores the request for printed output or for a TOC entry.

Programmer Response: If print or TOC file output is required, perform the following:

1. Terminate the IPCS session.
2. Allocate the print or TOC file to a suitably sized print data set.
3. Initiate a new IPCS session.

Otherwise, if print file output is not needed, proceed with the IPCS session.

BLS21046I **PRINT file** *ddname* **already open**

Explanation: An explicit user request to OPEN the print file was issued, and the print file was already open.

System Action: Processing continues.

Programmer Response: None.

BLS21050I **Table of contents parameter is not valid:** *reason*

Explanation: The interface to the Table of Contents exit service is not correctly set up by the calling exit program. *reason* may be one of the following.

1. length = 0
2. length > 40
3. undefined code
4. text is blank.

System Action: For undefined codes, "UNDEFINED TOC CODE" is placed in the table of contents entry. For other reasons, the buffer contents are used as is, but limited to 40 bytes.

Programmer Response: Correct the code in the exit program.

BLS21051I **UNABLE TO CREATE AN RPL**

Explanation: An attempt to create a request parameter list (RPL) for accessing the VSAM cluster allocated to FILE(IPCSDDIR) failed.

System Action: IPCS terminates processing for which the RPL was required.

Programmer Response: None.

Problem Determination: Table I, items 7, and 13.

BLS21052I **UNABLE TO ACCESS FILE(IPCSDDIR) - ERROR MODIFYING AN RPL**

Explanation: An attempt to modify a request parameter list (RPL) for accessing the VSAM cluster allocated to FILE(IPCSDDIR) failed.

System Action: IPCS terminates the use of the RPL.

Programmer Response: None.

Problem Determination: Table I, items 7, and 13.

BLS21053I **RECURSIVE USE OF IPCS IS NOT SUPPORTED**

Explanation: The user entered the IPCS command but IPCS is already in use in the address space. FILE(IPCSDDIR) and, optionally, FILE(IPCSPRNT) are in use.

System Action: Terminates the second invocation of the IPCS command.

Programmer Response: None.

BLS21054I **ERROR ATTACHING pgm**

Explanation: Task management detected an error during ATTACH processing.

System Action: IPCS terminates.

Programmer Response: To have write-to-programmer messages written at the TSO terminal enter the command

```
PROFILE WTPMSG MSGID
```

and then re-execute the subcommand. If the problem recurs, the system will produce a diagnostic message for guidance in problem determination.

BLS21055I **COMMAND SCAN ERROR**

Explanation: The TSO IKJSCAN service routine produced unexpected results.

System Action: Processing of the current TSO subcommand terminates.

Programmer Response: None.

Problem Determination: Table I, items 7, and 13.

BLS21056I **INVALID COMMAND SYNTAX**

Explanation: The entered command did not begin with a valid command name, which is a 1 to 8 character identifier. The entered command may optionally be preceded by a percent sign.

System Action: IPCS ignores the command.

Programmer Response: Enter a valid command.

BLS21058I **TSO SUBCOMMAND DOES NOT SUPPORT CLISTS**

Explanation: The percent sign (%) cannot be used as a prefix to a TSO command.

System Action: Processing of the current TSO subcommand terminates.

Programmer Response: None.

BLS21060I **PRINT file not open**

Explanation: An explicit request to CLOSE the print file was issued by the user, and the print file was not open.

System Action: Processing continues.

Programmer Response: None.

BLS21061I **BUFFERS NOT WRITTEN TO FILE(IPCSDDIR)**

Explanation: The VSAM WRTBFR macro returned a non-zero return code while attempting to ensure the integrity of the dump directory.

System Action: The IPCS command or the IPCS subcommand for which this operation is being performed terminates.

Programmer Response: None.

Problem Determination: Table I, items 1, 3, 4, and 25a.

BLS21062I **A directory is required, but it is not available**

Explanation: An IPCS subcommand, which requires access to a problem directory or a data set directory, cannot be used during a session started without identifying both directories to IPCS.

System Action: The command is not processed.

Programmer Response: Terminate the current IPCS session and initiate a session specifying PARM(nn). Ensure that the PARMLIB member that you reference via the PARM(nn) options designates the problem directory and the data set directory that you want to use. Then reenter the subcommand.

BLS21066I **Dump directory must be a VSAM cluster**

Explanation: The IPCS dump directory allocated as FILE(IPCSDDIR) must be a VSAM CLUSTER.

System Action: The IPCS session terminates.

Programmer Response: Allocate FILE(IPCSDDIR) to a VSAM CLUSTER, which has been defined and prepared for use as an IPCS dump directory.

BLS21067I **Dump directory relative key position must be zero**

Explanation: The IPCS dump directory must be defined with a relative key position equal to zero.

System Action: The IPCS session terminates.

Programmer Response: Allocate FILE(IPCSDDIR) to a VSAM CLUSTER, which has been defined and prepared for use as an IPCS dump directory.

BLS21068I **Dump directory key length must be 128**

Explanation: The IPCS dump directory must be defined with a key length of 128.

System Action: The IPCS session terminates.

Programmer Response: Allocate FILE(IPCSDDIR) to a VSAM CLUSTER, which has been defined and prepared for use as an IPCS dump directory.

BLS21069I **UNABLE TO LOAD ENTRY POINT mod**

Explanation: An attempt to use the contents supervision LOAD service to access the entry point for module **mod** failed.

System Action: Processing, which is dependent upon the designated program, terminates.

Programmer Response: The contents supervisor makes more detailed information regarding this situation available via the write-to-programmer service. To have write-to-programmer messages written at the TSO terminal enter the command

```
PROFILE WTPMSG MSGID
```

Take the problem determination action appropriate for the contents supervisor message.

BLS21070I **ENTRY POINT mod NOT AVAILABLE**

Explanation: An attempt to access the entry point for module **mod** failed. (A prior attempt to access the same module during the current session caused message BLS21069I to be produced.)

System Action: Processing, which is dependent upon the designated program, terminates.

Programmer Response: See the discussion of message BLS21069I.

BLS21071I **UNABLE TO CREATE AN ACB**

Explanation: An attempt to create an access method control block failed.

System Action: The IPCS command or the IPCS subcommand for which this operation is being performed terminates.

Programmer Response: None.

Problem Determination: Table I, items 1, 3, 4, and 25a.

BLS21072I **IKJSTCK ERROR rc**

Explanation: The IKJSTCK TSO service routine returned unexpected return code **rc** in response to an IPCS request.

System Action: The request that caused IPCS to use the IKJSTCK service is terminated.

Programmer Response: None.

Problem Determination: Table I, items 1, 3, 4, and 25a.

BLS21073I **CLIST is not current**

Explanation: The CLIST keyword option was used on an IPCS subcommand when no CLIST was active.

System Action: The subcommand is terminated.

Programmer Response: None.

BLS21074I **CLIST variable *name* not stored. It is {RESTRICTED | A LABEL | GLOBAL}**

Explanation: The CLIST keyword option on an IPCS subcommand designated an ineligible CLIST variable.

System Action: The subcommand is terminated.

Programmer Response: None.

BLS21075I **No data sets closed**

Explanation: The CLOSE subcommand completed processing without releasing IPCS's use of any of the data sets.

System Action: The CLOSE subcommand generates a return code of 12 and IPCS subcommand processing continues.

Programmer Response: If the CLOSE subcommand was used simply to ensure that dump sources were not being used, continue the session. Otherwise, reissue the CLOSE subcommand, specifying the dump source IPCS is to release.

Problem Determination: Table I, items 1, 3, 4, and 25a.

BLS21076I **No data sets opened**

Explanation: An OPEN subcommand request did not cause IPCS to prepare any data sets for use.

System Action: Processing continues.

Programmer Response: Review all messages generated during the processing of the OPEN subcommand. If the data sets required for your processing are not ready for use, eliminate any impediments to successful OPEN processing, then enter another OPEN subcommand.

Problem Determination: Table I, items 1, 3, 4, and 25a.

BLS21077I **{ACTIVE| DSNAME(*dsname*) |FILE(*ddname*)| PRINT} is not open**

Explanation: The CLOSE subcommand attempted to close a dump source or output data set, but the data set had not been opened.

System Action: The CLOSE subcommand generates a return code of 4 and continues processing any remaining requests.

Programmer Response: If the CLOSE subcommand was used simply to ensure that dump sources were not being used, continue the session. Otherwise, reissue the CLOSE subcommand, specifying the data set IPCS is to release.

Problem Determination: Table I, items 1, 3, 4, and 25a.

BLS21078I **{ACTIVE| DSNAME(*dsname*)| FILE(*ddname*)} is already open**

Explanation: The OPEN subcommand attempted to open a dump source, but the dump source had been previously opened.

System Action: The OPEN subcommand generates a return code of 4 and continues processing any remaining requests.

Programmer Response: If the OPEN subcommand was used simply to ensure that dump sources were ready for use, continue the session. Otherwise, reissue the OPEN subcommand, specifying the dump source IPCS is to access.

Problem Determination: Table I, items 1, 3, 4, and 25a.

BLS21080I **Command terminated by attention request**

Explanation: The current subcommand was terminated because the IPCS user pressed the attention key.

System Action: The subcommand terminates.

Programmer Response: None.

BLS21081I **device-number, device-type, DDname, operation-attempted, error-description, block-address, access-method**

Explanation: This message provides details regarding the I/O error identified by a preceding BLS21082I message. Details regarding each of the values formatted in this message is provided as part of the discussion of the SYNADAF data management macro-instruction.

System Action: The I/O operation is terminated.

Programmer Response: None.

BLS21082I **{ DSNAME** *dsn* **\| FILE** *ddn* **} I/O error**

Explanation: While processing *dsn* or *ddn*, the system detected an error.

System Action: The system issues message BLS21081I which provides details pertaining to the error.

Programmer Response: None.

BLS21083I **FILE(IPCSDDIR)** *expansion suspended*

Explanation: During an IPCS session, an attempt to expand the space allocated to FILE(IPCSDDIR) failed, and dump directory data may have been lost. IPCS stops processing requests that might repeat the failed attempt to expand FILE(IPCSDDIR). If IPCS detects one of these requests, it issues messages BLS21084I and BLS21085I and attempts to terminate the subcommand.

The session continues allowing the user to utilize any of the following IPCS facilities: LISTDUMP, LISTMAP, LISTSYM, DROPDUMP, DROPMAP, and DROPSYM.

IPCS will restore the option to expand FILE(IPCSDDIR) when you use the DROPDUMP, DROPMAP, or DROPSYM subcommands to remove any record in FILE(IPCSDDIR). IPCS will also restore the option to expand FILE(IPCSDDIR) when you terminate the current session and initiate a new one.

System Action: IPCS stops processing requests that might repeat the failed attempt to expand FILE(IPCSDDIR).

Programmer Response: You may choose either one of the following two options:

1. To increase the available space in your dump directory, end the IPCS session and issue the IDCAMS ALTER command. Then issue the IDCAMS DEFINE command to create a new, larger dump directory. Finally, issue the IDCAMS REPRO command to copy the records from the current directory to the new one.
2. To use the available space, use the IPCS LISTDUMP, LISTMAP, and LISTSYM subcommands to determine whether the dump directory contains expendable information. Then issue the IPCS DROPDUMP, DROPMAP, and DROPSYM subcommands to remove that information. You can reorganize the dump directory:

   a. End the IPCS session.

   b. Copy the contents of the dump directory to another data set using IDCAMS REPRO.

   c. Reload the dump directory using IDCAMS REPRO with the REUSE option.

   It is important to keep in mind that while deleting records in a dump directory creates available space, this space may not be where you will need it.

BLS21084I *subcommand-name* **terminated - No new FILE(IPCSDDIR) records may be created**

Explanation: In an attempt to expand FILE(IPCSDDIR) during an IPCS session *subcommand-name* failed.

System Action: The system does not process the request to store data in FILE(IPCSDDIR), and terminates the failing subcommand *subcommand-name*.

Programmer Response: The system issues message BLS21083I. Follow the information contained in this message to correct the error condition.

BLS21085I *subcommand-name* **terminated - No FILE(IPCSDDIR) {records may be lengthened \| record keys may be updated }**

Explanation: In an attempt to expand FILE(IPCSDDIR), during an IPCS session, *subcommand-name* failed.

System Action: The system does not process the request to store data in FILE(IPCSDDIR), and terminates the failing subcommand *subcommand-name*.

Programmer Response: The system issues message BLS21083I. Follow the information contained in this message to correct the error condition.

BLS21100I **Output is incomplete. PPR2 passed by exit** *exitname* **is not valid.** *reason*

Explanation: The Expanded Print service was called with a parameter list (mapped by BLSUPPR2) that failed a validity check. *exitname* passed the bad PPR2. *reason* may be one of the following error descriptions:

NO PPR2 ACRONYM.
The passed parameter list did not contain the PPR2 identifier. Verify that the exit has generated a good print parameter list and that it is initialized properly.

ADDRESS OR LENGTH OF PRINT DATA IS ZERO.
The address of the print buffer or the length of the print buffer is zero. These two fields are required unless the caller is performing a simple page eject or cancelling a conditional header.

CONDITIONAL HEADER EXCEEDS 250 BYTES.
The caller requested a conditional header to be saved. The maximum size of a conditional header is 250 bytes.

System Action: The system terminates the print request.

Programmer Response: Correct the code that calls the Expanded Print service so that it specifies a valid request.

Problem Determination: When running in an IPCS environment, consider using the TRAPON function to view the parameter list on entry to the Expanded Print service.

BLS22 Messages

BLS22001I ASVTMAXU is greater than 32767

Explanation: ASVTMAXU is a field in the address space vector table which identifies the maximum number of address spaces allowed in a given system. The number of address spaces cannot exceed 32767 (x'7FFF') because most MVS programs store the ASID is a signed halfword. The most likely cause of this error is a storage overlay of the ASVT.

System Action: Validity checking terminates for this control block.

Programmer Response: None.

BLS22002I ASVTENTY (*nnn*) = *adr*. It may be damaged

Explanation: Validation failed for the address space control block (ASCB) at address **adr**. It was located by the ASVTENTY field of the address space vector table (ASVT) for address space **nnn**.

System Action: Validity checking continues for this control block.

Programmer Response: None.

BLS22003I Address of asvt is less than address of common

Explanation: The address space vector table (ASVT) is normally located in the system common area. The pointer must be invalid.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22004I ASVTASVT NOT EQUAL C'ASVT'

Explanation: The ASVTASVT field in the address space vector table (ASVT) has an invalid acronym identifier.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22007I RBFTP is not recognizable

Explanation: The RBFTP field does not indicate a valid type of request block (RB).

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22008I RBSIZE times 8 is less than 128

Explanation: The smallest possible request block (RB) is 128 bytes.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22009I RB type and RB length fields are inconsistent

Explanation: The RB length field indicates an RB with a length that does not match the expected length for this RB type.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22012I UCBID NOT EQUAL X'FF'

Explanation: The identification field of the unit control block (UCB) is invalid.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22018I ASCB is not in the common area

Explanation: An address space control block (ASCB) must be in the system common area. The pointer must be invalid. See message BLS18058I for a description of the control block in error.

System Action: Severe condition; no further validity checking is done on this control block.

Programmer Response: None.

BLS22020I ASCBASCB NOT EQUAL C'ASCB'

Explanation: The ASCBASCB field in the address space control block (ASCB) has an invalid acronym identifier.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22021I ASCPCPUS is greater than 16

Explanation: No hardware currently supports more than sixteen CPUs. See message BLS18058I for a description of the control block in error.

System Action: Severe condition; no further validity checking is done on this control block.

Programmer Response: None.

BLS22022I ASCBLOCK does not contain a valid value

Explanation: ASCBLOCK can have any of the following values:

- 0
- 4FFFFFFF
- 7FFFFFFF
- FFFFFFFF
- 00000040 to 0000004F

If ASCBLOCK contains any value other that those allowed, the ASCB is recorded as having failed validity check. See message BLS18058I for a description of the control block in error.

System Action: Severe condition. No further validity checking is done on this control block.

Programmer Response: None.

Problem Determination: Determine the contents of the ASCBLOCK field by listing the ASCB with the LIST subcommand. From the contents of the ASCBLOCK field, try to determine which program overlayed this section of the ASCB.

BLS22029I LCCALCCA NOT EQUAL C'LCCA'

Explanation: The LCCALCCA field in the logical configuration communication area (LCCA) has an invalid acronym identifier.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22033I CVT address should not be less than x'00001000'

Explanation: The communications vector table (CVT) cannot be in the prefix storage area (PSA).

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22036I CVTTCBP does not contain x'00000218'

Explanation: The CVTTCBP field in the communications vector table (CVT) is invalid. It should contain the hexadecimal address 218.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22037I CVTEXIT does not contain X'0A03'

Explanation: The CVTEXIT field in the communications vector table (CVT) is invalid. It should contain the SVC EXIT instruction.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22038I CVTBRET does not contain X'07FE'

Explanation: The CVTBRET field in the communications vector table (CVT) is invalid. It should contain a return instruction using register 14.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22039I CVT4MS1 does not contain B'1'

Explanation: Flag field CVT4MS1 in the communications vector table (CVT) was not set to a '1' value. The operating system from which the dump was taken was not a OS/VS2 system.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22040I to BLS22085I

BLS22040I **CVT6DAT does not contain B'1'**

Explanation: Flag field CVT6DAT in the communications vector table (CVT) does not contain a '1' value. This CPU does not contain dynamic address translation.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22041I **CVTMVS2 does not contain B'1'**

Explanation: Flag field CVTMVS2 in the communications vector table (CVT) does not contain a '1' value. This is not an MVS operating system.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22042I **CVTQABST does not contain X'0A0D'**

Explanation: The CVTQABST field in the communications vector table (CVT) is invalid. It should contain the SVC ABEND instruction.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22043I **CVTLNKSC does not contain X'0A06'**

Explanation: The CVTLNKSC field in the communications vector table (CVT) is invalid. It should contain a SVC LINK instruction.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22052I **Address of TCB is less than 4096**

Explanation: The specified address is too low for a task control block (TCB), which should be located in the local system queue area (LSQA) of the private area.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22054I **TCBTCBID NOT EQUAL C'TCB'**

Explanation: The TCBTCBID field in the task control block (TCB) has an invalid acronym identifier.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22055I **TCBZERO not equal B'0000'**

Explanation: The TCBZERO field contains a nonzero value.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22073I **PCCAPCCA NOT EQUAL C'PCCA'**

Explanation: The PCCAPCCA field in the physical configuration communication area (PCCA) has an invalid acronym identifier.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22080I **PGTE(*asid*) contains nonzero reserved bits**

Explanation: A page table entry contains non-zero reserved flag bits which may be an indication of a PGTE storage overlay.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22083I **PSASTOR not a valid MVS segment table designator**

Explanation: The prefix storage area (PSA) should be located on a page boundary.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22084I **PSAPSA NOT EQUAL C'PSA'**

Explanation: The PSAPSA field in the prefix storage area (PSA) has an invalid acronym identifier.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22085I **FLCCVT not equal FLCCVT2**

Explanation: The content of the FLCCVT field is not equal to the content of the FLCCVT2 field.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22107I **SGTE(*segnum*) contains a zero page table address**

Explanation: IPCS processing found the address of a segment table entry (SGTE) to be lower than the address of the private area.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22111I **XTLNRFAC not equal 1**

Explanation: The XTLNRFAC (relocation factor) field should be equal to 1.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22112I **Common area XTLST describes private area module**

Explanation: A common area extent list (XTLST) described a private area module. The XTLST should only describe a common area module.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22114I **XTLMSBLN greater than X'13FFFF'**

Explanation: The length value in the extent list main storage block (XTLMSBLN) should not exceed the value X'13FFFF'.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22117I **ASXBASXB NOT EQUAL C'ASXB'**

Explanation: The ASXBASXB field in the address space extension block (ASXB) has an invalid acronym identifier.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22118I **ASXBTCBS greater than 100**

Explanation: IPCS processing found the value of the ASXBTCBS field to be larger than 100.

System Action: Warning condition. Validity checking continues for this control block.

Programmer Response: None.

BLS22141I **RTCT NAME NOT EQUAL C'RTCT'**

Explanation: The RTCTNAME field in the RTM recovery termination control block (RTCT) has an invalid acronym identifier.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22142I **CVTXTNT2 address should not be less than x'00001000'**

Explanation: The CVTXTNT2 field (OS/VS1-OS/VS2 Common CVT Extension) cannot be in the prefix storage area (PSA).

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22144I **CVTNUCLS contains neither a letter nor a decimal digit**

Explanation: The CVTNUCLS field in the communications vector table (CVT) is invalid.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22148I **{ASTEATO *nnn* contains a zero authorization table address**
ASTEATO *nnn* contains non-zero reserved bits
ASTEATL *nnn* contains non-zero reserved bits
ASTESTD *nnn* contains a zero segment table address }

Explanation: The problem described has been detected in an ASN-SECOND-TABLE.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22151I **GVTID NOT EQUAL C'GVT '**

Explanation: The GVTID field in the global resource serialization vector table (ISGGVT) has an invalid acronym identifier.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22155I to BLS22306I

BLS22155I **GVTXID NOT EQUAL C'GVTX'**

Explanation: The GVTXID field in the global resource serialization vector table extension (ISGGVTX) has an invalid acronym identifier.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22159I **QCBRNAML greater than 255**

Explanation: MVS ENQ/DEQ limits minor resource names to a length of 255 characters.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22167I **QHTID is not valid**

Explanation: The QHTID field should contain either C'GQHT' to represent global resource serialization global queue hash table (ISGQHTG) or C'LQHT' to represent global resource serialization local queue hash table (ISGQHTL).

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22170I **RSVID NOT EQUAL C'RSV '**

Explanation: The RSVID field in the global resource serialization ring status vector (ISGGRSV) has an invalid acronym identifier.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22187I **INCONSISTENT DESCRIPTION OF UCMLIST**

Explanation: Fields UCMVEA and UCMVEL should address the first and last UCMLIST entries, respectively, and field UCMVEZ should indicate the length of each array entry. However, either UCMVEL contains a smaller address than UCMVEA, or UCMVEA and UCMVEL differ by an amount which cannot contain an integral number of UCMLIST entries, each of which is UCMVEZ bytes in length.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22189I **UCMCBID NOT EQUAL C'UCM '**

Explanation: The UCMCBID field in the unit control module (UCM) has an invalid acronym identifier.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22192I **INVALID REPLY IDENTIFIER**

Explanation: The OREID field in the operator reply element (ORE) contains invalid data. This field should contain two EBCDIC-decimal digits.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22303I **CSD not in common**

Explanation: The common system data area (CSD) should reside in the common area.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22304I **CSDCSD NOT EQUAL C'CSD'**

Explanation: The CSDCSD field in the common system data area (CSD) has an invalid acronym identifier.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22305I **CSDCHAD OUTSIDE RANGE 1:256**

Explanation: The CSDCHAD field in the common system data area (CSD) is invalid.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22306I **CSDCPUOL outside range 1:4**

Explanation: The CSDCPUOL field in the common system data area (CSD) is invalid.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22307I **CSDGDCC outside range 0:4**

Explanation: The CSDGDCC field in the common system data area (CSD) is invalid.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22308I **CSDGDINT outside range 0:4**

Explanation: The CSDGDINT field in the common system data area (CSD) is invalid.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22309I **CSDGDTOD outside range 0:4**

Explanation: The CSDGDTOD field in the common system data area (CSD) is invalid.

System Action: Severe condition. Validity checking terminates for this control block.

Programmer Response: None.

BLS22322I **ADDR(GDA) < ADDR(COMMON)**

Explanation: The address of the GDA was below the address of common storage.

System Action: This is a severe condition and terminates the validity checking of the GDA.

Programmer Response: None.

BLS22327I **{AFTE(*nnn*) contains a zero ASN-SECOND-TABLE address | AFTE(*nnn*) contains non-zero reserved bits }**

Explanation: The problem described has been detected in an ASN-FIRST-TABLE.

System Action: This is a severe error condition. The ASN-FIRST-TABLE is marked as not valid.

Programmer Response: None.

BLS22328I **ADDR(LDA) less than ADDR(PRIVATEX)**

Explanation: The LDA is not in extended private storage.

System Action: This is a severe error condition. The LDA is marked as not valid.

Programmer Response: None.

Real Storage Manager Messages (IAR)

Component Name	RSM
Program Producing Message	Real Storage Manager (RSM)
Audience and Where Produced	For the user of IPCS: at the terminal and at the IPCS print file for some designated messages.
Message Format	IARnnnnnt text IAR Component identifier nnnnn Message serial number. t Type code: A Action; the user must perform a specific action. D Decision; the user must choose an alternative. E Error; the user must correct the error before continuing. I Information; no user action is required. text Message text.
Associated and Referenced Publications	• *MVS/XA Data Administration: Macro Instruction Reference*, GC26-4014 • *MVS/XA Integrated Catalog Administration: Access Method Services Reference*, GC26-4019 • *MVS/XA Interactive Problem Control System (IPCS) Planning and Customization*, GC28-1406 • *MVS/XA Interactive Problem Control System (IPCS) User's Guide*, GC28-1407 • *MVS/XA Interactive Problem Control System (IPCS) Command Reference*, GC28-1408 • *MVS/XA Message Library: System Codes*, GC28-1157 • *MVS/XA SPL: Service Aids*, GC28-1159 • *MVS/XA SPL: System Macros and Facilities*, Volumes 1 and 2, GC28-1150, GC28-1151 • *MVS/XA SPL: System Modifications*, GC28-1152 • *MVS/XA TSO Guide to Writing a Terminal Monitor Program (TMP) or a Command Processor*, GC28-1295 • *MVS/XA TSO Terminal Messages*, GC38-1046 • *MVS/XA VSAM Catalog Administration: Access Method Services Reference*, GC26-4075

IAR10001I **----> THE CVT AT LOCATION location COULD NOT BE ACCESSED, NO FURTHER DUMP PROCESSING ATTEMPTED**

Explanation: The CVT was inaccessible from the dump because it was not dumped or the storage access service routine was unable to access it.

System Action: When the CVT cannot be accessed, dump processing terminates at that point.

Programmer Response: Look at what was dumped. If part of Nucleus for the CVT was not dumped, the control block will not be processed. If you need RSMDATA, you may need to take another dump.

Problem Determination: Table I, items 16,29.

IAR10002I to IAR10006I

IAR10002I **----> THE PVT AT LOCATION location COULD NOT BE ACCESSED, DUMP PROCESSING CONTINUES**

Explanation: The PVT was inaccessible from the dump because it was not dumped or the storage access service routine was unable to access it.

System Action: When the PVT cannot be accessed, the system begins the next phase of dump processing is started.

Programmer Response: Look at what was dumped. If the part of Nucleus containing the control block was not dumped, the system will not process the control block. If you need RSMDATA, you may need to take another specifying the appropriate storage ranges.

Problem Determination: Table I, items 16,29.

IAR10003I **----> THE RCE/RIT AT LOCATION location COULD NOT BE ACCESSED, NO FURTHER DUMP PROCESSING ATTEMPTED**

Explanation: A control block was inaccessible from the dump because it was not dumped or the storage access service routine was unable to access it.

System Action: When the RCE/RIT cannot be accessed, the system terminates dump processing at location **location**.

Programmer Response: Look at what was dumped. If the part of the Extended Nucleus containing the control block was not dumped, the system will not process the control block. If you need RSMDATA, you will need to take another dump specifying the appropriate storage ranges.

Problem Determination: Table I, items 16,29.

IAR10004I **----> THE RPB BIT MAP COULD NOT BE BUILT**

Explanation: The system could not build the RPB bit map because one of the following conditions existed:

- There were no RPBs.
- The system could not access the RPB pool header because it was (1) not dumped or (2) the storage access service routine could not access the pool header.
- There was an error in the RPB pool header.

System Action: The system can not determine the number of unqueued RPBs will not be determined.

Programmer Response: Take one of the following actions:

- Look at what was dumped. If the system could not access the RPB pool header, then that part of the Extended SQA containing the RPB pool must be

  dumped in order that the system may process the pool header. If you need RSMDATA, take another dump specifying the appropriate storage ranges.
- Look in SYS1.LOGREC. A System Diagnostic Work Area is filled in when a problem occurs in RSM processing causing RSM Recovery to receive control. The SDWA contains both the status of the system at the time of the error and the Variable Recording Area. The Variable Recording area contains additional diagnostic data.
- Look at the control block data. If the control block was overlaid, then trying to find out whose data is in the control block may be helpful in locating the cause of the problem that led to the dump. The overlay may include one or more of the RPB/PCB/FCBs within the pool.

Problem Determination: Table I, items 16,18,29.

IAR10005I **----> THE ESTE AT LOCATION location COULD NOT BE ACCESSED**

IAR10005I **----> THERE ARE nn ESTES AT LOCATION location THAT COULD NOT BE ACCESSED**

IAR10006I **----> THE MPE AT LOCATION location COULD NOT BE ACCESSED**

Explanation: The dump did not contain the **location** of the ESTE or MPE because either the ESTE or MPE was not dumped or the storage access service routine was unable to access it.

System Action: The action that the system takes will depend on whether the system was processing either an ESTE or MPE queue, or an Extended Store Table. If the system was processing an ESTE or MPE queue, the system will terminate processing at **location** and then start the next phase of dump processing. In the case of an Extended Store Table, table processing continues with the next two ESTEs.

Programmer Response: Look at what was dumped. If the part of Extended SQA containing the ESTE or MPE was not dumped, the system will not process the control block. If you need RSMDATA, take another dump specifying the appropriate storage ranges.

Problem Determination: Table I, items 16,29.

IAR10007I **----> THE RAB AT LOCATION location COULD NOT BE ACCESSED**

Explanation: The dump did not contain the RAB at **location** because it was either not dumped or the storage access service routine was unable to access it.

System Action: When the RAB cannot be accessed, the system terminates RAB queue processing at **location**, and starts the next phase of dump processing.

Programmer Response: Look at what was dumped. If the part of Extended Nucleus or Extended SQA containing the RAB was not dumped, the system will not process the control block. If you need RSMDATA, take another dump, specifying the appropriate storage ranges.

Problem Determination: Table I, items 16,29.

IAR10008I **----> THE RPB AT LOCATION location COULD NOT BE ACCESSED**

Explanation: The dump did not contain the RAB/PCB/FCB at **location** because it was either not

System Action: When the RPB/PCB/FCB block cannot be accessed, RPB/PCB/FCB queue processing terminates at that point and the system starts the next phase of dump processing.

Programmer Response: Look at the dump. If the part of Extended SQA containing the RPB/PCB/FCB was not dumped, the system will not process the control block. If you need RSMDATA, take another dump, specifying the appropriate storage ranges.

Problem Determination: Table I, items 16,29.

IAR10009I **----> THE PCB AT LOCATION location COULD NOT BE ACCESSED**

Explanation: The dump did not contain the RPB/PCB/FCB at **location** because it was either not dumped or the storage access service routine was unable to access it.

System Action: When the RPB/PCB/FCB block cannot be accessed, RPB/PCB/FCB queue processing terminates at that point and the system starts the next phase of dump processing.

Programmer Response: Look at the dump. If the part of Extended SQA containing the RPB/PCB/FCB was not dumped, the system will not process the control block. If you need RSMDATA, take another dump, specifying the appropriate storage ranges.

Problem Determination: Table I, items 16,29.

IAR10010I **----> THE PFTE AT LOCATION location COULD NOT BE ACCESSED**

IAR10010I **----> THERE ARE nn PFTES AT LOCATION location THAT COULD NOT BE ACCESSED**

Explanation: The dump did not contain the PFTE at **location** because it was not dumped or the storage access service routine was unable to access them.

System Action: The action that the system takes will depend on whether the system was processing a PFTE queue, or the Page Frame Table. If the system was processing a PFTE queue, the system will terminate processing at **location** and then start the next phase of dump processing. In the case of the Page Frame Table, table processing continues with the next two PFTEs.

Programmer Response: Look at the dump. If the part of Extended Nucleus containing the PFTE was not dumped, the system will not process the control block. If you need RSMDATA, take another dump, specifying the appropriate storage ranges.

Problem Determination: Table I, items 16,29.

IAR10011I **----> THE RVR AT LOCATION location COULD NOT BE ACCESSED**

Explanation: The dump did not contain the RVR at **location** because it was either not dumped or the storage access service routine was unable to access it.

System Action: When the RVR cannot be accessed, RVR queue processing terminates at that point and the system starts the next phase of dump processing.

Programmer Response: Look at the dump. If the part of E/LSQA that contained the RVR was not dumped, the system will not process the control block. If you need RSMDATA, take another dump, specifying the appropriate storage ranges.

Problem Determination: Table I, items 16,29.

IAR10012I **----> THE RPB POOL HEADER AT LOCATION location COULD NOT BE ACCESSED**

Explanation: The dump did not contain the RPB Pool Header at **location** because it was either not dumped or the storage access service routine was unable to access it.

System Action: When the Pool Header cannot be accessed, Pool Header processing terminates and the system starts the next phase of dump processing.

Programmer Response: Look at the dump. If the part of Extended SQA for that pool was not dumped, it will not be processed. If you need RSMDATA, take another dump, specifying the appropriate storage ranges.

Problem Determination: Table I, items 16,29.

IAR10013I **----> THE RSM TRACE TABLE AT LOCATION location COULD NOT BE ACCESSED**

Explanation: The dump did not contain the RSM Trace Table at **location** because it was either not dumped or the storage access service routine was unable to access it.

System Action: When the Trace Table cannot be accessed, Trace Table processing terminates and the system starts the next phase of dump processing.

Programmer Response: Look at the dump. If the part of Extended SQA for the trace table was not dumped, it will not be processed. If you need RSMDATA, take another dump, specifying the appropriate storage ranges.

Problem Determination: Table I, items 16,29.

IAR10014I **----> THE SEGMENT TABLE AT LOCATION location COULD NOT BE ACCESSED**

Explanation: The dump did not contain the Segment Table at **location** because it was either not dumped or the storage access service routine was unable to access it.

System Action: When the Segment Table cannot be accessed, table processing terminates and the system starts the next phase of dump processing.

Programmer Response: Look at the dump. If the part of E/SQA or E/LSQA that contained the Segment Table was not dumped, the system will not process the control block. be processed. If you need RSMDATA, take another dump, specifying the appropriate storage ranges.

Problem Determination: Table I, items 16,29.

IAR10015I **----> THE PGT/XPT AT LOCATION location COULD NOT BE ACCESSED**

Explanation: The dump did not contain the PGT/XPT at **location** because it was either not dumped or the storage access service routine was unable to access it.

System Action: When the PGT/XPT cannot be accessed, the table processing terminates and the system starts the next phase of dump processing.

Programmer Response: Look at the dump. If the part of E/SQA or E/LSQA that contained the PGT/XPT was not dumped, the system will not process the control block. If you need RSMDATA, take another dump, specifying the appropriate storage ranges.

Problem Determination: Table I, items 16,29.

IAR10016I **----> THE LCCAVT AT LOCATION location COULD NOT BE ACCESSED, NO RSM STACKS ARE PROCESSED**

IAR10016I **----> THE LCCA AT LOCATION location COULD NOT BE ACCESSED, NO RSM STACKS ARE PROCESSED FOR THIS CPU**

IAR10016I **----> THE WSAVT FIELD AT LOCATION location COULD NOT BE ACCESSED, NO RSM STACKS ARE PROCESSED FOR THIS CPU**

IAR10016I **----> THE RSM WORK AREA AT LOCATION location COULD NOT BE ACCESSED, NO RSM STACKS ARE PROCESSED FOR THIS CPU**

Explanation: The dump did not contain the LCCAVT, LCCA, WSAVT field or the RSM work area at **location** because it was either not dumped or the storage access service routine was unable to access it.

System Action: When the LCCAVT cannot be accessed, the system does not process the RSM stacks for any of the CPUs. The system then starts the next phase of dump processing. When the LCCA, WSAVT field, or RSM work area cannot be accessed, RSM stack processing terminates for that CPU and processing continues with the RSM stacks for the next CPU.

Programmer Response: Look at the dump. If the part of E/SQA that contained the LCCAVT, LCCA, WSAVT field, or RSM work area was not dumped, the system will not process RSM work areas. If you need RSMDATA, take another dump, specifying the appropriate storage ranges.

Problem Determination: Table I, items 16,29.

IAR10017I **----> THE PVT CONTROL BLOCK ID IS NOT CORRECT**

IAR10017I **----> THE RCE CONTROL BLOCK ID IS NOT CORRECT**

IAR10017I **----> THE RIT CONTROL BLOCK ID IS NOT CORRECT**

Explanation: The control block ID was incorrect. The PVT contains pointers to other routines and the RSM recovery tables. Incorrect data in the PVT may have caused (1) ABENDs in other routines that rely on the pointers or (2) an error in RSM recovery when it used the recovery table. The RCE/RIT contains counts, queue headers, and other data. Incorrect counts in the RCE/RIT may have affected other components that used them.

System Action: The system continues dump processing continues.

Programmer Response: You can take the following actions:

- Look in SYS1.LOGREC. A System Diagnostic Work Area is filled in when a problem occurs in RSM processing causing RSM Recovery to receive control. The

SDWA contains the (1) system status at the time of the error and (2) the Variable Recording Area, which contains additional diagnostic data.

- Look at the control block data. If the data appears to have been overlaid, try to determine whose data is there. This can help you locate the cause of the problem that led to the dump.

Problem Determination: Table I, items 16,18,29

IAR10018I **----> THE MPE AT LOCATION location IS INCORRECTLY QUEUED**

Explanation: The system issues this message when either one of two conditions is true. First, the MPE at **location** is incorrectly queued. That means the current back pointer is not equal to the previous forward pointer. Second, the ID is incorrect for the queue that the system is processing.

During system execution, the next attempt to either validate a segment for which there are MPEs or swap in an address space will terminate the address space.

System Action: The system terminates MPE queue processing and starts the next phase of dump processing.

Programmer Response: You can take the following actions:

- Look in SYS1.LOGREC. A System Diagnostic Work Area is filled in when a problem occurs in RSM processing causing RSM Recovery to receive control. The SDWA contains (1) status of the system at the time of the error and (2) the Variable Recording Area, which contains additional diagnostic data.
- Look at the control block data. If the data appears to have been overlaid, try to determine whose data is there. This will help you locate the cause of the problem that led to the dump.

IAR10019I **----> THE ESTE AT LOCATION location IS INCORRECTLY QUEUED**

Explanation: The system issues this message when either one of two conditions is true. First, the ESTE at **location** is incorrectly queued. That means the current back pointer is not equal to the previous forward pointer. Second, the ID is incorrect for the queue that the system is processing.

An incorrectly queued ESTE may result in one or both of the following conditions being true.

- The XPTE for the page was updated incorrectly.
- When referencing the page, the user received an incorrect version of the page.

System Action: The system terminates ESTE queue processing and starts the next phase of dump processing.

Programmer Response: You can take the following actions:

- Look in SYS1.LOGREC. A System Diagnostic Work Area is filled in when a problem occurs in RSM processing causing RSM Recovery to receive control. The SDWA contains (1) status of the system at the time of the error and (2) the Variable Recording Area, which contains additional diagnostic data.
- Look at the control block data. If the data appears to have been overlaid, try to determine whose data is there. This will help you locate the cause of the problem that led to the dump.

Problem Determination: Table I, items 16,18,29

IAR10020I **----> THE STORAGE AT LOCATION location SHOULD BE AN ESTE, BUT IS NOT WITHIN THE RANGE OF THE EXTENDED STORE TABLE**

Explanation: The forward pointer from the previous ESTE is incorrect. An incorrect ESTE may result in one or both of the following conditions being true:

- The XPTE was updated incorrectly.
- The page of data that the user requested was either incorrect, or caused RSM to abnormally terminate the request.

System Action: The system terminates ESTE queue processing and starts the next phase of dump processing. In the case of the Extended Store Table, table processing continues.

Programmer Response: You can take the following actions:

- Look in SYS1.LOGREC. A System Diagnostic Work Area is filled in when a problem occurs in RSM processing causing RSM Recovery to receive control. The SDWA contains (1) status of the system at the time of the error and (2) the Variable Recording Area, which contains additional diagnostic data.
- Look at the control block data. If the data appears to have been overlaid, try to determine whose data is there. This will help you locate the cause of the problem that led to the dump.

Problem Determination: Table I, items 16,18,29

IAR10021I to IAR10024I

IAR10021I **----> THE RPB POOL HEADER ID IS INCORRECT**

Explanation: The pool header ID EBCDIC string was not 'RPB POOL'.

Explanation: The pool header ID EBCDIC string was not 'RPB POOL'. An incorrect RPB Pool Header will prevent the RPB bit map in RSM Print Dump from being generated and the number of unqueued RPBs will not be able to be determined.

System Action: The system terminates RPB Poll processing and starts the next phase of dump processing.

Programmer Response: You can take the following actions:

- Look in SYS1.LOGREC. A System Diagnostic Work Area is filled in when a problem occurs in RSM processing causing RSM Recovery to receive control. The SDWA contains (1) status of the system at the time of the error and (2) the Variable Recording Area, which contains additional diagnostic data.
- Look at the control block data. If the data appears to have been overlaid, try to determine whose data is there. This will help you locate the cause of the problem that led to the dump.

Problem Determination: Table I, items 16,18,29

IAR10022I **----> THE FOLLOWING RPB IS INCORRECTLY QUEUED**

Explanation: The RPB/PCB/FCB is incorrectly queued (the current back pointer is not equal to the previous forward pointer). An incorrectly queued RPB/PCB/FCB could have caused units of work (TCBs and SSRBs) to be hung. Also, updates to the PGTE/XPTE may have been incorrect, causing an incorrect version of a page of data being given to the user.

System Action: The system terminates RPB/PCB/FCB queue processing and starts the next phase of dump processing is started.

Programmer Response: You can take the following actions:

- Look in SYS1.LOGREC. A System Diagnostic Work Area is filled in when a problem occurs in RSM processing causing RSM Recovery to receive control. The SDWA contains (1) status of the system at the time of the error and (2) the Variable Recording Area, which contains additional diagnostic data.
- Look at the control block data. If the data appears to have been overlaid, try to determine whose data is there. This will help you locate the cause of the problem that led to the dump.

Problem Determination: Table I, items 16,18,29

IAR10023I **----> THE FOLLOWING RVR IS INCORRECTLY QUEUED**

Explanation: The RVR is incorrectly queued (the current back pointer is not equal to the previous forward pointer). An incorrectly queued RVR may mean the data the user requested was not accessible. In addition, RSM may have lost track of which pages were in use as VDAC pages.

System Action: The system terminates RVR queue processing and starts the next phase of dump processing is started.

Programmer Response: You can take the following actions:

- Look in SYS1.LOGREC. A System Diagnostic Work Area is filled in when a problem occurs in RSM processing causing RSM Recovery to receive control. The SDWA contains (1) status of the system at the time of the error and (2) the Variable Recording Area, which contains additional diagnostic data.
- Look at the control block data. If the data appears to have been overlaid, try to determine whose data is there. This will help you locate the cause of the problem that led to the dump.

Problem Determination: Table I, items 16,18,29

IAR10024I **----> THE FOLLOWING PFTE IS INCORRECTLY QUEUED**

Explanation: The PFTE is incorrectly queued (the current back pointer is not equal to the previous forward pointer). An incorrect PFTE may mean RSM could have misinterpreted the state of the page and given the user an incorrect version of the page.

System Action: The system terminates PFTE PFTE queue processing and starts the next phase of dump processing.

Programmer Response: You can take the following actions:

- Look in SYS1.LOGREC. A System Diagnostic Work Area is filled in when a problem occurs in RSM processing causing RSM Recovery to receive control. The SDWA contains (1) status of the system at the time of the error and (2) the Variable Recording Area, which contains additional diagnostic data.
- Look at the control block data. If the data appears to have been overlaid, try to determine whose data is there. This will

help you locate the cause of the problem that led to the dump.

Problem Determination: Table I, items 16,18,29

IAR10025I **----> THE STORAGE AT LOCATION location SHOULD BE A PFTE, BUT IS NOT WITHIN THE RANGE OF THE PFT**

Explanation: The forward pointer from the previous PFTE is incorrect. An incorrect PFTE may mean RSM could have misinterpreted the state of the page and given the user an incorrect version of the page.

System Action: The system terminates PFTE queue processing and starts the next phase of dump processing. When a Page Frame Table is involved, the system continues page table processing.

Programmer Response: You can take the following actions:

- Look in SYS1.LOGREC. A System Diagnostic Work Area is filled in when a problem occurs in RSM processing causing RSM Recovery to receive control. The SDWA contains (1) status of the system at the time of the error and (2) the Variable Recording Area, which contains additional diagnostic data.
- Look at the control block data. If the data appears to have been overlaid, try to determine whose data is there. This will help you locate the cause of the problem that led to the dump.

Problem Determination: Table I, items 16,18,29

IAR10026I **----> THE RAB CONTROL BLOCK ID IS NOT CORRECT**

IAR10027I **----> THE FOLLOWING RAB IS INCORRECTLY QUEUED**

Explanation: The system issues this message when either the control block ID is incorrect or the RAB is incorrectly queued. The incorrect queue results when the the current back pointer is not equal to the previous forward pointer. The RAB contains the anchors for the RSM queues for an address space. During system execution, an incorrect RAB could affect the address space related processing. This could result in the user receiving an incorrect version of a page.

System Action: The system terminates RAB queue processing and starts the next phase of dump processing.

Programmer Response: You can take the following actions:

- Look in SYS1.LOGREC. A System Diagnostic Work Area is filled in when a problem occurs in RSM processing causing RSM Recovery to receive control. The SDWA contains (1) status of the system at the time of the error and (2) the Variable Recording Area, which contains additional diagnostic data.
- Look at the control block data. If the data appears to have been overlaid, try to determine whose data is there. This will help you locate the cause of the problem that led to the dump.

Problem Determination: Table I, items 16,18,29

IAR10028I **----> THERE ARE nn ERRONEOUSLY UNQUEUED PFTES**

Explanation: The PFTEs have an ID that is incorrect for an unqueued PFTE.

System Action: The system continues dump processing.

Programmer Response: The PFTE queues were damaged and there may be incorrect data for some pages in either the address space owning the frame or in the common area. Also, some real storage may not be usable and could cause excessive paging. Check the PFTEs that are listed as erroneously unqueued in the RSMDATA section of the dump.

Problem Determination: Table I, items 16,29

IAR10029I **----> THERE ARE nn SYSTEM RELATED UNQUEUED OR PERMANENTLY ASSIGNED PFTES**

Explanation: **nn** PFTEs are reserved for system use.

System Action: The system continues dump processing.

Programmer Response: None.

Problem Determination: Table I, item 16

IAR10030I **THERE ARE nn TEMPORARILY UNQUEUED PFTES**

Explanation: **nn** PFTEs are temporarily unqueued in the process of changing queues.

System Action: The system continues dump processing.

Programmer Response: None.

Problem Determination: Table I, item 16

IAR10031I to IAR10033I

IAR10031I **----> THE RAB QUEUE WAS NOT PROCESSED TO COMPLETION**

IAR10031I **----> THE RAB QUEUE WAS PROCESSED TO COMPLETION**

Explanation: The RAB queue will not be processed to completion if a RAB is not accessible or in error. An example of an error would be an incorrect ID or the current back pointer not not equal to the previous forward pointer. There is no further queue or segment table processing for that address space or any address space after it.

System Action: The system continues dump processing.

Programmer Response: Look at the RSMDATA portion of the dump to determine what caused the RAB queue processing to terminate. If the PRINT or VERIFY option was not specified with RSMDATA, run the dump again with one of those options.

Problem Determination: Table I, items 16,29

IAR10032I **----> THERE ARE nn ADDRESS SPACES IN ERROR**

Explanation: **nn** address spaces are in error. These erroneous address spaces had RABs which passed the validity checking but some queue(s) anchored in the RAB were found to be in error. There is no error checking for segment tables.

System Action: The system continues dump processing.

Programmer Response: Look at the RSMDATA portion of the dump to determine which address space was in error. If the PRINT or VERIFY option was not specified with RSMDATA, run the dump again with one of those options.

Problem Determination: Table I, items 16,29

IAR10033I **SELECTED STACKS THAT ARE OR WERE ACTIVE:**

RECOVERY STACK
RSB STACK
RSB RECOVERY STACK
MACHINE CHECK STACK

Explanation: This message provides information on the types of RSM processing that have occurred.

- If the message text contains **RECOVERY STACK**, then RSM has entered its FRR.
- If the message text contains **RSB STACK**, a disabled summary dump has been requested.
- If the message text contains **RSB RECOVERY STACK**, an error occurred in RSM while processing a disabled summary dump.
- If the message text contains **MACHINE CHECK STACK**, a machine check occurred which required RSM support.

System Action: The system continues dump processing.

Programmer Response: You may take the following actions:

- Look at any previous dumps for previously active stacks. If the message text contains **RECOVERY STACK** or **RSB RECOVERY STACK**, look at the Logrec entries.
- If the message text contains **MACHINE CHECK STACK**, check the Logrec entries for related hardware data.

Problem Determination: Table I, items 16,18,29

MVS Component Messages (IEA)

Component Name	IEA
Program Producing Message	Supervisor, LOGREC, RTM and COMMTASK
Audience and Where Produced	For the user of IPCS: at the terminal and at the IPCS print file for some designated messages.
Message Format	IEAnnnnnt text IEA Component identifier 31nnn Message serial number. t Type code: A Action; the user must perform a specific action. D Decision; the user must choose an alternative. E Error; the user must correct the error before continuing. I Information; no user action is required. text Message text.
Associated and Referenced Publications	• *MVS/XA Data Administration: Macro Instruction Reference*, GC26-4014 • *MVS/XA Integrated Catalog Administration: Access Method Services Reference*, GC26-4019 • *MVS/XA Interactive Problem Control System (IPCS) Planning and Customization*, GC28-1406 • *MVS/XA Interactive Problem Control System (IPCS) User's Guide*, GC28-1407 • *MVS/XA Interactive Problem Control System (IPCS) Command Reference*, GC28-1408 • *MVS/XA Message Library: System Codes*, GC28-1157 • *MVS/XA SPL: Service Aids*, GC28-1159 • *MVS/XA SPL: System Macros and Facilities*, Volumes 1 and 2, GC28-1150, GC28-1151 • *MVS/XA SPL: System Modifications*, GC28-1152 • *MVS/XA TSO Guide to Writing a Terminal Monitor Program (TMP) or a Command Processor*, GC28-1295 • *MVS/XA TSO Terminal Messages*, GC38-1046 • *MVS/XA VSAM Catalog Administration: Access Method Services Reference*, GC26-4075

IEA11 Messages (Supervisor)

IEA11001I **Unable to complete suspend lock analysis**

Explanation: The dump did not contain the control blocks required for suspend lock analysis. The system performed suspend lock analysis on the available storage. However the analysis may be incomplete for the entire dump. The system requires the following storage: All ASCBs, TCBs in locally locked address spaces, SRBs and SSRBs on the lock suspended queues.

System Action: The dump analysis exit bypasses processing for a given lock and continues processing on the next lock or address space.

Programmer Response: None

Problem Determination: The major cause of this message is that storage was not dumped. This message will almost always appear when processing a virtual dump.

IEA21 Messages (RTM)

IEA21001I **Address space is terminating [ab]normally [due to** *cde-rsnc*]

Explanation: The address space is in the process of either normal or abnormal termination. All work in the address space has terminated. For abnormal termination the message text contains the memory termination completion code **cde** along with an associated reason code **rsnc**.

System Action: None.

Programmer Response: Refer to *MVS/XA System Codes* for an explanation of the system completion and reason codes to determine why the address space was terminated. You can also run SUMMARY FORMAT for the address space in question.

IEA21002I **Task has terminated [ab]normally DUE TO [***cde - rsnc***]**

Explanation: The system issues this message when (1) the task terminated normally and is no longer active, or (2) the task has already terminated abnormally and all recovery routines percolated or failed. The message also contains the the TCB completion code *cde* and any associated reason codes *rsnc*.

System Action: None.

Programmer Response: Refer to the *MVS/XA System Codes* for for an explanation of the system completion and reason codes to determine why the address space was terminated. You can also browse the TCB or run SUMMARY FORMAT for the address space which contains the task in question.

IEA21003I **Task is scheduled for abend due to** *cde-rsnc* **at** *address*

Explanation: RTM1 has set up this task to issue SVC 13 (ABEND) when it is next dispatched. The message contains the TCB completion code *cde* and any associated reason codes *rsnc*. The message text also contains the PSW address *address* from RBRTPSW1 at the time the task was last interrupted.

System Action: None.

Programmer Response: Refer to *MVS/XA System Codes* for an explanation of the system completion and reason codes to determine why the address space was terminated. You can also browse the TCB or run SUMMARY FORMAT for the address space which contains the task in question.

IEA21004I **Task is terminating [ab]normally [due to** *cde-rsnc*]

Explanation: The task is in the process of normal or abnormal termination. Abnormal termination occurs when (1) all recovery routines have percolated or (2) recovery routing is still active and a non-retryable error has occurred. For a non-retryable error the TCBFA flag is set on in the TCB. The message text contains the TCB completion code *cde*, and any associated reason codes *rsnc*.

Otherwise, the task is in the process of normal termination. All PRBs have exited normally.

System Action: None.

Programmer Response: Refer to *MVS/XA System Codes* for an explanation of the system completion and reason codes to determine why the address space was terminated. You can also browse the TCB or run SUMMARY FORMAT for the address space which contains the task in question.

IEA21005I **Task is in recovery processing, LIFO summary of active recovery environments follows.**

Explanation: This is a general message to indicate that some RTM activity is in progress for the task. The system will issue message IEA21006I, which contains more specific information on the existing condition.

System Action: None.

Programmer Response: Wait for the rest of the messages to come out and read the description of those messages.

IEA21006I **In RTM1/RTM2 FOR** *cde-rsnc* **at** *pswaddr* **[(SDWA at** *sdwaaddr***)]**

Explanation: Either RTM1 or RTM2 is active. The message text contains the TCB completion code *cde* and any associated reason codes **rsnc**. If RTM2 is active, the message text also contains the PSW address *pswaddr* from the abnormally terminating RB. *pswaddr* is not printed for RTM1 processing. If the active SDWA address is available, the message text contains the SDWA address **sdwaaddr**.

System Action: None.

Programmer Response: See *MVS/XA System Codes* for an explanation of the system completion and reason codes to determine why the address space was terminated. You can also

browse the TCB or run SUMMARY FORMAT for the address space which contains the task in question.

IEA21007I **In wwwww at *enptaddr* for *cde-rsnc* at *pswaddr* [(SDWA at *sdwaaddr*)]**

Explanation: The system displays message IEA21007I with the following fields:

- *wwwww* indicates an (E)STAE or (E)STAI is active.
- *enptaddr* is the entry point address of the recovery routine.
- *cde-rsnc* is the TCB completion code and associated reason code.
- *pswaddr* is the PSW address from the abnormally terminating RB.
- *sdwaaddr* is the address of the active SDWA if the SDWA still exists.

System Action: None.

Programmer Response: See *MVS/XA System Codes* for an explanation of the system completion and reason codes to determine why the address space was terminated. You can also browse the TCB or run SUMMARY FORMAT for the address space which contains the task in question.

IEA21010I **RTM2ADDR does not point to this RTM2WA**

Explanation: The storage thought to be an RTM2 work area does not contain the address of this RTM2WA in the RTM2ADDR field.

System Action: The system does not format the RTM2WA.

Programmer Response: Browse the storage in question or check to see if the proper address and ASID were specified for the validity check exit.

IEA24 Messages (LOGREC)

IEA24001I **LOGREC buffer could not be accessed, possible cause: data not in dump.**

Explanation: The dump did not contain the control blocks necessary to locate the LOGREC buffer.

System Action: The system does not format the the LOGREC buffer entries.

Programmer Response: None.

IEA24002I **LOGREC buffer could not be formatted. header information is invalid.**

Explanation: The LOGREC buffer header is invalid.

System Action: The LOGREC buffer entries are not formatted.

Programmer Response: If you need LOGREC information, dump the buffer in hexadecimal and locate the unformatted entries manually.

IEA24003I **EREP enhancement is not installed, LOGREC entries displayed in dump format data.**

Explanation: The EREP routines used to format the LOGREC entries are not installed.

System Action: The system formats all LOGREC entries in hexadecimal and EBCDIC.

Programmer Response: You need to install the appropriate level of EREP for the release of MVS you have on your system. Consult your MVS installation requirements for the correct level of EREP.

IEA24004I **There are no LOGREC entries in the buffer.**

Explanation: There are no entries in the LOGREC buffer.

System Action: No entries are formatted.

Programmer Response: None.

IEA24005I **Some entries could not be formatted due to errors in the recording process.**

Explanation: The system could not locate some of the buffer entries due to the error recovery actions of the Recording Facility.

System Action: The system will format only those entries it can find.

Programmer Response: If you need LOGREC information, dump the buffer in hexadecimal and locate the unformatted entries manually.

IEA24006I **This entry was incomplete at the time of the dump.**

Explanation: The LOGREC entry had not been completely copied into the buffer at the time of the dump.

System Action: The entry is formatted in hexadecimal and EBCDIC.

Programmer Response: None.

IEA24007I **This entry was not buffered and may contain invalid data.**

Explanation: The data for this entry was not copied to the LOGREC buffer. The storage containing the data may have been changed.

System Action: If possible, the system formats the entry.

Programmer Response: None.

IEA24008I **EREP formatting failed for this entry. It is displayed in dump format.**

Explanation: The EREP formatting routine could not format the LOGREC entry.

System Action: The entry is formatted in hexadecimal and EBCDIC.

Programmer Response: None.

IEA24009I **Processing errors encountered in EREP formatting. remaining entries formatted as hexadecimal data.**

Explanation: The EREP formatting routine encountered severe errors when attempting to format a LOGREC entry.

System Action: The EREP routines will not be used for any further formatting. All remaining entries will be formatted in hexadecimal and EBCDIC.

Programmer Response: None.

IEA24010I **Unable to locate the next entry in the buffer.**

Explanation: The algorithm used to locate entries in the buffer produced an address outside the boundaries of the buffer.

System Action: No further entries are formatted.

Programmer Response: If you need LOGREC information, dump the buffer in hexadecimal and locate the unformatted entries manually.

IEA24011I **A non-buffered entry could not be located. processing continued with the next entry.**

Explanation: The entry within the buffer does not contain the address of the unbuffered data. The data could not be located.

System Action: The system does not format the entry.

Programmer Response: None.

IEA24012I **A non-buffered entry could not be retrieved from the dump.**

Explanation: An attempt to access the storage containing the LOGREC data failed.

System Action: The system does not format the entry.

Programmer Response: None.

IEA24050I **LOGDATA formatter completed successfully.**

Explanation: The LOGDATA verb exit returned control to IPCS/Print Dump. No errors were encountered in LOGDATA processing.

System Action: LOGDATA returns control to IPCS/Print Dump.

Programmer Response: None.

IEA24060I **LOGDATA formatter terminated due to errors.**

Explanation: The LOGDATA verb exit returned control to IPCS/Print Dump. Errors were encountered which prevented LOGDATA from properly formatting all the entries in the LOGREC buffer.

System Action: LOGDATA returns control to IPCS/Print Dump.

Programmer Response: None.

IEA31 Messages (CommTask)

IEA31001I **Number of messages queued (UCMWQNR) is** *nnnn*. **Limit (UCMWQLM) is** *xxxx* .

Explanation: The unit control module (UCM) was obtained. The UCMWQNR field contains decimal *nnnn* messages. The UCMWQLM field contains decimal *xxxx* messages.

System Action: None.

Programmer Response: None.

IEA31002I *nnn* **Major WQEs chained from UCM**

Explanation: The UCMWTOQ field of the unit control module (UCM) addresses the write-to-operator queue element (WQE). Starting with the first WQE, the system counts the the WQEs chained via their WQELKP fields. Decimal **nnn** is the total number of WQEs.

System Action: None.

Programmer Response: None.

IEA31003I **UCMSTS status flag byte is X'xx' for following console**

Explanation: In examining the unit control module device entry (UCME), the system found a nonzero UCMSTS status byte value **xx**. The system issues message IEA31004I to indicate the console associated with this UCME.

System Action: None.

Programmer Response: None.

IEA31004I ***nnn* WQEs found for console *adr***

Explanation: This message contains the following information:

- The system obtains the unit control block (UCB) for the console and the message contains the console address *adr*. This is the address of the console associated with the UCME, which is referenced in message IEA31003I.
- The system also searches the the console queue element (CQE) chain from a given unit control module device entry (UCME), and decimal *nn* is the total number of WQEs encountered.

System Action: None.

Programmer Response: None.

IEA31005I **Operator reply *nn* was outstanding**

Explanation: An operator reply element (ORE) was found with reply identifier *nn*.

System Action: None.

Programmer Response: None.

IEA31006I **Inconsistent description of UCMLIST**

Explanation: Fields UCMVEA and UCMVEL should address the first and last UCMLIST entries, respectively, and field UCMVEZ should indicate the length of each array entry. However, either UCMVEL contains a smaller address than UCMVEA, or UCMVEA and UCMVEL differ by an amount which cannot contain an integral number of UCMLIST entries, each of which is UCMVEZ bytes in length.

System Action: The system does no further validity checking on this control block.

Programmer Response: None.

IEA31007I **Reply identifier is not valid**

Explanation: Field OREID should contain two EBCDIC-decimal digits.

System Action: This is a severe condition, and the system does no further validity checking on this control block.

Programmer Response: None.

IEA31009I **This task was waiting for operator reply *rr: hh:mm:ss * rr text***

Explanation: An outstanding write to operator with reply (WTOR) was found for the requested task (TCB) with WTOR number *rr*. *hh::mm:ss* is the time that the system issued the message. The message also contains the *text* of the operator message.

System Action: None.

Programmer Response: None.

Master Scheduler Messages (IEE)

Component Name	IEE
Program Producing Messages	Master Scheduler
Audience and Where Produced	For the user of IPCS: at the terminal and at the IPCS print file for some designated messages.
Message Format	IEEnnnnnt text IEE Component identifier nnnnn Message serial number. t Type code: A Action; the user must perform a specific action. D Decision; the user must choose an alternative. E Error; the user must correct the error before continuing. I Information; no user action is required. text Message text.
Associated and Referenced Publications	• *MVS/XA Data Administration: Macro Instruction Reference*, GC26-4014 • *MVS/XA Integrated Catalog Administration: Access Method Services Reference*, GC26-4019 • *MVS/XA Interactive Problem Control System (IPCS) Planning and Customization*, GC28-1406 • *MVS/XA Interactive Problem Control System (IPCS) User's Guide*, GC28-1407 • *MVS/XA Interactive Problem Control System (IPCS) Command Reference*, GC28-1408 • *MVS/XA Message Library: System Codes*, GC28-1157 • *MVS/XA SPL: Service Aids*, GC28-1159 • *MVS/XA SPL: System Macros and Facilities*, Volumes 1 and 2, GC28-1150, GC28-1151 • *MVS/XA SPL: System Modifications*, GC28-1152 • *MVS/XA TSO Guide to Writing a Terminal Monitor Program (TMP) or a Command Processor*, GC28-1295 • *MVS/XA TSO Terminal Messages*, GC38-1046 • *MVS/XA VSAM Catalog Administration: Access Method Services Reference*, GC26-4075

IEE30001I **UNABLE TO ESTABLISH RECOVERY ENVIRONMENT FOR MTRACE - MTRACE FUNCTION TERMINATED**

Explanation: ESTAE recovery environment could not be setup for MTRACE processing.

System Action: The system terminates formatting for the MTRACE keyword.

Problem Determination: Table I, items 5, 16, 29.

IEE30002I **UNABLE TO ACCESS xxxxx - MTRACE FUNCTION TERMINATED**

Explanation: When the system issues this message, **xxxxx** may be either one of the following:

- When **xxxxx** is CVT this indicates that the CVT pointer provided in the print dump parameter list (ABDPL) was either not valid or the pointer to the master scheduler

resident data area (BASEA) in the CVT could not be obtained. The CVT may not be in the dump or there was a storage overlay in the dumped system.

- When **xxxxx** is BASEA this indicates that the master scheduler resident data area (BASEA) resides in the system nucleus and contains a pointer to the Master Trace Table. BASEA may not be in the dump or there was a storage overlay in the dumped system.

System Action: The system terminates formatting for the MTRACE keyword.

Programmer Response: When the message text contains **CVT** include the CVT=cvtaddr control statement and try to reformat the dump.

Problem Determination: Table I, items 5, 16, 29.

IEE30003I **MASTER TRACE NOT ACTIVE IN DUMPED SYSTEM - MTRACE FUNCTION TERMINATED**

Explanation: No Master Trace Table existed in the system when the dump was taken.

System Action: The system terminates formatting for the MTRACE keyword.

Programmer Response: None

Problem Determination: Table I, items 5, 16, 29.

IEE30004I **MASTER TRACE RECOVERY WAS ENTERED ON DUMPED SYSTEM - USING RECOVERY MASTER TRACE TABLE FOR FURTHER PROCESSING**

Explanation: The Master Trace Table on the dumped system could not be used. The recovery Master Trace Table will be formatted.

System Action: Formatting continues with the recovery Master Trace Table.

Programmer Response: None

Problem Determination: Table I, items 5, 16, 29.

IEE30005I **MASTER TRACE TABLE NOT FOUND, POINTER address NOT VALID - MTRACE FUNCTION TERMINATED**

Explanation: The pointer to the Master Trace Table was not valid. The Master Trace Table pointer pointed to data that was not incorrectly formatted. The formatter could not process the data. There may have been a storage overlay in the dumped system causing the incorrect data.

System Action: The system terminates all formatting for the MTRACE keyword.

Programmer Response: Print 4K bytes of storage beginning at **address**. If there was a storage overlay of the Master Trace Table, some of the data in the table may not have been affected.

Problem Determination: Table I, items 5, 16, 29.

IEE30006I **UNABLE TO ACCESS A PAGE OF MASTER TRACE TABLE AT address - MTRACE FUNCTION TERMINATED**

Explanation: The Master Trace Table beginning at address **address** is not in the dump. A complete dump may not have been taken.

System Action: The system terminates formatting for the MTRACE keyword.

Programmer Response: None.

Problem Determination: Table I, items 5, 16, 29.

IEE30007I **INSUFFICIENT STORAGE - nnnnK ADDITIONAL BYTES NEEDED - MTRACE FUNCTION TERMINATED**

Explanation: The system could not obtain **nnnnK** additional bytes of working storage for the MTRACE function.

System Action: The system terminates formatting for the MTRACE keyword.

Programmer Response: Increase region size and reformat the dump.

Problem Determination: Table I, items 5, 16, 29.

IEE30008I **MTRACE FUNCTION ABENDED - ABEND CODE = cde, REASON CODE = rsnc**

Explanation: ABEND occurred during MTRACE function processing.

System Action: If the Master Trace Table could not be accessed, or the error occurred during formatting, the system terminates processing for the MTRACE keyword. Otherwise, the system attempts to format the storage.

Programmer Response: See *MVS/XA System Codes* for an explanation of the system completion code **cde** and reason code **rsnc**.

Problem Determination: Table I, items 5, 16, 29.

IEE30009I **NO NIP MESSAGES IN DUMPED SYSTEM - NIP MESSAGE TRACE TERMINATED**

Explanation: There were no buffers containing NIP messages at the time of the dump. The NIP messages were written to hardcopy before the dump was taken.

System Action: Processing continues with formatting of the Master Trace Table.

Programmer Response: None.

Problem Determination: None.

IEE30010I **NIP MESSAGE TABLE NOT FOUND, POINTER address NOT VALID - NIP MESSAGE TRACE TERMINATED**

Explanation: The pointer to the NIP Message Table was invalid. The NIP Message Table pointer pointed to data that was not incorrectly formatted. The formatter could not process the data. There may have been a storage overlay in the dumped system causing the incorrect data.

System Action: Processing continues with formatting of the Master Trace Table.

Programmer Response: Print 4K bytes of storage beginning at address **address**. If there was a storage overlay of the NIP Message Table, some of the data in the table may not have been affected.

Problem Determination: Table I, items 4 and 16.

IEE30011I **UNABLE TO ACCESS A PAGE OF NIP MESSAGE TABLE AT address - NIP MESSAGE TRACE TERMINATED**

Explanation: The NIP Message Table beginning at address **address** is not in the dump. The dump may not be complete.

System Action: Processing continues with formatting of the Master Trace Table.

Programmer Response: None.

Problem Determination: None.

IEE30012I **UNABLE TO ACCESS UCM - NIP MESSAGE TRACE TERMINATED**

Explanation: The Unit Control Module (UCM) resides in the system nucleus and contains a pointer to the NIP Message Table. The UCM may not be in the dump or there was a storage overlay in the dumped system.

System Action: Processing continues with formatting of the Master Trace Table.

Programmer Response: None.

Problem Determination: None.

TSO Programmer Messages (IKJ)

This section describes TSO messages directed to a terminal user.

Component Name	IKJ
Program Producing Message	Time Sharing Option
Audience and Where Produced	For terminal user: at the terminal. For system programmer: SYSPRINT listing.
Message Format	IKJxxyyyn xx System module prefix (in decimal). yyy Message serial number identifying the program that issued the message. n Type code: A Action. The terminal user must perform the action specified in the message text. I Information; no action is required.
Comments	The words printed in lower case letters in the message text indicate that the system supplies specific information which varies depending on the circumstances. That information is available only when the system displays the message at the terminal. A plus sign (+) after the message text indicates that more information is available online. To receive more information, type a question mark (?) and press the ENTER key. Many messages in this section have no message ID (number). These messages follow and refer back to a message marked with a plus sign (+).
Associated and Referenced Publications	• Message Library: System Messages • *MVS/XA Interactive Problem Control System (IPCS) Planning and Customization*, GC28-1406 • *MVS/XA Interactive Problem Control System (IPCS) User's Guide*, GC28-1407 • *MVS/XA Interactive Problem Control System (IPCS) Command Reference*, GC28-1408

IKJ58630I THE SYMBOL NAME TABLE CANNOT BE SEARCHED SEQUENTIALLY.

Explanation: The symbol name table cannot be searched sequentially to compare the number of symbols found to the number found in the binary tree.

IKJ58631I LOCAL SYMBOLS FOR CLIST PROCEDURE procname

Explanation: This heading precedes the display of local symbols for the sub(procedure) indicated.

IKJ58632I to IKJ58640I

IKJ58632I TSO DUMP FORMAT ROUTINE

Explanation: The "TSO Dump Format Routine" is the title for the TSODATA exit. The formatted dump data that you requested follows this message.

IKJ58633I END OF TSO DUMP FORMAT ROUTINE

Explanation: This message marks the end of the formatted dump data that you requested.

IKJ58634I AN ECT ADDRESS WAS SPECIFIED AND MORE THAN ONE ADDRESS HAS BEEN SELECTED TO HAVE DATA FORMATTED. NO DATA WAS PROCESSED.

Explanation: The (ECT) environment control table contains information about a user's current TSO environment. You specified an ECT address, but more than one ASID (address space identifier) was selected for data formatting. Respecify one or more of the following address space keywords: CURRENT, ERROR, TCBERROR, ASIDLIST, and JOBNAME. Specify the keywords so that only one address space is selected.

System Action: TSODATA terminates processing.

Programmer Response: For correct TSODATA syntax, refer to *MVS/XA IPCS Command Reference*.

IKJ58635I AN LSD ADDRESS WAS SPECIFIED AND MORE THAN ONE ADDRESS SPACE HAS BEEN SELECTED TO HAVE DATA FORMATTED. NO DATA WAS PROCESSED.

Explanation: The LSD (list source descriptor) contains information pertinent to a CLIST or an in-storage list. You specified an LSD address, but more than one ASID (address space identifier) was selected for data formatting. Respecify one or more of the following address space keywords: CURRENT, ERROR, TCBERROR, ASIDLIST, and JOBNAME. Specify the keywords so that only one address space is selected.

System Action: TSODATA terminates processing.

Programmer Response: For correct TSODATA syntax, refer to *MVS/XA IPCS Command Reference*.

IKJ58636I WARNING: THE DUMP IS NOT AT THE SAME LEVEL OF TSO/E AS THE TSODATA VERB EXIT, THE FORMATTED DATA MAY BE INCORRECT.

Explanation: Because the dump you are requesting to format was obtained on a system that had a different release of TSO than the release of TSODATA you are using, part of the formatted data may be incorrect.

System Action: The system issues this warning message and continues dump formatting. However, TSODATA will not process any output for the SYMBOLS keyword.

Programmer Response: You might want to rerun the program on which you obtained the dump while you are logged onto the present level of TSO. Then reissue TSODATA.

IKJ58637I TSO DATA FOLLOWS FOR ADDRESS SPACE NUMBER (address space ID) JOBNAME IS (jobname).

Explanation: The address space identified in the message text was selected for data formatting.

System Action: This message is informational; the system continues processing.

IKJ58638I THE FOLLOWING KEYWORDS ARE IN EFFECT: keywords.

Explanation: The keywords listed in the message text are in effect for TSODATA. Dump formatting continues.

System Action: This message is informational; the system continues processing.

IKJ58639I THE LWA CANNOT BE OBTAINED, THE LWA KEYWORD CANNOT BE PROCESSED.

Explanation: The message that the system issued to your terminal immediately preceding IKJ58639I, contains the reason that TSODATA cannot obtain the LWA (LOGON workarea).

System Action: TSODATA does not display data for the LWA keyword.

Programmer Response: Check the dump for the error described in the message which preceeded IKJ58639I.

IKJ58640I THE ECT CANNOT BE OBTAINED, THE ECT KEYWORD CANNOT BE PROCESSED.

Explanation: The message that the system issued to your terminal immediately preceding IKJ58640I, contains the reason that TSODATA cannot obtain the ECT (environment control table).

System Action: TSODATA does not display data for the ECT keyword.

Programmer Response: Check the dump for the error described in the message which preceded IKJ58640I.

IKJ58641I **THE ECT CANNOT BE OBTAINED, THE STACK KEYWORD CANNOT BE PROCESSED.**

Explanation: The message that the system issued to your terminal immediately preceding IKJ58641I, contains the reason why TSODATA cannot display data for the ECT (environment control table).

System Action: TSODATA cannot process the STACK keyword, because the ECT should contain a pointer to the stack header.

Programmer Response: Check the dump for the error described in the message which preceded IKJ58641I.

IKJ58642I **THE STORAGE FOR THE ECT IS INACCESSIBLE, THE STACK KEYWORD CANNOT BE PROCESSED.**

Explanation: TSODATA uses IPCS (Interactive Problem Control System) to access storage and print information about the dump. The return code from the storage access service is 4; this indicates the pointer to the ECT (environment control table) contained an (1) address that is not valid or (2) an address that IPCS cannot access.

System Action: TSODATA cannot process the STACK keyword, because the ECT, which contain a pointer to the stack header, cannot be accessed.

Programmer Response: Check the dump for the error described in message IKJ58642I.

IKJ58643I **THE ECTIOWA FIELD IS ZERO, THE STACK KEYWORD CANNOT BE PROCESSED.**

Explanation: The ECTIOWA field should contain a pointer to the stack header. The fact that it contains zero means that the pointer to the stack is not available.

System Action: TSODATA cannot process the STACK keyword without the pointer to the stack.

Programmer Response: Check the dump for the error described in message IKJ58643I.

IKJ58644I **THE ECTIOWA FIELD CONTAINS A 31-BIT ADDRESS, THE STACK KEYWORD CANNOT BE PROCESSED.**

Explanation: The ECTIOWA field should contain a 24-bit address, because it points to an address below 16 megabytes in virtual storage. The ECTIOWA field should also contain a pointer to the stack header.

System Action: Because the ECTIOWA contains an invalid pointer, TSODATA cannot obtain the stack.

Programmer Response: Check the *MVS/XA IPCS Command Reference*manual, and look at the dump for the error described in message IKJ58644I.

IKJ58645I **THE STORAGE POINTED TO BY ECTIOWA IS INACCESSIBLE, THE STACK KEYWORD CANNOT BE PROCESSED.**

Explanation: TSODATA uses IPCS (Interactive Problem Control System) to access storage and print information about the dump. The return code from the storage access service is 4; this indicates the pointer to the environment control table (ECT) contained an (1) address that is not valid or (2) an address that IPCS cannot access.

System Action: THE ECTIOWA field should contain a pointer to the stack header. TSODATA cannot process the STACK keyword, because IPCS must access the stack header to continue processing.

Programmer Response: Check the dump for the error described in message IKJ58645I.

IKJ58646I **TSO I/O SERVICES STACK ELEMENTS, FROM STACK HEADER AT nnnnnnnn**

Explanation: The stack header is located at address **nnnnnnnn**. Following this message, the system provides information contained in the TSO I/O Service stack elements (INSTACK). For each element, the system provides the following information:

- A valid element type. The valid types are: terminal, termin, barrier, CLIST, data set, or in-storage list;
- The value of the pointer to the list source descriptor (LSD). If the top or current element in the stack represents a CLIST or an in-storage list, TSODATA will format the LSD that the element points to.

This message is informational only.

IKJ58647I **STORAGE FOR THE STACK IS INACCESSIBLE, PROCESSING FOR THE STACK KEYWORD CANNOT CONTINUE.**

Explanation: TSODATA uses IPCS (Interactive Problem Control System) to access storage and print information about the dump. The return code from the storage access service is 4; this indicates that either (1) the storage for the stack is not valid, or (2) IPCS cannot access the storage for the stack.

System Action: TSODATA cannot process the STACK keyword, because IPCS must access storage for the stack to continue processing.

Programmer Response: Check the *MVS/XA IPCS Command Reference*, and look at the dump for the error described in message IKJ58647I.

IKJ58648I to IKJ58657I

IKJ58648I **THE STORAGE FOR THE ASCB IS INACCESSIBLE.**

Explanation: TSODATA uses IPCS (Interactive Problem Control System) to access storage and print information about the dump. The return code from the storage access service is 4; this indicates that the pointer to the ASCB contains an either (1) an invalid address or (2) an address that IPCS cannot access.

System Action: TSODATA terminates processing.

Programmer Response: Check the *MVS/XA IPCS Command Reference*, and look at the dump for the error described in message IKJ58648I.

IKJ58649I **THE ASCBASCB FIELD IS NOT 'ASCB'.**

Explanation: The ASCBASCB field is the identifier field of the ASCB and should contain the characters ASCB.

System Action: TSODATA terminates processing.

Programmer Response: Look at the dump for the error described in message IKJ58649I.

IKJ58650I **THE ASCBASXB FIELD IS ZERO.**

Explanation: The ASCBASXB field should contain a pointer to the ASXB; the field contains a zero.

System Action: TSODATA terminates processing.

Programmer Response: Look at the dump for the error described in message IKJ58650I.

IKJ58651I **THE STORAGE POINTED TO BY ASCBASXB IS INACCESSIBLE.**

Explanation: TSODATA uses IPCS (Interactive Problem Control System) to access storage and print information about the dump. The return code from the storage access service is 4; this indicates that the ASCBASXB contains an address that is either (1) invalid or (2) an address that IPCS cannot access.

System Action: TSODATA terminates processing.

Programmer Response: Check the *MVS/XA Command Reference*, and look at the dump for the error described in message IKJ58651I.

IKJ58652I **THE ASXBASXB FIELD IS NOT 'ASXB'.**

Explanation: The ASXBASXB field is the identifier field of the ASXB and should contain the characters ASXB.

System Action: TSODATA terminates processing.

Programmer Response: Check the dump for the error described in message IKJ58652I.

IKJ58653I **THE ASXBLWA FIELD IS ZERO.**

Explanation: The ASXBLWA field should contain a pointer to the LWA (LOGON workarea).

System Action: TSODATA terminates processing.

Programmer Response: Check the dump for the error described in message IKJ58653I.

IKJ58654I **THE STORAGE POINTED TO BY ASXBLWA IS INACCESSIBLE.**

Explanation: TSODATA uses IPCS (Interactive Problem Control System) to access storage and print information about the dump. The return code from IPCS is 4; this indicates that the ASXBLWA contains an address that is not valid or that IPCS cannot access.

System Action: TSODATA terminates processing.

Programmer Response: Check the dump for the error described in message IKJ58654I.

IKJ58655I **THE LWALWA FIELD IS NOT 'LWA'.**

Explanation: The LWALWA field is the identifier field of the LWA (LOGON workarea) and should contain the characters LWA.

System Action: TSODATA terminates processing.

Programmer Response: Check the dump for the error described in message IKJ58655I.

IKJ58656I **THE LWAPECT FIELD IS ZERO.**

Explanation: The LWAPECT field of the LWA (LOGON workarea) should contain a pointer to the ECT (environment control table).

System Action: TSODATA will provide formatted output if you specified the LWA keyword, an ECT address with the ECT keyword, or an LSD (List Source Descriptor) address with the SYMBOLS keyword. If you did not specify one of these three, then TSODATA will not provide any formatted output.

Programmer Response: If you want formatted output from TSODATA, specify the LWA keyword, an ECT address with the ECT keyword or and LSD (List Source Descriptor) address with the symbols keyword.

IKJ58657I **THE ECT CANNOT BE OBTAINED, THE SYMBOLS KEYWORD CANNOT BE PROCESSED.**

Explanation: The reason that TSODATA cannot obtain the ECT (environment control table) is contained in the message that the system issued immediately preceding IKJ58657I.

System Action: TSODATA cannot process the SYMBOLS keyword, because TSODATA must obtain the ECT to continue processing.

Programmer Response: You can take either one of the following two actions:

- Check the dump for the error described in the message which preceded IKJ58639I.
- Specify SYMBOLS with LSDADDRESS to get symbols output.

IKJ58658I THE STORAGE FOR THE ECT IS INACCESSIBLE, THE SYMBOLS KEYWORD CANNOT BE PROCESSED.

Explanation: TSODATA uses IPCS (Interactive Problem Control System) to access storage and print information about the dump. The return code from IPCS is 4; this indicates that the pointer to the ECT (environment control table) contains an address that is not valid or that IPCS cannot access.

System Action: TSODATA cannot process the SYMBOLS keyword, because IPCS must access the ECT to continue processing.

Programmer Response: You can take either one of the following two actions:

- Check the dump for the error described in message IKJ58658I.
- Specify SYMBOLS with LSDADDRESS to get symbols output.

IKJ58659I THE ECTIOWA FIELD CONTAINS A ZERO, THE SYMBOLS KEYWORD CANNOT BE PROCESSED.

Explanation: The ECTIOWA field should contain a pointer to the stack header.

System Action: When the ECTIOWA field does not contain a valid pointer, TSODATA cannot process the SYMBOLS keyword.

Programmer Response: You can take either one of the following two actions:

- Check the dump for the error described in message IKJ58659I.
- Specify SYMBOLS with LSDADDRESS to get symbols output.

IKJ58660I THE ECTIOWA FIELD CONTAINS A 31-BIT ADDRESS, THE SYMBOLS KEYWORD CANNOT BE PROCESSED.

Explanation: The ECTIOWA field should contain a 24-bit address, because it should point to an address below 16 megabytes in virtual storage. The ECTIOWA field should contain a pointer to the stack header.

System Action: The ECTIOWA field does not contain a valid pointer; TSODATA cannot process the SYMBOLS keyword.

Programmer Response: You can take either one of the following two actions:

- Check the dump for the error described in message IKJ58660I.
- Specify SYMBOLS with LSDADDRESS to get symbols output.

IKJ58661I THE STORAGE POINTED TO BY ECTIOWA IS INACCESSIBLE, THE SYMBOLS KEYWORD CANNOT BE PROCESSED.

Explanation: TSODATA uses IPCS (Interactive Problem Control System) to access storage and print information about the dump. The return code from the storage access service is 4; this indicates that the ECTIOWA contains (1) an invalid address or (2) an address that IPCS cannot access.

System Action: TSODATA cannot process the SYMBOLS keyword, because IPCS must access the storage that ECTIOWA is pointing to.

Programmer Response: You can take either one of the following two actions:

- Check the dump for the error described in the message IKJ58661I.
- Specify SYMBOLS with LSDADDRESS to get symbols output.

IKJ58662I THE POINTER TO THE TOP ELEMENT ON THE STACK IS ZERO, THE SYMBOLS KEYWORD CANNOT BE PROCESSED.

Explanation: The pointer to the top element on the stack should contain a valid address. The pointer contains zero and TSODATA cannot obtain the top element on the stack.

System Action: TSODATA cannot process the SYMBOLS keyword, because TSODATA must obtain the top element on the stack to continue processing.

Programmer Response: You can take either one of the following two actions:

- Check the dump for the error described in the message IKJ58662I.
- Specify SYMBOLS with LSDADDRESS to get symbols output.

IKJ58663I to IKJ58670I

IKJ58663I **THE STORAGE FOR THE TOP ELEMENT ON THE STACK IS INACCESSIBLE, THE SYMBOLS KEYWORD CANNOT BE PROCESSED.**

Explanation: TSODATA uses IPCS (Interactive Problem Control System) to access storage and print information about the dump. The return code from IPCS is 4; this indicates that the pointer to the top element on the stack contains an address that is not valid or that IPCS cannot access.

System Action: TSODATA cannot process the SYMBOLS keyword, because IPCS must access the top element on the stack to continue processing.

Programmer Response: You can take either one of the following two actions:

- Check the dump for the error described in message IKJ58663I.
- Specify SYMBOLS with LSDADDRESS to get symbols output.

IKJ58664I **THE TOP ELEMENT ON THE STACK DOES NOT REPRESENT A CLIST, THE SYMBOLS KEYWORD CANNOT BE PROCESSED.**

Explanation: The SYMBOLS keyword pertains only to CLISTs.

System Action: TSODATA cannot process the SYMBOLS keyword.

Programmer Response: You can take either one of the following two actions:

- Check the dump for the error described in message IKJ58664I.
- Specify SYMBOLS with LSDADDRESS to get symbols output.

IKJ58665I **THE LSD ADDRESS OF THE TOP ELEMENT ON THE STACK IS ZERO, THE SYMBOLS KEYWORD CANNOT BE PROCESSED.**

Explanation: The top element on the stack should contain a pointer to the LSD (List Source Descriptor).

System Action: TSODATA cannot process the SYMBOLS keyword, because the top element on the stack does not contain a valid LSD address.

Programmer Response: You can take either one of the following two actions:

- Check the dump for the error described in message IKJ58665I.
- Specify SYMBOLS with LSDADDRESS to get symbols output.

IKJ58666I **THE STORAGE FOR THE LSD IS INACCESSIBLE, THE SYMBOLS KEYWORD CANNOT BE PROCESSED.**

Explanation: TSODATA uses IPCS (Interactive Problem Control System) to access storage and print information about the dump. The return code from IPCS is 4; this indicates that the pointer to the LSD (List Source Descriptor) contains an address that is not valid or that IPCS cannot access.

System Action: TSODATA cannot process the SYMBOLS keyword, because IPCS must access the LSD to continue processing.

Programmer Response: Check the dump for the error described in message IKJ58666I.

IKJ58668I **THE POINTER TO THE FIRST SYMBOL NAME TABLE CONTAINS A ZERO, THE SYMBOLS CANNOT BE DISPLAYED.**

Explanation: The pointer to the first symbol name table should contain a valid address, but it contains a zero.

System Action: TSODATA cannot obtain the symbol name table or display the symbols.

Programmer Response: Check the dump for the error described in message IKJ58668I.

IKJ58669I **THE POINTER TO THE FIRST SYMBOL NAME TABLE CONTAINS A 31-BIT ADDRESS, THE SYMBOLS CANNOT BE DISPLAYED.**

Explanation: The pointer to the first symbol name table should contain a 24-bit address, because it should point to an address below 16 megabytes in virtual storage.

System Action: TSODATA cannot display the symbols, because the pointer to the first symbol name table is not valid.

Programmer Response: Check the dump for the error described in message IKJ58669I.

IKJ58670I **THE POINTER TO THE FIRST SYMBOL VALUE TABLE CONTAINS A ZERO, THE SYMBOLS CANNOT BE DISPLAYED.**

Explanation: The pointer to the first symbol value table should contain a valid address, but it contains a zero.

System Action: TSODATA cannot obtain the symbol value table or display the symbols.

Programmer Response: Check the dump for the error described in message IKJ58670I.

IKJ58671I **THE POINTER TO THE FIRST SYMBOL VALUE TABLE CONTAINS A 31-BIT ADDRESS, THE SYMBOLS CANNOT BE DISPLAYED.**

Explanation: The pointer to the first symbol value table should contain a 24-bit address, because it should point to an address below 16 megabytes in virtual storage.

System Action: TSODATA cannot display the symbols, because the pointer to the first symbol value table is not valid.

Programmer Response: Check the dump for the error described in message IKJ58671I.

IKJ58672I **SYMBOLS FOR CLIST WITH EXEC COMMAND DATA BLOCK AT nnnnnnnn.**

Explanation: The formatted dump of the symbols for the CLIST with the exec command data block at address **nnnnnnnn** follows message IKJ58672I.

System Action: The system displays the symbols for the CLIST.

IKJ58673I **THE STORAGE FOR THE FIRST SYMBOL VALUE TABLE HEADER IS INACCESSIBLE, THE SYMBOLS CANNOT BE DISPLAYED.**

Explanation: TSODATA uses IPCS (Interactive Problem Control System) to access storage and print information about the dump. The return code from IPCS is 4; this indicates that the pointer to the first symbol value table header contains an address that is not valid or that IPCS cannot access.

System Action: TSODATA cannot display the symbols, because IPCS must access the first symbol value table header.

Programmer Response: Check the dump for the error described in message IKJ58673I.

IKJ58674I **THE STORAGE FOR THE SECOND SYMBOL VALUE TABLE HEADER IS INACCESSIBLE, THE SYMBOLS CANNOT BE DISPLAYED.**

Explanation: TSODATA uses IPCS (Interactive Problem Control System) to access storage and print information about the dump. The return code from IPCS is 4; this indicates that the pointer to the second symbol value table header contains an address that is not valid or that IPCS cannot access.

System Action: TSODATA cannot display the symbols, because IPCS must access the second symbol value table header to continue processing.

Programmer Response: Check the dump for the error described in message IKJ58674I.

IKJ58676I **THE STORAGE FOR THE NEXT SYMBOL NAME TABLE POINTED TO FROM THE SYMBOL NAME TABLE AT nnnnnnnn IS INACCESSIBLE.**

Explanation: There can be one or more symbol name tables linked with pointers from one to the other. The symbol name table at address **nnnnnnnn** is the last symbol name table that IPCS (Interactive Problem Control System) could access. The return code from IPCS is 4; this indicates that the pointer to the next symbol name table contains an address that is not valid or an address that IPCS cannot access.

System Action: TSODATA cannot continue to process the SYMBOLS keyword, because the first symbol name table must be obtained in order to complete symbols processing.

Programmer Response: Check the dump for the error described in message IKJ58676I.

IKJ58677I **PROCESSING FOR THE SYMBOLS KEYWORD CANNOT CONTINUE.**

Explanation: The message that the system issued to your terminal immediately preceding IKJ58677I, contains the reason that TSODATA cannot continue processing the SYMBOLS keyword.

System Action: TSODATA stops processing the SYMBOLS keyword.

Programmer Response: Check the dump for the error described in message immediately preceding IKJ58677I.

IKJ58678I **THE ADDRESS OF THE SYMBOL VALUE IS NOT A VALID SYMBOL VALUE TABLE ELEMENT ADDRESS, A SYMBOL VALUE HAS NOT BEEN DISPLAYED.**

Explanation: The address of the symbol value is not in the range of valid symbol value table element addresses.

System Action: The system does not display the symbol value for this particular symbol.

Programmer Response: Check the dump for the error described in message IKJ58678I.

IKJ58679I **THE SYMBOL NAME TABLE ELEMENT CURRENTLY BEING PROCESSED DOES NOT POINT BACK TO ITS PARENT NODE.**

Explanation: Every symbol name table element should contain a pointer back to its parent node.

System Action: TSODATA continues processing.

Programmer Response: Check the dump for the error described in message IKJ58679I.

IKJ58680I to IKJ58686I

IKJ58680I **THE STORAGE FOR THE SYMBOL VALUE IS INACCESSIBLE. A VALUE HAS NOT BEEN DISPLAYED.**

Explanation: TSODATA uses IPCS (Interactive Problem Control System) to access storage and print information about the dump. The return code from IPCS is 4; this indicates that the pointer to the symbol value contains an address that is not valid or that IPCS cannot access.

System Action: TSODATA terminates processing.

Programmer Response: Check the dump for the error described in message IKJ58680I.

IKJ58681I **THE STORAGE FOR THE LEFT NODE OF SYMBOL NAME TABLE ELEMENT AT nnnnnnnn IS INACCESSIBLE, SOME SYMBOLS MAY NOT BE DISPLAYED.**

Explanation: The symbol name table element at address **nnnnnnnn** contains a non-zero pointer to the left node. The pointer to the left node is probably incorrect, because IPCS (Interactive Problem Control System) cannot access the left node.

System Action: TSODATA stops processing.

Programmer Response: Check the dump for the error described in message IKJ58681I.

IKJ58682I **THE STORAGE FOR THE RIGHT NODE OF SYMBOL NAME TABLE ELEMENT AT nnnnnnnn IS INACCESSIBLE, SOME SYMBOLS MAY NOT BE DISPLAYED.**

Explanation: The symbol name table element at address **nnnnnnnn** contains a non-zero pointer to the right node. The pointer to the right node is probably incorrect, because IPCS (Interactive Problem Control System) cannot access the right node.

System Action: TSODATA stops processing.

Programmer Response: Check the dump for the error described in message IKJ58682I.

IKJ58683I **THE SYMBOL VALUE HAS A LENGTH OF nnnnnn, THE FIRST 256 CHARACTERS HAVE BEEN DISPLAYED.**

Explanation: The symbol value precedes this message.

System Action: The system displays only the first 256 characters of the symbol value.

Programmer Response: Check the dump for the entire length (**nnnnnnn**) of the symbol value.

IKJ58684I **THE STORAGE FOR THE ELEMENTS IN THE SYMBOL NAME TABLE AT nnnnnnnn IS INACCESSIBLE.**

Explanation: TSODATA uses IPCS (Interactive Problem Control System) to access storage and print information about the dump. The return code from IPCS is 4; this indicates that IPCS cannot acccss the storage for the elements in the symbol name table.

The symbol name table begins at address **nnnnnnnn**. One of two conditions caused this error condition. IPCS could access some, but not all, of the elements in the symbol name table, or IPCS could not access any of the elements in the symbol name table.

System Action: TSODATA stops processing symbols.

Programmer Response: Check the dump for the error described in message IKJ58684I.

IKJ58685I **THE NUMBER OF SYMBOLS IN THE BINARY TREE DOES NOT MATCH THE NUMBER FOUND SEQUENTIALLY SEARCHING THE SYMBOL NAME TABLES.**

Explanation: There are two sets of chains, the symbol name table chain and the binary tree. Both sets of chains should contain the same elements linked with pointers in different ways. Therefore, the number of elements in both chains should be the same, unless there is an error in one or both chains.

System Action: TSODATA continues processing.

Programmer Response: You can check the symbol name table chain and the binary tree to make sure that they have an equal number of elements.

IKJ58686I **THE STORAGE FOR THE TOP ELEMENT OF THE BINARY TREE IS INACCESSIBLE, THE SYMBOLS KEYWORD CANNOT BE DISPLAYED.**

Explanation: TSODATA uses IPCS (Interactive Problem Control System) to access storage and print information about the dump. The return code from IPCS indicates that the pointer to the top element of the binary tree contains an address that is not valid or that IPCS cannot access.

System Action: TSODATA cannot display the symbols.

Programmer Response: Check the dump for the error described in message IKJ58686I.

IKJ58687I **THE ASID WAS SELECTED BECAUSE: reason.**

Explanation: The ASID (address space identifier) was selected for processing. The message contains the **reason** why the system selected the ASID for processing.

System Action: The system selects the ASID for processing.

IKJ58688I **THE LSDEXEC FIELD CONTAINS A ZERO, THE SYMBOLS KEYWORD CANNOT BE PROCESSED.**

Explanation: The LSDEXEC field must contain a pointer. It is necessary to process the symbols keyword.

System Action: TSODATA cannot process the SYMBOLS keyword.

Programmer Response: Check the dump for the error described in message IKJ58688I.

IKJ58689I **NO ASIDS WERE SELECTED FOR DATA FORMATTING FOR THE ADDRESS SPACE KEYWORDS IN EFFECT.**

Explanation: None of the address spaces in the dump meet the selection criteria for the address space keywords in effect. For example, if you specified a particular ASID (address space identifier) for data formatting, it is probably not in your dump.

System Action: TSODATA terminates processing.

Programmer Response: Respecify TSODATA with a different ASID or with different address space keywords (CURRENT, ERROR, TCBERROR, ASIDLIST, or JOBNAME).

IKJ58690I **THE STACK ELEMENT AT nnnnnnnn IS INVALID, PROCESSING FOR THE STACK KEYWORD CANNOT CONTINUE.**

Explanation: The stack element at the address **nnnnnnnn** is not valid, because the element type field does not contain a valid type.

When TSODATA obtains stack elements, each stack element should contain the following:

- A valid element type. The valid types are: terminal, termin, barrier, CLIST, data set, or in-storage list.
- A pointer to the List Source Descriptor (LSD) if one exists.

System Action: TSODATA stops processing the STACK keyword.

Programmer Response: Check the dump for the error described in message IKJ58690I.

IKJ58691I **LOCAL SYMBOLS FOR ATTENTION/ERROR EXIT IN CLIST PROCEDURE procname.**

Explanation: This heading precedes the display of local symbols for the attention or error exit in the CLIST (sub)procedure indicated.

IKJ58692I **LOCAL SYMBOLS FOR ATTENTION/ERROR EXIT WITH EXEC COMMAND DATA BLOCK AT nnnnnnnn.**

Explanation: If the return code from the IPCS exit storage access service indicates that the CLIST (sub)procedure name could not be accessed, then this heading precedes the display of local symbols for the attention or error exit in CLIST (sub)procedure.

IKJ58693I **THE STORAGE FOR THE NEXT EXEC COMMAND DATA BLOCK POINTED TO BY THE EXEC COMMAND DATA BLOCK AT nnnnnnnn IS INACCESIBLE.**

Explanation: TSODATA uses IPCS exit storage access service to access storage from a dump. The return code from the IPCS storage access service indicates that the next exec command data block pointed to by the exec command data block at the address indicated could not be processed.

IKJ58694I **THE STORAGE FOR THE CLIST PROCEDURE NAME IS INACCESSIBLE, AN ALTERNATE LOCAL SYMBOLS HEADER FOLLOWS.**

Explanation: TSODATA uses IPCS exit storage access service to access storage from a dump. The return code from the IPCS storage access service indicates that the CLIST (sub)procedure name could not be accessed. Because the CLIST (sub)procedure name could not be accessed, the local SYMBOLS heading that contains the (sub)procedure name cannot be displayed and an alternate heading is displayed.

IKJ58695I **GLOBAL SYMBOLS FOR CLIST WITH EXEC COMMAND DATA EXTENSION BLOCK AT nnnnnnnn.**

Explanation: This heading precedes the display of global symbols for the CLIST indicated.

IKJ58696I **LOCAL SYMBOLS FOR CLIST PROCEDURE WITH EXEC COMMAND DATA BLOCK AT nnnnnnnn.**

Explanation: If the return code from the IPCS exit storage access service indicates that the CLIST (sub)procedure name could not be accessed, then this heading precedes the display of local symbols for the CLIST (sub)procedure.

IKJ58697I to IKJ58701I

IKJ58697I **THE POINTER TO THE EXEC COMMAND DATA EXTENSION BLOCK CONTAINS A ZERO, THE GLOBAL SYMBOLS CANNOT BE DISPLAYED.**

Explanation: If the pointer to the exec command data extension block contains a zero, it is not valid and the global symbols cannot be displayed.

IKJ58698I **THE STORAGE FOR THE EXEC COMMAND DATA EXTENSION BLOCK IS INACCESSIBLE, THE GLOBAL SYMBOLS CANNOT BE DISPLAYED.**

Explanation: TSODATA uses IPCS exit storage access service to access storage from a dump. The return code from the IPCS storage access service indicates that the exec command data extension block could not be accessed. Therefore, the global symbols cannot be displayed.

IKJ58699I **THE TOTAL NUMBER OF SYMBOLS IN THE BINARY TREE DOES NOT MATCH THE TOTAL SEARCHED SEQUENTIALLY IN THE SYMBOL NAME TABLES.**

Explanation: The total number of symbols found searching all of the binary trees for the CLIST procedure and is subprocedures is calculated, as well as the total number of symbols found sequentially searching all of the symbol name tables for the CLIST procedure and its subprocedures. The two values do not match.

IKJ58700I **THE STORAGE FOR THE EXEC COMMAND DATA BLOCK POINTED TO BY LSDEXEC IS INACCESSIBLE.**

Explanation: TSODATA uses IPCS (Interactive Problem Control System) to access storage and print information about the dump. The return code from IPCS is 4; this indicates that the LSDEXEC contains an address that is not valid or that IPCS cannot access.

System Action: TSODATA cannot process the SYMBOLS keyword, because IPCS must access the storage pointed to by LSDEXEC to continue processing.

Programmer Response: Check the dump for the error described in message IKJ58700I.

IKJ58701I **THE STORAGE FOR THE FIRST SYMBOL NAME TABLE IS INACCESSIBLE, THE SYMBOLS NAME TABLE CANNOT BE SEARCHED SEQUENTIALLY.**

Explanation: TSODATA uses IPCS (Interactive Problem Control System) to access storage and print information about the dump. The return code from IPCS is 4; this indicates that the pointer to the first symbol name table contains an address that is not valid or that IPCS cannot access.

System Action: TSODATA cannot search the symbols name table sequentially, because the first symbol name table must be obtained in order to complete symbols processing.

Programmer Response: Check the dump for the error described in message IKJ58701I.

Input/Output (I/O) Supervisor Messages (IOS)

Component Name	IOS
Program Producing Message	Input/Output Supervisor (IOS)
Audience and Where Produced	For the user of IPCS: at the terminal and at the IPCS print file for some designated messages.
Message Format	IOSnnnnnt text IOS Component identifier nnnnn Message serial number. t Type code: A Action; the user must perform a specific action. D Decision; the user must choose an alternative. E Error; the user must correct the error before continuing. I Information; no user action is required. text Message text.
Associated and Referenced Publications	• *MVS/XA Data Administration: Macro Instruction Reference*, GC26-4014 • *MVS/XA Integrated Catalog Administration: Access Method Services Reference*, GC26-4019 • *MVS/XA Interactive Problem Control System (IPCS) Planning and Customization*, GC28-1406 • *MVS/XA Interactive Problem Control System (IPCS) User's Guide*, GC28-1407 • *MVS/XA Interactive Problem Control System (IPCS) Command Reference*, GC28-1408 • *MVS/XA Message Library: System Codes*, GC28-1157 • *MVS/XA SPL: Service Aids*, GC28-1159 • *MVS/XA SPL: System Macros and Facilities*, Volumes 1 and 2, GC28-1150, GC28-1151 • *MVS/XA SPL: System Modifications*, GC28-1152 • *MVS/XA TSO Guide to Writing a Terminal Monitor Program (TMP) or a Command Processor*, GC28-1295 • *MVS/XA TSO Terminal Messages*, GC38-1046 • *MVS/XA VSAM Catalog Administration: Access Method Services Reference*, GC26-4075

IOS10001I **Unable to complete analysis of active and queued I/O**

Explanation: The IOS ANALYZE exit was unable to access data necessary to identify all of the I/O resource contention current at the time of the dump.

System Action: The system continues processing.

Programmer Response: To determine the state of the I/O devices, run the IOSDATA verb exit.

Problem Determination: None

IOS10101I to IOS10302I

IOS10101I **Unable to access cbname AT *addr*. Storage not in dump.**

Explanation: The control block *cbname* was not successfully accessed in the dump.

System Action: The system continues processing.

IOS10104I **DCQ header at *xxxxxxxx* contains incorrect count of DCQ elements**

Explanation: The DCQCOUNT field does not contain the correct number of DCQ elements.

System Action: The system continues processing.

IOS10105I **UCB at *xxxxxxxx* has incorrect ID**

Explanation: The UCBID field of the UCB at *xxxxxxxx* did not contain 'FF'X.

System Action: The system continues processing.

IOS10106I **DCQ element at *xxxxxxxx* contains incorrect count of UCBs**

Explanation: The DCQUCBCT field of the DCQ element at *xxxxxxxx* does not contain the number of UCBs counted for this device class.

System Action: The system continues processing.

IOS10107I **IOQ at *xxxxxxxx* does not point to UCB at *ucbaddr***

Explanation: The IOQUCB field of the IOQ at *xxxxxxx* does not contain the address *ucbaddr* of the UCB on which this IOQ is chained. The IOQ chain may be damaged.

System Action: The system continues processing.

IOS10108I **UCB at *xxxxxxxx* does not point to last IOQ AT *ioqaddr***

Explanation: The UCBIOQL field of the UCB at *xxxxxxxx* does not contain the address *ioqaddr* of the last IOQ chained off of this UCB.

System Action: The system continues processing.

IOS10110I ***cbname* chain at *cbaddr* appears circular**

Explanation: The control block *cbname* chain contains more entries then the threshold value. The threshold value is 256 for the IOQ and CRWQ chains, and is 8 for the EWA chain. The *cbname* chain may be damaged.

System Action: The system continues processing.

IOS10111I **Associated IOQ chain at *xxxxxxxx* did not pass validity checking**

Explanation: The IOQ chain at *xxxxxxxx* is not one of the valid types. The IOQ chain may be damaged.

System Action: The system continues processing.

IOS10112I **IOSB at *xxxxxxxx* does not point to UCB at *ucbaddr***

Explanation: The IOSUCB field of the IOSB at *xxxxxxxx* does not contain the address *ucbaddr* of the UCB on which the IOQ pointing to this IOSB is chained

System Action: The system continues processing.

IOS10113I **IOQs on chain at *xxxxxxxx* are not in order of priority**

Explanation: The IOQ chain at *xxxxxxxx* did not have all the non-start requests queued ahead of the start requests or the start requests were not in descending priority order. The IOQ chain may be damaged.

System Action: The system continues processing.

IOS10114I **Level mismatch between IOSB at *iosbaddr* and UCB at *ucbaddr***

Explanation: A level mismatch was detected. This occurs when the level indicated by the UCBLEVEL field was higher than both the level indicated by the IOSLEVEL field and the QUIESCE level.

System Action: The system continues processing.

IOS10201I **Unsupported acronym found:** ***name***

Explanation: The control block acronym *name* was not found in the table of control blocks that this function supports.

System Action: Processing continues

IOS10202I **Unable to access *name*. Storage not in dump.**

Explanation: The control block *name* was not successfully accessed in the dump.

System Action: The system continues processing.

IOS10302I **No UCBs were formatted for operand *opname***

Explanation: A list of device numbers and device number ranges was specified for operand *opname*. Either one of the following conditions is true. First, No UCBs were found for these device numbers. Second, if the operand was a device class name, none of these UCBs were in the requested device class.

System Action: The system continues processing.

IOS10303I **Channel path *cp* undergoing recovery**

Explanation: Channel path *cp* is undergoing recovery.

System Action: The system continues processing.

IOS10304I **SCD at *xxxxxxxx* indiactes that hot I/O was detected on device *ddd***

Explanation: Hot I/O was detected on device *ddd*

System Action: The system continues processing.

IOS10305I **The last UCB accessed was at *xxxxxxxx***

Explanation: An error occurred while accessing the UCB chain. The system will format the last valid UCB.

System Action: The system continues processing.

System Resources Manager Messages (IRA)

Component Name	SRM
Program Producing Message	System Resources Manager (SRM)
Audience and Where Produced	For the user of IPCS: at the terminal and at the IPCS print file for some designated messages.
Message Format	IRAnnnnnt text IRA Component identifier nnnnn Message serial number. t Type code: A Action; the user must perform a specific action. D Decision; the user must choose an alternative. E Error; the user must correct the error before continuing. I Information; no user action is required. text Message text.
Associated and Referenced Publications	• *MVS/XA Data Administration: Macro Instruction Reference*, GC26-4014 • *MVS/XA Integrated Catalog Administration: Access Method Services Reference*, GC26-4019 • *MVS/XA Interactive Problem Control System (IPCS) Planning and Customization*, GC28-1406 • *MVS/XA Interactive Problem Control System (IPCS) User's Guide*, GC28-1407 • *MVS/XA Interactive Problem Control System (IPCS) Command Reference*, GC28-1408 • *MVS/XA Message Library: System Codes*, GC28-1157 • *MVS/XA SPL: Service Aids*, GC28-1159 • *MVS/XA SPL: System Macros and Facilities*, Volumes 1 and 2, GC28-1150, GC28-1151 • *MVS/XA SPL: System Modifications*, GC28-1152 • *MVS/XA TSO Guide to Writing a Terminal Monitor Program (TMP) or a Command Processor*, GC28-1295 • *MVS/XA TSO Terminal Messages*, GC38-1046 • *MVS/XA VSAM Catalog Administration: Access Method Services Reference*, GC26-4075

IRA10101I **Unable to produce SRM analysis for this address space**

Explanation: SRM will not produce an analysis of this address space. The OUCB failed validity checks. IPCS or IRARMCBV issue message BLS18058I or IRA10201I to indicate the nature of the OUCB error.

System Action: The system will not perform any additional SRM analysis of this address space.

Programmer Response: The situation which prevents analysis may be a contributor to the error being analyzed.

Problem Determination: Table I, items 5, 16, 29.

IRA10102I to IRA10107I

IRA10102I **This address space is on the SRM *srmqueue* queue**

Explanation: The SRM queue *srmqueue*, reflects the current status of the address space. The IN queue consists of address spaces which are swapped in and running or are non-swappable. The OUT queue consists of address spaces which have ready work but have not been swapped in to execute. The Logical Swap WAIT queue consists of address spaces which currently have no work to do and have been logically swapped out. The WAIT queue consists of address spaces which have no work to do and have been physically swapped out.

System Action: The system continues analysis.

Programmer Response: Confirm that the address space is on the expected queue.

Problem Determination: Table I, items 5, 16, 29.

IRA10103I **It is moving to the SRM *srmqueue* queue**

Explanation: Message IRA10102I always precedes this message. The status of the address space has changed and it is in the process of moving to a different SRM queue. The SRM queue **srmqueue** can be one of the following: IN, OUT, LOGICAL SWAP WAIT or WAIT.

System Action: The system continues analysis.

Programmer Response: Confirm that this description matches the expected state of the address space.

Problem Determination: Table I, items 5, 16, 29.

IRA10104I **The reason for swap-out is *rsnc***

Explanation: The swapout reason code *rsnc* is stored in the OUCBSRC field in the OUCB and can be one of the following:

Swapout Reason Code	Explanation
1	input terminal wait
2	output terminal wait
3	long wait
4	auxiliary storage shortage
5	real pageable storage shortage
6	detected wait
7	requested swap
8	enqueue exchange
9	exchange on recommendation value
10	unilateral swap
11	transition to nonswappable

System Action: The system continues analysis.

Programmer Response: None.

Problem Determination: Table I, items 5, 16, 29.

IRA10105I **In the swap-out process, *zzz***

Explanation: The system issues this informational message where *zzz* is one of the following.

- RCT has been posted.
- QUIESCE has started.
- QUIESCE has completed.
- QUIESCE has failed.

System Action: The system continues analysis.

Programmer Response: In the case of an address space which appears to be "hung" this will give the location of a possible bottleneck. This refers to RCT QUIESCE.

Problem Determination: Table I, items 5, 16, 29.

IRA10106I **A TRANSWAP SYSEVENT is waiting completion of DONTSWAP/OKSWAP SEQUENCE.**

Explanation: The TRANSWAP SYSEVENT causes the following results to occur: First, the address space is swapped out, second, the address space is swapped into preferred storage and finally the address space is made nonswappable. The use of the DONTSWAP SYSEVENT is limited to situations of very short duration, so the TRANSWAP SYSEVENT will wait for its completion. An incorrect use of these SYSEVENTs may explain why an address space is hung.

System Action: Analysis continues.

Programmer Response: Review the use of these SYSEVENTs. See *Supervisor Macros* for explanations and guidelines concerning the use of these SYSEVENTs.

Problem Determination: Table I, items 5, 16, 29.

IRA10107I **The OUCB is on an SRM queue whose anchor block has a queue name field that does not contain a valid value.**

Explanation: The OUCB representing the address space submitted for analysis is on a queue whose anchor block (RMQH) does not contain a valid name in the RMQHNAME field. Valid queue names are:

- WTQE for the WAIT queue.

- OTQE for the OUT queue.
- LSQE for the Logical Swap Wait queue.
- INQE for the IN queue.

System Action: If the address space is not in transition, the OUCBOUT, OUCBLSW, and the OUCBOFF bits in the OUCB determine the address space's queue. The system issues message IRA10112I to indicate which queue the address space is on.

Programmer Response: The situation which interferes with analysis may be a contributor to the error being analyzed.

Problem Determination: Table I, items 5, 16, 29.

IRA10108I **The OUCB for this address space is on an SRM queue with a forward pointer that is not valid.**

Explanation: In attempting to determine which queue (IN, OUT, WAIT, or LOGICAL SWAP WAIT) the OUCB for this address space was on, the analysis routine found that it could not resolve a forward pointer reference. To find out which queue the OUCB was on, the analysis routine followed an OUCBFWD pointer to an OUCB that was not in the dump in an attempt to reach the RMQH block.

System Action: To determine which queue the OUCB is on, the analysis routine checks the backward pointers (OUCBBAK) starting from the OUCB which represents the address space being analyzed.

Programmer Response: The situation which interferes with analysis may be a contributor to the error being analyzed.

Problem Determination: Table I, items 5, 16, 29.

IRA10109I **The OUCB for this address space is on an SRM queue with a circular forward loop.**

Explanation: While trying to determine which queue (IN, OUT, WAIT, or LOGICAL SWAP WAIT) the OUCB for this address space was on, the analysis routine found that the forward pointers from the OUCB's (OUCBFWD) formed a circular loop. Due to the circular loop, the analysis routine could not reach the queue anchor (RMQH).

System Action: The analysis routine will determine the type of queue by following the OUCBs backward pointers (OUCBBCK).

Programmer Response: The situation which interferes with analysis may be a contributor to the error being analyzed.

Problem Determination: Table I, items 5, 16, 29.

IRA10110I **The OUCB for this address space is on an SRM queue with a backward pointer that is not valid.**

Explanation: The analysis routine tried two methods to determine the type of queue that the address space for the OUCB was on. First, the analysis routine failed in its attempt to find the queue header by following the forward pointers. Then the routine failed in its attempt to find the queue header when it could not resolve a backward pointer reference. The analysis routine was following an OUCBBCK pointer to an OUCB that was not in the dump.

System Action: If the address space is not in transition, the system issues message IRA101121 which contains the status of the bits in the OUCB.

Programmer Response: The situation which interferes with analysis may be a contributor to the error being analyzed.

Problem Determination: Table I, items 5, 16, 29.

IRA10111I **The OUCB for this address space is on an SRM queue with a circular backward loop.**

Explanation: By following the forward pointers, the analysis routine tried to determine if the OUCB address space was on an IN, OUT, WAIT or LOGICAL SWAP WAIT queue. After failing at this attempt, the analysis routine discovered that the backward pointers from the OUCBs (OUCBBCK) formed a circular loop. The routine could not reach the queue anchor (RMQH).

System Action: If the address space is not transitioning, the system issues message IRA10112I, which contains the status of the OUCB. The queue will be determined from bits in the OUCB.

Programmer Response: The situation which interferes with analysis may be a contributor to the error being analyzed.

Problem Determination: Table I, items 5, 16, 29.

IRA10112I **According to the fields OUCBOUT, OUCBOFF, and OUCBLSW, this address space is on the SRM *srmqueue* queue.**

Explanation: *srmqueue* can be one of the following SRM queues: IN, OUT, WAIT or LOGICAL SWAP WAIT. The system issues message IRA10112I if the analysis routine failed to determine the type of SRM queue by following the OUCB forward pointer (OUCBFWD) and the OUCB backward pointers (OUCBBCK). Message IRA10112I is always preceded by either message IRA10110I or IRA10111I. The system issues message IRA101121I when an address space is not in transition to another queue (OUCBGOB, OUCBGOI and OUCBGOO are 0).

System Action: The analysis routine continues.

Programmer Response: The situation which interferes with analysis may be a contributor to the error being analyzed.

Problem Determination: Table I, items 5, 16, 29.

IRA10113I **The fields in the OUCB: OUCBOUT, OUCBOFF, and OUCBLSW indicate that this address space is on the SRM *srmqueue* queue; however, the address space is on the SRM *srmqueue* queue.**

Explanation: The SRM queue **srmqueue** can be one of the following: IN, OUT, WAIT, or LOGICAL SWAP WAIT. For some reason, the queue that the OUCB was found on is not the same queue as the one that the OUCB fields point to. The system issues message IRA101113I when the address space is not in transition to another queue (OUCBGOB, OUCBGOI and OUCBGOO are 0). In this instance, with an address space in transition, OUCBOUT, OUCBOFF and OUCBLSW represent the OUCBs destination queue, and not the queue that the OUCB is currently on.

System Action: The analysis routine continues.

Programmer Response: The inconsistency in the OUCB queue may indicate the cause of the error being analyzed.

Problem Determination: Table I, items 5, 16, 29.

IRA10114I **Unable to determine the identity of the SRM queue for this address space.**

Explanation: The analysis routine could not determine the type of queue for the OUCB address space. The system will issue message IRA10114I for only those address spaces that are in transition to another queue.

System Action: The analysis routine continues.

Programmer Response: The situation which prevents analysis may be a contributor to the error being analyzed.

Problem Determination: Table I, items 5, 16, 29.

IRA10201I **OUCB is marked as not valid.**

Explanation: The bit OUCBINV bit is set on. This bit must be off for a valid OUCB.

System Action: The OUCB should not be used for analysis.

Programmer Response: None.

Problem Determination: Table I, items 5, 16, 29.

IRA10202I **OUCB queue indicators are not valid.**

Explanation: OUCBOUT, OUCBOFF, or OUCBLSW flags do not indicate a valid queue.

System Action: The OUCB should not be used for analysis.

Programmer Response: The situation which interferes with analysis may be a contributor to the error being analyzed.

Problem Determination: Table I, items 5, 16, 29.

IRA10203I **OUCB swap out reason code is not valid.**

Explanation: OUCBSRC is not in the range 00 to 11. See message IRA10104I for the valid OUCBSRC ranges.

System Action: The OUCB should not be used for analysis.

Programmer Response: None.

Problem Determination: Table I, items 5, 16, 29.

Global Resource Serialization Messages (ISG)

Component Name	GRS
Program Producing Message	Global Resource Serialization (GRS)
Audience and Where Produced	For the user of IPCS: at the terminal and at the IPCS print file for some designated messages.
Message Format	ISGnnnnnt text ISG Component identifier nnnnn Message serial number. t Type code: A Action; the user must perform a specific action. D Decision; the user must choose an alternative. E Error; the user must correct the error before continuing. I Information; no user action is required. text Message text.
Associated and Referenced Publications	• *MVS/XA Data Administration: Macro Instruction Reference*, GC26-4014 • *MVS/XA Integrated Catalog Administration: Access Method Services Reference*, GC26-4019 • *MVS/XA Interactive Problem Control System (IPCS) Planning and Customization*, GC28-1406 • *MVS/XA Interactive Problem Control System (IPCS) User's Guide*, GC28-1407 • *MVS/XA Interactive Problem Control System (IPCS) Command Reference*, GC28-1408 • *MVS/XA Message Library: System Codes*, GC28-1157 • *MVS/XA SPL: Service Aids*, GC28-1159 • *MVS/XA SPL: System Macros and Facilities*, Volumes 1 and 2, GC28-1150, GC28-1151 • *MVS/XA SPL: System Modifications*, GC28-1152 • *MVS/XA TSO Guide to Writing a Terminal Monitor Program (TMP) or a Command Processor*, GC28-1295 • *MVS/XA TSO Terminal Messages*, GC38-1046 • *MVS/XA VSAM Catalog Administration: Access Method Services Reference*, GC26-4075

ISG10001I **Analysis of global resources terminated. GQHT data unavailable.**

Explanation: The information used in the Global Queue Hash Table header to analyze the data is unavailable.

System Action: The system continues processing.

Programmer Response: None..

Problem Determination: Table I, items 5, 16, 29.

ISG10002I to ISG10009I

ISG10002I **Analysis of local resources terminated. LQHT data unavailable.**

Explanation: The information used in the local queue hash table header to analyze the data is unavailable.

System Action: The system continues processing.

Programmer Response: None.

Problem Determination: Table I, items 5, 16, 29.

ISG10003I **Default QCB processing limit used. RPT unavailable.**

Explanation: The default value for the Queue Control Block (QCB) process limit is 500. The system used this default limit because process limit in the Resource Pool Table (RPT) was unavailable. This limit is necessary to prevent infinite if there is an error in the QCB chain.

System Action: The system continues processing.

Programmer Response: None.

Problem Determination: Table I, items 5, 16, 29.

ISG10004I **Default QEL processing limit used. RPT unavailable.**

Explanation: The default value for the Queue Control Block (QCB) process limit is 500. The system used this default limit because the process limit in the resource pool table (RPT) was unavailable. This limit is necessary to prevent infinite looping in the event of an error in the QEL chain.

System Action: The system continues processing.

Programmer Response: None.

Problem Determination: Table I, items 5, 16, 29.

ISG10005I **Resource information truncated. Error in obtaining storage.**

Explanation: When obtaining storage to format resources, the GETMAIN macro instruction returned a non-zero return code.

System Action: The system continues processing.

Programmer Response: None.

Problem Determination: Table I, items 5, 16, 29.

ISG10006I **Resource analysis terminated. Error in sorting data.**

Explanation: When obtaining storage to format resources, the GRS DUMP Sort routine (ISGDSORT) returned a non-zero return code.

System Action: The system continues processing.

Programmer Response: None.

Problem Determination: Table I, items 5, 16, 29.

ISG10007I **GQHT unavailable.**

Explanation: The system issues this message when one of the following conditions is true:

- An I/O error occurred while the system attempted to read the Global Vector Table Extension (GVTX) control block which contains the address (GVTXGQHT) of the Global Queue Hash Table (GQHT).
- The GVTX control block containing the address of the GQHT was missing from the dump.
- The address of GQHT (GVTXGQHT) was zero.

System Action: The system continues processing.

Programmer Response: None.

Problem Determination: Table I, items 5, 16, 29.

ISG10008I **GRPT unavailable.**

Explanation: The system issues this message when either one of the following conditions is true:

- The system encountered an I/O error while attempting to read the Global Vector Table Extension (GVTX) control block which contains the address (GVTXGRPT) of the Global Resource Pool Table (GRPT).
- The GVTX control block, which contains the address of the GRPT, was missing from the dump.

System Action: The system continues processing.

Programmer Response: None.

Problem Determination: Table I, items 5, 16, 29.

ISG10009I **GRS analysis terminated. CVT unavailable.**

Explanation: The CVT was not available, because the caller of the dump analysis routine did not pass the address of the CVT in the ABDUMP Parameter List (ABDPL).

System Action: The system continues processing.

Programmer Response: None.

Problem Determination: Table I, items 5, 16, 29.

ISG10010I **GRS analysis terminated. GRS ASCB unavailable.**

Explanation: The system issues this message when either one of the following conditions is true:

- The system encountered an I/O error while attempting to read the Global Vector Table (GVT) control block which contains the address (GVTGASCB) of the GRS ASCB.
- The GVT control block, which contains the address of the ASCB, was missing from the dump.

System Action: The system continues processing.

Programmer Response: None..

Problem Determination: Table I, items 5, 16, 29.

ISG10011I **GRS analysis terminated. GRS ASID unavailable.**

Explanation: The system issues this message when either of the following conditions is true:

- The system encountered an I/O error while attempting to read the GRS ASCB which contains the ASID (ASCBASID) of the GRS address space.
- The ASCB control block, which contains the ASID, was missing from the dump.

System Action: The system continues processing.

Programmer Response: None..

Problem Determination: Table I, items 5, 16, 29.

ISG10012I **GRS analysis terminated. GVT unavailable.**

Explanation: The system issues this message when either of the following conditions is true:

- The system encountered an I/O error while attempting to read the Communication Vector Table (CVT) control block which contains the address (CVTGVT) of the Global Vector Table (GVT).
- The CVT control block, which contains the address of the GVT, was missing from the dump.

System Action: The system continues processing.

Programmer Response: None..

Problem Determination: Table I, items 5, 16, 29.

ISG10013I **GRS analysis terminated. GVTX unavailable.**

Explanation: The system issues this message when either one of the following conditions is true:

- The system encountered an I/O error while attempting to read the Global Vector Table (GVT) control block which contains the address (GVTGVTX) of the Global Vector Table Extension (GVTX).
- The GVT control block, which contains the address of the GVTX, was missing from the dump.

System Action: The system continues processing.

Programmer Response: None..

Problem Determination: Table I, items 5, 16, 29.

ISG10014I *xxx* **information truncated.** *xxx* **data at** *addr* **is not valid.**

Explanation: The system issues this message if one of the following conditions is true:

- The Queue Control Block (QCB) contained an RNAME length greater than decimal 256.
- The QCB had more than one scope bit turned on.
- The Queue Element Block (QEL) contained an ASID value equal to zero.

System Action: The system continues processing.

Programmer Response: None..

Problem Determination: Table I, items 5, 16, 29.

ISG10015I **LQHT unavailable.**

Explanation: The system issues this message when one of the following conditions is true:

- The system encountered an I/O error while attempting to read the Global Vector Table Extension (GVTX) control block which contains the address (GVTXLQHT) of the Local Queue Hash Table (LQHT).
- The GVTX control block, which contains the address of the LQHT, was missing from the dump.
- The address of LQHT (GVTXLQHT) was zero.

System Action: The system continues processing.

Programmer Response: None..

Problem Determination: Table I, items 5, 16, 29.

ISG10016I to ISG10024I

ISG10016I **LRPT unavailable.**

Explanation: The system issues this message when either one of the following conditions is true:

- The system encountered an I/O error while attempting to read the Global Vector Table Extension (GVTX) control block which contains the address (GVTXLRPT) of the Local Resource Pool Table (LRPT).
- The GVTX control block, which contains the address of the LRPT, was missing from the dump.

System Action: The system continues processing.

Programmer Response: None..

Problem Determination: Table I, items 5, 16, 29.

ISG10017I **No global ENQ resources allocated.**

Explanation: The system has no global resources allocated.

System Action: The system continues processing.

Programmer Response: None..

Problem Determination: Table I, items 5, 16, 29.

ISG10018I **No local ENQ resources allocated.**

Explanation: The system has no local resources allocated.

System Action: The system continues processing.

Programmer Response: None..

Problem Determination: Table I, items 5, 16, 29.

ISG10019I **no QELS anchored on** *qcb* **at** *addr*

Explanation: The ENQ/DEQ FRR routine (ISGGFRR0) detected queue damage for the Queue Control Block (QCB) synonym chain at *addr*.

System Action: The system continues processing.

Programmer Response: None..

Problem Determination: Table I, items 5, 16, 29.

ISG10020I **QCB information truncated. QCB chain damage at** *addr*.

Explanation: The pointer to the next Queue Control Block (QCB) was unavailable.

System Action: The system continues processing.

Programmer Response: None..

Problem Determination: Table I, items 5, 16, 29.

ISG10021I **QCB data on QCB synonym chain anchored at** *addr* **unavailable.**

Explanation: The system issues this message when either one of the following conditions is true:

- The Queue Control Block (QCB) address pointer within the QCB or within the Queue hash table (QHT) was unavailable
- There was no data in the Queue Element (QEL).

System Action: The system continues processing.

Programmer Response: None..

Problem Determination: Table I, items 5, 16, 29.

ISG10022I **QCB information truncated. QCB processing limit exceeded on QCB chain anchored at** *addr*.

Explanation: The Resource Pool Table (RPT) QCB processing limit has been exceeded. This condition could indicate this QCB chain was damaged causing an infinite loop.

System Action: The system continues processing.

Programmer Response: None..

Problem Determination: Table I, items 5, 16, 29.

ISG10023I *xxx* **synonym chain anchored at** *addr* **marked damaged.**

Explanation: During processing, the system detected damage to either the Queue Control Block (QCB) or Queue Element Block (QEL) chain.

System Action: The system continues processing.

Programmer Response: None..

Problem Determination: Table I, items 5, 16, 29.

ISG10024I **QEL information truncated. QEL chain damage at** *xxxxxxxx*.

Explanation: The system issues this message when either one of the following conditions is true:

- The pointer in the current Queue Element (QEL) to the next QEL control block was unavailable.
- The Queue Control Block (QCB) address pointer within the QEL was invalid.

System Action: The system continues processing.

Programmer Response: None..

Problem Determination: Table I, items 5, 16, 29.

ISG10025I **QEL data on QEL chain anchored at *addr* unavailable.**

Explanation: The system issues this message when either one of the following conditions is true:

- The Queue Element (QEL) address pointer within the Queue Control Block (QCB) was unavailable.
- There was no data in the Queue Element (QEL).

System Action: The system continues processing.

Programmer Response: None..

Problem Determination: Table I, items 5, 16, 29.

ISG10026I **QEL information truncated. QEL processing limit exceeded on QEL chain anchored at *addr*.**

Explanation: The Resource Pool Table (RPT) QEL processing limit has been exceeded. This condition could indicate this QEL chain was damaged causing an infinite loop.

System Action: The system continues processing.

Programmer Response: None..

Problem Determination: Table I, items 5, 16, 29.

ISG10027I **Unable to complete ENQ contention analysis.**

Explanation: An error occurred or data was unavailable in the dump, which prevented the ENQ analysis from reaching completion. Some indication of the problem should have preceded this message. Some ENQ contention data may have been generated.

System Action: The system continues processing.

Programmer Response: None..

Problem Determination: Table I, items 5, 16, 29.

ISG10028I **Event Tracing is inactive.**

Explanation: Event tracing is not available.

System Action: The system continues processing.

ISG10028I **RSA Tracing is inactive.**

Explanation: RSA tracing is not available.

System Action: The system continues processing.

ISG10050I **Formatting of the Trace Table terminated. The oldest entry in the Trace Table is unavailable.**

Explanation: The system cannot find the required entry in the Trace Table.

System Action: The system stops formatting the Trace Table.

ISG10051I **Formatting of the Trace Table terminated. Unsuccessful GETMAIN for the Trace Table buffer.**

Explanation: A GETMAIN for the Trace Table Buffer failed.

System Action: The system stops formatting the Trace Table.

ISG10052I **Formatting of the Trace Table terminated. The Trace Control Area data is unavailable.**

Explanation: The Trace Control Area data was not found.

System Action: The system stops formatting the Trace Table.

ISG10053I **Formatting of the Trace Table terminated. Invalid Trace Table data in the Trace Control Area.**

Explanation: The Trace Control Area contains invalid data.

System Action: The system stops formatting the Trace Table.

ISG10054I **Formatting of the Trace Table terminated. Trace Table data unavailable.**

Explanation: The Trace Table data was not found.

System Action: The system stops formatting the Trace Table.

ISG10055I **Formatting of the Trace Table terminated. An error occurred formatting RSA data.**

Explanation: Formating of RSA data was incomplete.

System Action: The system stops formatting the Trace Table.

ISG10056I **Formatting of the Trace Table terminated. An error occurred in formatting QWB data.**

Explanation: Formating of QWB data was incomplete.

System Action: The system stops formatting the Trace Table.

ISG10057I to ISG10059I

ISG10057I **Formatting of the Trace Table terminated. The Trace Table is empty.**

Explanation: There is no data in the Trace Table.

System Action: The system stops formatting the Trace Table.

ISG10058I **Formatting of the Trace Table terminated due to invalid data in the Trace Table header.**

Explanation: The Trace Table header contains invalid data.

System Action: The system stops formatting the Trace Table.

ISG10059I **Formatting of the Trace Table terminated. The Trace Table Entry data is invalid.**

Explanation: The data in the Trace Table is invalid.

System Action: The system stops formatting the Trace Table.

Data-In-Virtual Messages (ITV)

Component Name	Data-In-Virtual
Program Producing Message	Data-In-Virtual
Audience and Where Produced	For the user of IPCS: at the terminal and at the IPCS print file for some designated messages.
Message Format	ITVnnnnnt text ITV Component identifier nnnnn Message serial number. t Type code: A Action; the user must perform a specific action. D Decision; the user must choose an alternative. E Error; the user must correct the error before continuing. I Information; no user action is required. text Message text.
Associated and Referenced Publications	• *MVS/XA Data Administration: Macro Instruction Reference*, GC26-4014 • *MVS/XA Integrated Catalog Administration: Access Method Services Reference*, GC26-4019 • *MVS/XA Interactive Problem Control System (IPCS) Planning and Customization*, GC28-1406 • *MVS/XA Interactive Problem Control System (IPCS) User's Guide*, GC28-1407 • *MVS/XA Interactive Problem Control System (IPCS) Command Reference*, GC28-1408 • *MVS/XA Message Library: System Codes*, GC28-1157 • *MVS/XA SPL: Service Aids*, GC28-1159 • *MVS/XA SPL: Supervisor Services and Macro Instructions*, GC28-1154 • *MVS/XA SPL: System Modifications*, GC28-1152 • *MVS/XA TSO Guide to Writing a Terminal Monitor Program (TMP) or a Command Processor*, GC28-1295 • *MVS/XA TSO Terminal Messages*, GC38-1046 • *MVS/XA VSAM Catalog Administration: Access Method Services Reference*, GC26-4075

ITV10001I **No report keyword was specified. The default of EXCEPTION will be used.**

Explanation: The DETAIL, SUMMARY or EXCEPTION keyword was not specified in the DIVDATA subcommand. The system will use the EXCEPTION keyword as a default. See *MVS/XA: IPCS Command Reference* for a description of the DIVDATA subcommand.

System Action: The system continues processing.

Programmer Response: If you do not want the default keyword, respecify the keyword(s) for the DIVDATA subcommand and reissue the subcommand.

Problem Determination: None.

ITV10002I to ITV10009I

ITV10002I **No address space meets the selection criteria - no local Data-In-Virtual data processed**

Explanation: Data-in-virtual local data processing is on an address space basis. If no address space matches the user's request, the system will terminate processing of address space related data-in-virtual data and will not process any further local data-in-virtual data. This condition occurs because the requested address space is not in the dump.

System Action: The system continues processing.

Programmer Response: To determine the address spaces that were dumped, issue the 'SUMMARY JOBSUMMARY ALL' subcommand under IPCS. Make sure the requested address spaces are in the dump. If they are not, another dump must be taken.

Problem Determination: None.

ITV10003I **The ASID X'*asid*' (with ASCB at ADDRESS *dumpadd*) meets the following selection criteria:** *selection*

Explanation: *selection* identifies why the DIVDATA subcommand processed the address space *asid* at dump address *dumpadd*. The possible *selection* criteria are:

- All ASIDS were requested
- This ASID is a current ASID
- This ASID is an error ASID
- This ASID has a TCBERROR
- This ASID is in the ASIDLIST
- This ASID is in the JOBLIST

System Action: The system continues processing.

Programmer Response: None.

Problem Determination: None.

ITV10004I **No errors were detected for** *globallocal* **Data-In-Virtual data**

Explanation: Processing of the DIVDATA subcommand is successful. The message text indicates that the subcommand did not find any errors for the global/local data-in-virtual data described by *globallocal*.

System Action: The system continues processing.

Programmer Response: None.

Problem Determination: None.

ITV10005I **The** *qname* **for the** *cblock* **at ASID(X'*asid*') *dumpadd* is empty**

Explanation: The queue *qname* anchored in control block *cblock* at address *dumpadd* for address space *asid* is empty.

System Action: The system continues processing.

Programmer Response: None.

Problem Determination: None.

ITV10006I **The number of correctly queued DOAs of TYPE = *type* that could be accessed from the dump is** *num*

Explanation: The number of Data-in-virtual Object Access Control Blocks (DOAs) of type *type* that could be accessed from the dump and correctly queued is *num*.

System Action: The system continues processing.

Programmer Response: None.

Problem Determination: None.

ITV10007I **The number of correctly queued** *cblocks* **that could be accessed from the dump is** *num1*. **The number of** *cblocks* **indicated by** *cbfield* **is** *num2*.

Explanation: *cbfield* indicates that there are *num2* queued *cblocks*. Only *num1* of the *cblocks* can be accessed from the dump and are correctly queued.

System Action: The system continues processing.

Programmer Response: None.

Problem Determination: None.

ITV10008I **Errors were detected while processing DIVDATA subcommand**

Explanation: The DIVDATA subcommand detected errors while processing the EXCEPTION keyword.

System Action: The system continues processing.

Programmer Response: If the subcommand detected the errors in data-in-virtual control blocks, run DETAIL report to show the contents of the data-in-virtual control blocks.

Problem Determination: None.

ITV10009I **No address space keyword was specified. The default of CURRENT and ERROR will be used.**

Explanation: If ALL, CURRENT, ERROR, TCBERROR, ASIDLIST or JOBLIST is not specified in the DIVDATA subcommand, the system uses the default keywords CURRENT and ERROR. See *MVS/XA: IPCS Command Reference* for a description of the DIVDATA subcommand.

System Action: The system continues processing.

Programmer Response: If you do not want the default keywords CURRENT and ERROR, respecify the keyword(s) for DIVDATA subcommand and reissue the subcommand.

Problem Determination: None.

ITV10011I **The *dataarea* AT ASID(X'*asid*') *dumpadd reason***

Explanation: The system encountered an error while processing *dataarea* at address *dumpadd* for the specified address space *asid*. *reason* may be one of the following:

has a back pointer that does not point to the previous element on the queue
The *dataarea* has a back pointer that does not point to the previous element on the queue.

is queued to the wrong TCB
The *dataarea* points to a TCB that is not the current TCB.

is queued to the wrong DOA
The *dataarea* points to a Data-in-virtual Object Access Control Block (DOA) that is not the current DOA.

System Action: The system continues processing.

Programmer Response: None.

Problem Determination: Table I, items 16, 29.

ITV10012I **DOA type is not valid**

Explanation: The Data-in-virtual Object Access Control Block (DOA) type code is not valid. For a type DA data-in-virtual object, the valid type code should be 1.

System Action: The system continues processing.

Programmer Response: None

Problem Determination: Table I, items 16, 29

ITV10013I **MMSID is not equal to C'MMSP'**

Explanation: The Media Manager Parameter List (MMSPL) identifier field MMSID is not valid.

System Action: The system continues processing.

Programmer Response: None.

Problem Determination: Table I, items 16, 29

ITV10014I **Fields DOAMMIB and MMSMMIB are not equal**

Explanation: The Data-in-virtual Object Access Control Block (DOA) points to a Media Manager Interface Block (MMIB) that is different from the MMIB that the Media Manager Parameter List points to.

System Action: The system continues processing.

Programmer Response: None.

Problem Determination: Table I, items 16, 29

ITV10101I **Data-In-Virtual tracing was turned off as a result of recovery processing**

Explanation: Prior to this dump being taken, the following situation occurred:

1. Data-in-virtual recovery was entered and could not establish addressability to the original data-in-virtual trace table.
2. Data-in-virtual tracing then set up a new trace table.
3. Subsequently, when data-in-virtual recovery was entered, addressability could not be established to the new table.

System Action: The DIVDATA IPCS exit data-in-virtual trace table processing terminates.

Programmer Response: The data-in-virtual trace tables may have been dumped in the SQA/ESQA portion of the dump. Look for the data-in-virtual trace table header which contains: '**** COMPONENT TRACE TABLE **** - COMPONENT= SCDIV'. If a data overlay caused this problem, you may not find the trace table header.

Problem Determination: Table I, item 29.

ITV10102I **Data-In-Virtual tracing was turned off as a result of a GETMAIN failure for the Data-In-Virtual trace control block or the trace table**

Explanation: The dump does not contain a data-in-virtual Component Trace Table Control Area (CTC). The dump does not contain a data-in-virtual CTC because of a GETMAIN failure for either the CTC or the data-in-virtual trace table prior to the dump being taken.

System Action: The DIVDATA IPCS exit data-in-virtual trace table processing terminates. Prior to the dump being taken, the system issued message ITV001I to the console at the time of the GETMAIN failure.

Programmer Response: None.

Problem Determination: Table I, items 29,54.

ITV10103I to ITV10111I

ITV10103I **Data-In-Virtual tracing was active but there is no trace data available due to recovery processing**

Explanation: Prior to this dump being taken, a data-in-virtual trace table was built but the data-in-virtual recovery routine could not establish addressability to it.

System Action: The DIVDATA IPCS exit data-in-virtual trace table processing terminates.

Programmer Response: The previously used data-in-virtual trace table may have been dumped in the SQA/ESQA portion of the dump. Look for the data-in-virtual trace table header which contains: '**** COMPONENT TRACE TABLE **** - COMPONENT= SCDIV'. If a data overlay problem caused this error, you may not find the trace table header.

Problem Determination: Table I, item 29.

ITV10104I **The Data-In-Virtual trace table was not built yet**

Explanation: The data-in-virtual trace table is built when the first trace entry is created. Tracing was requested prior to the dump being taken, but no trace entries were built because no data-in-virtual activity matched the event that the user specified for tracing.

System Action: The DIVDATA IPCS exit data-in-virtual trace table processing terminates.

Programmer Response: None.

Problem Determination: None.

ITV10105I **Data-In-Virtual tracing was not active**

Explanation: The Data-in-virtual Information Block Extension (DIBX) contains information on whether tracing was active when the system was running. Data-in-virtual tracing was not active at the time the dump was taken.

System Action: The DIVDATA IPCS exit data-in-virtual trace table processing terminates.

Programmer Response: None.

Problem Determination: None.

ITV10106I **The trace table has been replaced as a result of recovery processing, previous data may have been lost**

Explanation: Prior to this dump being taken, data-in-virtual recovery could not establish addressability to the data-in-virtual trace table. As a result, that trace table is no longer available. Any subsequent data-in-virtual trace table entries were put in a new table. The system is currently processing that new table.

System Action: The DIVDATA IPCS exit data-in-virtual trace table processing continues.

Programmer Response: The previously used data-in-virtual trace table may have been dumped in the SQA/ESQA portion of the dump. Look for the data-in-virtual trace table header which contains: '**** COMPONENT TRACE TABLE **** - COMPONENT= SCDIV'. If a data overlay problem caused the error, you may not find a trace table header.

Problem Determination: Table I, item 29.

ITV10107I **No address space meets the selection criteria - the Data-In-Virtual trace table is not processed**

Explanation: No address spaces in the dump match the address spaces that the user requested for processing.

System Action: The DIVDATA IPCS exit data-in-virtual trace table processing terminates.

Programmer Response: Specify 'DIVDATA FULLTRACE' in order to format the entire trace table to determine what the ASIDs were for each entry. If data-in-virtual trace table data is needed for a specific address space, verify that the required address spaces are specified correctly in the DIVDATA invocation.

Problem Determination: None.

ITV10108I **Error events were not being traced**

Explanation: The Data-in-virtual Information Block Extension (DIBX) indicates that error event tracing was not active when the system was running. An EXCEPTION report was requested, which prints out information pertaining only to the error events.

System Action: The DIVDATA IPCS exit data-in-virtual trace table processing terminates for the EXCEPTION report.

Programmer Response: None.

Problem Determination: None.

ITV10111I **All or part of a Component Trace Table Control Area at *dumpadd* could not be accessed, no further trace table processing**

Explanation: The data-in-virtual Component Trace Table Control Area (CTC) which contains pertinent information about the data-in-virtual trace table, including the address of the table. The CTC starting at the dump address *dumpadd* could not be accessed in its entirety because it was either not dumped or the IPCS Storage Access Service Routine was unable to access it.

System Action: The DIVDATA IPCS exit data-in-virtual trace table processing terminates.

Programmer Response: Look at the storage ranges that were dumped. If the part of SQA/ESQA containing the CTC was not dumped, the IPCS Storage Access Service

Routine could not access the control block. If data-in-virtual trace table processing is required, another dump is needed that specifies the appropriate storage ranges in SQA/ESQA for the CTC and the data-in-virtual trace table.

Problem Determination: Table I, item 29.

ITV10112I **The storage at *dumpadd* was not a valid Component Trace Table Control Area, no further trace table processing**

Explanation: The data-in-virtual Component Trace Table Control Area (CTC) which contains pertinent information about the data-in-virtual trace table, including the address of the table. The CTC is divided into 4 sections. The CTC at the dump address *dumpadd* is not valid in this dump because either the CTC ID was not correct or the address of a section in the CTC did not match the address in the Data-in-virtual Information Block Extension (DIBX).

System Action: The DIVDATA IPCS exit data-in-virtual trace table processing terminates.

Programmer Response: Determine why the storage at that address was considered to be a CTC.

Problem Determination: Table I, item 29.

ITV10113I **The [following] Data-In-Virtual trace entry at *dumpadd* does not contain a valid length field for the length of the entry, there is no further trace table processing [*: hexdump*]**

Explanation: The length field for the data-in-virtual trace table entry at the dump address *dumpadd* was larger than the size of the work area used when creating the entry. This is probably due to a system storage overlay error.

When the user requests DETAIL, TRACE OR FULLTRACE processing the system will issue this message with a hex dump **hexdump** for the length of the work area.

When the user requests SUMMARY or EXCEPTION processing a hex dump will not follow this message.

When *hexdump* accompanies the message, the invalid length field may be either the first or last word of the hex dump.

- If the incorrect length field is the first word of the hex dump, the system will display up to 1024 bytes of the trace table preceding the first word.
- If the incorrect length field is the last word of the hex dump, the system will display up to 1024 bytes of the trace table preceding the last word.

If the system displays less than 1024 bytes of the trace table, this indicates that the system reached the trace table boundary.

System Action: The DIVDATA IPCS exit data-in-virtual trace table processing terminates, but the data-in-virtual trace table statistical report is processed.

Programmer Response: Determine why the length field was set to an incorrect value.

Problem Determination: Table I, item 29.

ITV10114I **The [following] Data-In-Virtual trace entry at *dumpadd* has an invalid event class value of *num* [*:hexdump*]**

Explanation: The data-in-virtual trace event at the dump address *dumpadd* is not one of the defined event classes. This is probably due to a system storage overlay error. A complete list of the defined classes and events is in the data-in-virtual section of the System Logic Library.

When the user requests DETAIL, TRACE OR FULLTRACE processing the system will issue this message with a hex dump *hexdump* for the length of the work area.

When the user requests SUMMARY or EXCEPTION processing a hex dump will not follow this message.

System Action: The DIVDATA IPCS exit data-in-virtual trace table processing continues.

Programmer Response: Determine why the event class field was set to an incorrect value.

Problem Determination: Table I, item 29.

ITV10115I **Unable to load models to format parts of the Data-In-Virtual trace entry, there is no further Data-In-Virtual variable information processing**

Explanation: The loading of the format models failed because either the modules containing the models were not found or there was an I/O error.

System Action: The DIVDATA IPCS exit data-in-virtual trace table processing continues with only the base portion of the trace entries formatted.

Programmer Response: If the user's profile allows write to programmer messages (PROFILE WTPMSG), the user will see a message from IPCS related to the load failure. If the model was not found, ask the system programmer to make sure that the appropriate modules are included in SYS1.LINKLIB for data-in-virtual IPCS exit formatting. Refer to the data-in-virtual SYSGEN macro for the complete list of the modules needed.

Problem Determination: Table I, items 29,54.

ITV10116I

ITV10116I **The [following] Data-In-Virtual trace entry at *dumpadd* has an invalid event code of *num*: [hexdump]**

Explanation: The event code *num* for the trace table entry was not a value that could be translated into one of the currently defined CLASS/EVENT names. This is probably due to a system storage overlay error. A complete list of the defined events is in the data-in-virtual section of the System Logic Library.

If the user requested DETAIL, TRACE or FULLTRACE processing, the system issues this message with a hex dump for the length of the work area.

If the user requested SUMMARY or EXCEPTION processing, a hex dump does not accompany this message.

If the user requested TRACE, SUMMARY or EXCEPTION processing, the system issues this message only when the requested ASIDs match the ASID in the trace table entry.

System Action: The DIVDATA IPCS exit data-in-virtual trace table processing continues formatting the trace entry.

Programmer Response: Determine why the event code was set to an incorrect value.

Problem Determination: Table I, item 29.

Unnumbered Messages Issued By Auxiliary Storage Manager (ASM)

ABNORMAL I/O COMPLETION

Explanation I/O did not complete normally.

ABOVE PCCW HAS PREVIOUSLY BEEN FORMATTED. POSSIBLE LOOP IN PCCW QUEUE, OR PCCW RESIDES ON MORE THAN ONE PCCW QUEUE. FORMATTING OF THIS PCCW QUEUE IS TERMINATED.

Explanation This message appears when the same PCCW has been formatted more than once.

ABOVE SCCW HAS PREVIOUSLY BEEN FORMATTED. POSSIBLE LOOP IN SCCW QUEUE, OR SCCW RESIDES ON MORE THAN ONE QUEUE. FORMATTING OF THIS SCCW QUEUE IS TERMINATED.

Explanation The same SCCW was formatted more than once.

AIA ANCHORED IN AVAILABLE PCCW - PAGE I/O REQUEST MAY HAVE BEEN LOST

Explanation An AIA is erroneously anchored in an available PCCW.

AIA IS INACCESSIBLE IN THE DUMP DATASET.
FORMATTING OF AIA IS TERMINATED.

AIA IS INACCESSIBLE IN THE DUMP DATASET.
FORMATTING OF AIA QUEUE IS TERMINATED.

Explanation: Either PRDMP was not able to access the control block in the dump dataset or the dump dataset was incomplete and did not contain the control block. The control block and those control blocks anchored in it are not formatted.

AIAS ANCHORED IN AVAILABLE SCCW - SWAP I/O REQUESTS MAY HAVE BEEN LOST

Explanation This message appears when an AIA queue is erroneously anchored in an available SCCW.

ASMVT AND 480 BYTES PRECEDING AND FOLLOWING ASMVT IN HEX AND EBCDIC.

ASMVT DOES NOT CONTAIN ASMVTID - POSSIBLE STORAGE OVERLAY.
FORMATTING OF ASMVT, ERROR RECORD, MESSAGE BUFFER, PAGING AND SWAPPING RELATED CONTROL BLOCKS IS TERMINATED.

ASMVT DOES NOT CONTAIN ASMVTID - POSSIBLE STORAGE OVERLAY.
WILL FORMAT AND PRINT ASMVT IN HEX AND EBCDIC. NO ATTEMPT TO FORMAT ERROR RECORD, MESSAGE BUFFER, BPF CACHE ARRAY, PAGING AND SWAPPING RELATED CONTROL BLOCKS.

Explanation: The control block ID is invalid. It is possible that all or portions of the control block was overlaid. In addition to formatting the control block, the unformatted control block plus 480 bytes preceding and following the area are printed in hex and in EBCDIC. No attempt is made to format those control blocks anchored in the control block.

ASMVT IS NOT ACCESSIBLE - FORMATTING OF ASMVT, ERROR RECORD, MESSAGE BUFFER, PAGING AND SWAPPING RELATED CONTROL BLOCKS IS TERMINATED.

ASMVT IS NOT ACCESSIBLE IN THE DUMP DATASET.
FORMATTING OF ASMVT, ERROR RECORD, MESSAGE BUFFER, BPF CACHE ARRAY, PAGING AND SWAPPING RELATED CONTROL BLOCKS IS TERMINATED.

Explanation: Either PRDMP could not access the control block in the dump data set or the dump data set was incomplete and did not contain the control block. The control block and those control blocks anchored in it are not formatted.

BACK UP OF PAGING REQUESTS - THE NUMBER OF REQUESTS STARTED BUT NOT YET COMPLETE EXCEEDS THE THRESHOLD.

Explanation The back up of paging requests exceeds 40.

BACK UP OF SWAPPING REQUESTS - THE NUMBER OF REQUESTS STARTED BUT NOT YET COMPLETE EXCEEDS THE THRESHOLD.

Explanation The back up of swapping requests exceeds 40.

BAD CHANNEL STATUS

Explanation Paging I/O with bad channel status.

BAD DEVICE STATUS

Explanation Paging I/O with bad device status.

BAD SENSE DATA

Explanation Paging I/O with bad sense data.

CACHE ARRAY AND 480 BYTES PRECEDING AND FOLLOWING THE CONTROL BLOCK IN HEX AND EBCDIC.

CACHE ARRAY DOES NOT CONTAIN ILRCACHE ID -
POSSIBLE STORAGE OVERLAY.
WILL FORMAT AND PRINT CACHE ARRAY IN HEX AND EBCDIC.

Explanation: The control block ID is invalid. It is possible that all or portions of the control block was overlaid. In addition to formatting the control block, the unformatted control block plus 480 bytes preceding and following the area are printed in hex and in EBCDIC. No attempt is made to format those control blocks anchored in the control block.

CACHE ARRAY IS INACCESSIBLE IN THE DUMP DATASET.
FORMATTING OF THE CACHE ARRAY IS TERMINATED.

COMMON PAGE DATASET WAS MARKED BAD

Explanation The page dataset was marked bad.

DEIB ENTRY IS INACCESSIBLE IN THE DUMP DATA SET -
FORMATTING OF ALL DEIB ENTRIES IS TERMINATED.

Explanation: Either PRDMP could not access the control block in the dump data set or the dump data set was incomplete and did not contain the control block. The control block and those control blocks anchored in it are not formatted.

DEIB HEADER AND 480 BYTES PRECEDING AND FOLLOWING DEIB IN
HEX AND EBCDIC.

DEIB HEADER DOES NOT CONTAIN THE DEIBID - POSSIBLE STORAGE OVERLAY.
FORMATTING OF THE DEIB HEADER AND THE DEIB ENTRIES IS
TERMINATED.

DEIB HEADER DOES NOT CONTAIN THE DEIBID - POSSIBLE STORAGE OVERLAY.
WILL FORMAT AND PRINT DEIB HEADER AND 480 BYTES PRECEDING AND
FOLLOWING DEIB HEADER IN HEXADECIMAL AND EBCDIC. NO ATTEMPT
TO FORMAT DEIB ENTRIES.

Explanation: The control block ID is invalid. It is possible that all or portions of the control block was overlaid. In addition to formatting the control block, the unformatted control block plus 480 bytes preceding and following the area are printed in hex and in EBCDIC. No attempt is made to format those control blocks anchored in the control block.

DEIB HEADER IS INACCESSIBLE IN THE DUMP DATA SET.
FORMATTING OF THE DEIB HEADER AND THE DEIB ENTRIES IS
TERMINATED.

Explanation: Either PRDMP could not access the control block in the dump data set or the dump data set was incomplete and did not contain the control block. The control block and those control blocks anchored in it are not formatted.

DUPLEX PAGE DATASET WAS MARKED BAD

Explanation The page dataset was marked bad.

END OF ASM DATA

Explanation: Indicates that the ASM format routines have successfully completed their processing and formatted those control blocks that were accessible.

EREC hhhhhhhh ANCHORED IN ASMEREC.

Explanation: Displays the address of the EREC (error record) in the dump data set.

EREC IS INACCESSIBLE IN THE DUMP DATASET.
FORMATTING OF EREC IS TERMINATED.

Explanation: Either PRDMP could not access the control block in the dump dataset or the dump dataset was incomplete and did not contain the control block. The control block and those control blocks anchored in it are not formatted.

ERROR ENCOUNTERED DURING FORMATTING OF AIA -
FORMATTING OF AIA QUEUE IS TERMINATED.

ERROR ENCOUNTERED DURING FORMATTING OF ASMVT -
FORMATTING OF ASMVT IS TERMINATED

ERROR ENCOUNTERED DURING FORMATTING OF ASMVT AND 480 BYTES
PRECEDING AND FOLLOWING THE ASMVT - FORMATTING OF THIS ASMVT
IS TERMINATED.

ERROR ENCOUNTERED DURING FORMATTING OF CACHE ARRAY -
FORMATTING OF CACHE ARRAY IS TERMINATED
ERROR ENCOUNTERED DURING FORMATTING OF CACHE ARRAY AND 480 BYTES
PRECEDING AND FOLLOWING THE CONTROL BLOCK -
FORMATTING OF CACHE ARRAY IS TERMINATED

ERROR ENCOUNTERED DURING FORMATTING OF DEIB HEADER -
FORMATTING OF THIS DEIB HEADER IS TERMINATED.

ERROR ENCOUNTERED DURING FORMATTING OF DEIB HEADER AND 480 BYTES
PRECEDING AND FOLLOWING DEIB HEADER - FORMATTING OF THIS
DEIB HEADER IS TERMINATED.

ERROR ENCOUNTERED DURING FORMATTING OF EREC -
FORMATTING OF EREC IS TERMINATED.

ERROR ENCOUNTERED DURING FORMATTING OF IORB.
FORMATTING OF IORB IS TERMINATED.

ERROR ENCOUNTERED DURING FORMATTING OF IORB AND 480 BYTES
PRECEDING AND FOLLOWING THE IORB - FORMATTING OF THIS IORB IS
TERMINATED. WILL ATTEMPT TO FORMAT THE IOSB-SRB-SRB CHAIN

ERROR ENCOUNTERED DURING FORMATTING OF IORB -
FORMATTING OF IORB IS TERMINATED.
WILL ATTEMPT TO FORMAT THE IOSB-SRB-SRB CHAIN
AND SCCW/AIA QUEUE ANCHORED IN THE IORB.

ERROR ENCOUNTERED DURING FORMATTING OF IOSB AND 480 BYTES
PRECEDING AND FOLLOWING IOSB. FORMATTING OF IOSB IS
TERMINATED.

ERROR ENCOUNTERED DURING FORMATTING OF IOSB -
FORMATTING OF IOSB IS TERMINATED.
WILL ATTEMPT TO FORMAT THE SRB ANCHORED IN THE IOSB.

ERROR ENCOUNTERED DURING FORMATTING OF MSGBF,
FORMATTING OF MSGBF IS TERMINATED.
ERROR ENCOUNTERED DURING FORMATTING OF PART AND 480 BYTES
PRECEDING AND FOLLOWING THE PART. FORMATTING OF THIS PART
IS TERMINATED.

ERROR ENCOUNTERED DURING FORMATTING OF PART HEADER -
FORMATTING OF PART HEADER IS TERMINATED

ERROR ENCOUNTERED DURING FORMATTING OF PARTE -
FORMATTING OF PARTE IS TERMINATED.
WILL ATTEMPT TO FORMAT THE CONTROL BLOCKS
ANCHORED IN THE PARTE.

ERROR ENCOUNTERED DURING FORMATTING OF THE PAT HEADER.
FORMATTING OF THE PAT HEADER AND PATMAP IS TERMINATED.

ERROR ENCOUNTERED DURING FORMATTING OF THE PATMAP -
FORMATTING OF PATMAP IS TERMINATED.

ERROR ENCOUNTERED DURING FORMATTING OF PCCW -
FORMATTING OF PCCW QUEUE AND THE
AIA ANCHORED IN THE PCCW IS TERMINATED.

ERROR ENCOUNTERED DURING FORMATTING OF PCCW -
FORMATTING OF THIS PCCW IS TERMINATED.
ERROR ENCOUNTERED DURING FORMATTING OF PCCW AND 480 BYTES
PRECEDING AND FOLLOWING THE PCCW - FORMATTING OF THIS PCCW
IS TERMINATED.

ERROR ENCOUNTERED DURING FORMATTING OF PCT -
FORMATTING OF PCT QUEUE IS TERMINATED.

ERROR ENCOUNTERED DURING FORMATTING OF PCT AND 480 BYTES
PRECEDING AND FOLLOWING THE PCT - FORMATTING OF THIS PCT
IS TERMINATED.

ERROR ENCOUNTERED DURING FORMATTING OF SARTHDR AND 480 BYTES
PRECEDING AND FOLLOWING THE CONTROL BLOCK -
FORMATTING OF THIS SARTHDR IS TERMINATED

ERROR ENCOUNTERED DURING FORMATTING OF SART HEADER -
FORMATTING OF SART HEADER IS TERMINATED.

ERROR ENCOUNTERED DURING FORMATTING OF SARTE -
FORMATTING OF SARTE IS TERMINATED.
WILL ATTEMPT TO FORMAT THE CONTROL BLOCKS
ANCHORED IN THIS SARTE.

ERROR ENCOUNTERED DURING FORMATTING OF THE SATHDR AND 480 BYTES
PRECEDING AND FOLLOWING THE SAT HEADER -
FORMATTING OF THIS SAT HEADER IS TERMINATED.

ERROR ENCOUNTERED DURING FORMATTING OF THE SAT HEADER.
FORMATTING OF THE SAT HEADER AND SATMAP IS TERMINATED.

ERROR ENCOUNTERED DURING FORMATTING OF THE SATMAP -
FORMATTING OF SATMAP IS TERMINATED.

ERROR ENCOUNTERED DURING FORMATTING OF SCCW AND 480 BYTES
PRECEDING AND FOLLOWING THE SCCW - FORMATTING OF THE SCCW
IS TERMINATED.
ERROR ENCOUNTERED DURING FORMATTING OF SCCW -
FORMATTING OF THE SCCW QUEUE IS TERMINATED.

ERROR ENCOUNTERED DURING FORMATTING OF SCCW -
FORMATTING OF THE SCCW QUEUE AND THE
AIAS ANCHORED IN EACH SCCW IS TERMINATED.

ERROR ENCOUNTERED DURING FORMATTING OF SDCT AND 480 BYTES
PRECEDING AND FOLLOWING THE SDCT - FORMATTING OF THIS SDCT
IS TERMINATED.

ERROR ENCOUNTERED DURING FORMATTING OF SDCT ENTRY -
FORMATTING OF ALL SDCT ENTRIES IS TERMINATED.

ERROR ENCOUNTERED DURING FORMATTING OF SDCT HEADER -
FORMATTING OF SDCT HEADER IS TERMINATED. WILL ATTEMPT
TO FORMAT THE SDCT ENTRIES.

ERROR ENCOUNTERED DURING FORMATTING OF SRB -
FORMATTING OF SRB AND AIA QUEUE ANCHORED IN THE SRB
IS TERMINATED.

ERROR ENCOUNTERED DURING FORMATTING OF SRB AND 480 BYTES
PRECEDING AND FOLLOWING THE SRB -
FORMATTING OF THIS SRB IS TERMINATED.

ERROR ENCOUNTERED DURING FORMATTING OF THE DEIB ENTRY -
FORMATTING OF ALL DEIB ENTRIES IS TERMINATED.

ERROR ENCOUNTERED DURING FORMATTING OF THE DEIB HEADER -
FORMATTING OF THE DEIB HEADER IS TERMINATED.
WILL ATTEMPT TO FORMAT THE DEIB ENTRIES.

ERROR ENCOUNTERED DURING FORMATTING OF THE PCT SECTOR
VALUE TABLE -
FORMATTING OF THE PCT SECTOR VALUE TABLE IS TERMINATED.

Explanation: During the formatting of the control block, PRDMP encountered an error. The process was terminated.

FORMAT OF ASM DATA

Explanation: Title statement printed at the top of the dump before any formatting begins.

IORB AND 480 BYTES PRECEDING AND FOLLOWING THE IORB IN HEX AND EBCDIC

IORB DOES NOT CONTAIN THE IORBID - POSSIBLE STORAGE OVERLAY. FORMATTING OF EACH IORB-IOSB-SRB-SRB CHAIN AND THE SCCW/AIA QUEUE ANCHORED IN THE IORB IS TERMINATED.

IORB DOES NOT CONTAIN THE IORBID - POSSIBLE STORAGE OVERLAY. WILL FORMAT IORB AND 480 BYTES PRECEDING AND FOLLOWING IORB IN HEX AND EBCDIC. NO ATTEMPT TO FORMAT IOSB-SRB-SRB CHAIN AND THE SCCW/AIA QUEUE ANCHORED IN IORB.

Explanation: The control block ID is invalid. It is possible that all or portions of the control block was overlaid. In addition to formatting the control block, the unformatted control block plus 480 bytes preceding and following the area are printed in hex and in EBCDIC. No attempt is made to format those control blocks anchored in the control block.

IORB IS INACCESSIBLE IN THE DUMP DATA SET. FORMATTING OF EACH IORB-IOSB-SRB-SRB CHAIN AND THE PCCW/AIA QUEUE ANCHORED IN THE IORB IS TERMINATED.

Explanation: Either PRDMP could not access the control block in the dump dataset or the dump dataset was incomplete and did not contain the control block. The control block and those control blocks anchored in it are not formatted.

LOCAL PAGE DATASET WAS MARKED BAD

Explanation: The page dataset was marked bad.

IOSB AND 480 BYTES PRECEDING AND FOLLOWING IOSB IN HEX AND EBCDIC

IOSB IS INACCESSIBLE IN THE DUMP DATA SET. FORMATTING OF THE IOSB AND THE SRB ANCHORED IN THE IOSB IS TERMINATED. WILL ATTEMPT TO FORMAT THE SRB ANCHORED IN THE IORB.

MSGBF IS INACCESSIBLE IN THE DUMP DATASET. FORMATTING OF MSGBF IS TERMINATED.

Explanation: Either PRDMP could not access the control block in the dump dataset or the dump dataset was incomplete and did not contain the control block. The control block and those control blocks anchored in it are not formatted.

MSGBF hhhhhhhh ANCHORED IN ASMMSGBF.

Explanation: Displays the address of the MSGBF (message buffer) in the dump dataset.

nnnnnn PCCWS HAVE NOT BEEN FORMATTED - PCCWS AND I/O REQUESTS MAY HAVE BEEN LOST

Explanation One or more PCCWs are not found on any ASM queue. nnnnnn is the number of PCCWs not found.

nnnnnn SCCWS HAVE NOT BEEN FORMATTED - SCCWS AND I/O REQUESTS MAY HAVE BEEN LOST

Explanation One or more SCCWs is not found on any ASM queue. nnnnnn is the number of SCCWs not found.

PART ADDRESS IS ZERO - POSSIBLE STORAGE OVERLAY IN THE ASMVT.
FORMATTING OF THE PART, PCTS, NO PCCW AIA QUEUE AND THE CONTROL BLOCKS ANCHORED IN EACH PARTE IS TERMINATED.

PART AND 480 BYTES PRECEDING AND FOLLOWING PART IN HEX AND EBCDIC.

PART DOES NOT CONTAIN THE PARTID - POSSIBLE STORAGE OVERLAY.
FORMATTING OF THE PART, PCTS, NO PCCW AIA QUEUE AND THE CONTROL BLOCKS ANCHORED IN EACH PARTE IS TERMINATED.

PART DOES NOT CONTAIN THE PARTID - POSSIBLE STORAGE OVERLAY.
WILL FORMAT AND PRINT PART HEADER AND 480 BYTES PRECEDING AND FOLLOWING PART HEADER IN HEX AND EBCDIC. NO ATTEMPT TO FORMAT PCTS, PCCW/AIA QUEUE AND THE CONTROL BLOCKS ANCHORED IN EACH PARTE.

Explanation: The control block ID is invalid. It is possible that all or portions of the control block was overlaid. In addition to formatting the control block, the unformatted control block plus 480 bytes preceding and following the area are printed in hex and in EBCDIC. No attempt is made to format those control blocks anchored in the control block.

PART HEADER IS INACCESSIBLE IN THE DUMP DATA SET.
FORMATTING OF THE PART, PCTS, NO PCCW AIA QUEUE AND THE CONTROL BLOCKS ANCHORED IN EACH PARTE IS TERMINATED.

PARTE IS INACCESSIBLE IN THE DUMP DATA SET.
FORMATTING OF THIS PARTE AND THE CONTROL BLOCKS ANCHORED IN THIS PARTE IS TERMINATED. WILL ATTEMPT TO FORMAT THE REMAINING PARTES.

Explanation: Either PRDMP could not to access the control block in the dump data set or the dump data set was incomplete and did not contain the control block. The control block and those control blocks anchored in it are not formatted.

PAT AND 480 BYTES PRECEDING AND FOLLOWING PAT IN HEX AND EBCDIC.

PAT DOES NOT CONTAIN THE PAT ID - POSSIBLE STORAGE OVERLAY.
FORMATTING OF THE PAT HEADER AND THE PATMAP IS TERMINATED.

PAT DOES NOT CONTAIN THE PAT ID - POSSIBLE STORAGE OVERLAY.
WILL FORMAT AND PRINT PAT HEADER AND 480 BYTES PRECEDING AND FOLLOWING PAT HEADER IN HEXADECIMAL AND EBCDIC. NO ATTEMPT TO FORMAT PATMAP.

Explanation: The control block ID is invalid. It is possible that all or portions of the control block was overlaid. In addition to formatting the control block, the unformatted control block plus 480 bytes preceding and following the area are printed in hex and in EBCDIC. No attempt is made to format those control blocks anchored in the control block.

PAT IS INACCESSIBLE IN THE DUMP DATA SET.
FORMATTING OF THE PAT HEADER AND PATMAP IS TERMINATED.

Explanation: Either PRDMP could not access the control block in the dump data set or the dump data set was incomplete and did not contain the control block. The control block and those control blocks anchored in it are not formatted.

PATMAP

PATMAP IS INACCESSIBLE IN THE DUMP DATA SET.
FORMATTING OF THE PATMAP IS TERMINATED.

Explanation: Either PRDMP could not access the control block in the dump data set or the dump data set was incomplete and did not contain the control block. The control block and those control blocks anchored in it are not formatted.

PCCW AND 480 BYTES PRECEDING AND FOLLOWING PCCW IN HEX AND EBCDIC.

PCCW DOES NOT CONTAIN THE PCCWID - POSSIBLE STORAGE OVERLAY.
FORMATTING OF THE PCCW QUEUE AND THE AIA ANCHORED IN THE PCCW IS TERMINATED.
PCCW DOES NOT CONTAIN THE PCCWID - POSSIBLE STORAGE OVERLAY.
WILL FORMAT AND PRINT PCCW AND 480 BYTES PRECEDING AND FOLLOWING PCCW IN HEXADECIMAL AND EBCDIC. NO ATTEMPT TO FORMAT THIS PCCW AND AIA ANCHORED IN THE PCCW.

Explanation: The control block ID is invalid. It is possible that all or portions of the control block was overlaid. In addition to formatting the control block, the unformatted control block plus 480 bytes preceding and following the area are printed in hex and in EBCDIC. No attempt is made to format those control blocks anchored in the control block.

PCCW IS INACCESSIBLE IN THE DUMP DATA SET.
FORMATTING OF THE PCCW QUEUE AND THE AIA ANCHORED IN THE PCCW IS TERMINATED.

Explanation: Either PRDMP could not access the control block in the dump data set or the dump data set was incomplete and did not contain the control block. The control block and those control blocks anchored in it are not formatted.

PCT AND 480 BYTES PRECEDING AND FOLLOWING THE PCT IN HEX AND EBCDIC

PCT DOES NOT CONTAIN PCTID - POSSIBLE STORAGE OVERLAY.
FORMATTING OF PCT QUEUE IS TERMINATED.

PCT DOES NOT CONTAIN PCTID - POSSIBLE STORAGE OVERLAY.
WILL FORMAT AND PRINT PCT AND 480 BYTES PRECEDING AND FOLLOWING PCT IN HEXADECIMAL AND EBCDIC. NO ATTEMPT TO FORMAT PCT SECTOR VALUE TABLE

Explanation: The control block ID is invalid. It is possible that all or portions of the control block was overlaid. In addition to formatting the control block, the unformatted control block plus 480 bytes preceding and following the area are printed in hex and in EBCDIC. No attempt is made to format those control blocks anchored in the control block.

PCT IS INACCESSIBLE IN THE DUMP DATA SET.
FORMATTING OF THE PCT QUEUE IS TERMINATED.

PCT QUEUE IS INACCESSIBLE IN THE DUMP DATA SET.
FORMATTING OF THE PCT QUEUE IS TERMINATED.

Explanation: Either PRDMP could not access the control block in the dump data set or the dump data set was incomplete and did not contain the control block. The control block and those control blocks anchored in it are not formatted.

PCT SECTOR VALUE TABLE

PCT SECTOR VALUE TABLE IS INACCESSIBLE IN THE DUMP DATA SET.
FORMATTING OF THE PCT SECTOR VALUE TABLE IS TERMINATED.

Explanation: Either PRDMP could not access the control block in the dump data set or the dump data set was incomplete and did not contain the control block. The control block and those control blocks anchored in it are not formatted.

PLPA PAGE DATASET WAS MARKED BAD

Explanation The page dataset was marked bad.

POSSIBLE LOOP IN AIA QUEUE - ATTEMPTED TO FORMAT MORE AIAS THAN
IN THE AUXILIARY STORAGE MANAGER.

POSSIBLE LOOP IN AIA QUEUE. ATTEMPTED TO FORMAT MORE AIAS THAN IN
THE SYSTEM. THE AIANXAIA FIELD OF SOME AIA DOES NOT
CONTAIN A NULLPTR.

Explanation: While formatting the AIA queue, the number of AIAs formatted exceeded the maximum number of AIAs in the system at that time. Therefore, the noncircular AIA queue has a loop in it.

POSSIBLE LOOP IN PCT QUEUE.PCTNEXT FIELD OF THE LAST
PCT ON THE QUEUE DOES NOT CONTAIN A NULLPTR.
FORMATTING OF THE PCT QUEUE IS TERMINATED.

Explanation: The noncircular PCT queue does not contain a zero in the last PCT of that queue. Possible storage overlay in the last PCT.

POSSIBLE LOOP IN THE PCCW QUEUE.
PCCWPCCW FIELD OF THE LAST PCCW ON
THE PCCW QUEUE DOES NOT CONTAIN A NULLPTR.
FORMATTING OF THE PCCW QUEUE IS TERMINATED.

Explanation: The noncircular PCCW queue does not contain a NULLPTR in the last PCCW of that queue. Possible storage overlay in the last PCCW.

SART ADDRESS IS ZERO - POSSIBLE STORAGE OVERLAY IN THE ASMVT.
FORMATTING OF THE SART, SDCT, SARWAITQ OF AIAS AND THE
CONTROL BLOCKS ANCHORED IN EACH SARTE IS TERMINATED.

Explanation: Since the SART address in the ASMVT (ASMSART) is zero, ASMVT was probably overlaid.

SART AND 480 BYTES PRECEDING AND FOLLOWING THE SART IN HEX AND
EBCDIC

SART DOES NOT CONTAIN THE SARTID - POSSIBLE STORAGE OVERLAY.
FORMATTING OF THE SART, SDCT, SARWAITQ OF AIAS AND THE
CONTROL BLOCKS ANCHORED IN EACH SARTE IS TERMINATED.
SART DOES NOT CONTAIN THE SARTID - POSSIBLE STORAGE OVERLAY.
WILL FORMAT AND PRINT SART AND 480 BYTES PRECEDING AND
FOLLOWING SART IN HEX AND EBCDIC. NO ATTEMPT TO FORMAT
SDCT, SARTWAITQ/AIAS AND THE CONTROL BLOCKS ANCHORED
IN EACH SARTE.

Explanation: The control block ID is invalid. It is possible that all or portions of the control block was overlaid. In addition to formatting the control block, the unformatted control block plus 480 bytes preceding and following the area are printed in hex and in EBCDIC. No attempt is made to format those control blocks anchored in the control block.

SART HEADER IS INACCESSIBLE IN THE DUMP DATA SET.
FORMATTING OF THE SART, SDCT, SARWAITQ OF AIAS AND THE
CONTROL BLOCKS ANCHORED IN EACH SARTE IS TERMINATED.

SARTE IS INACCESSIBLE IN THE DUMP DATA SET.
FORMATTING OF THIS SARTE AND THE CONTROL BLOCKS ANCHORED IN
THIS SARTE IS TERMINATED.
WILL ATTEMPT TO FORMAT THE REMAINING SARTES.

Explanation: Either PRDMP was not able to access the control block in the dump data set or the dump data set was incomplete and did not contain the control block. The control block and those control blocks anchored in it are not formatted.

SAT DOES NOT CONTAIN THE SATID - POSSIBLE STORAGE OVERLAY.
FORMATTING OF THE SAT HEADER AND THE SATMAP IS TERMINATED.

SAT DOES NOT CONTAIN THE SATID - POSSIBLE STORAGE OVERLAY.
WILL FORMAT AND PRINT SAT HEADER AND 480 BYTES PRECEDING
AND FOLLOWING SAT HEADER IN HEX AND EBCDIC
NO ATTEMPT TO FORMAT SATMAP.

SAT HEADER AND 480 BYTES PRECEDING AND FOLLOWING THE SAT HEADER
IN HEXADECIMAL AND EBCDIC.

Explanation: The control block ID is invalid. It is possible that all or portions of the control block was overlaid. In addition to formatting the control block, the unformatted control block plus 480 bytes preceding and following the area are printed in hex and in EBCDIC. No attempt is made to format those control blocks anchored in the control block.

SAT HEADER IS INACCESSIBLE IN THE DUMP DATA SET.
FORMATTING OF THE SAT HEADER AND THE SATMAP IS TERMINATED.

Explanation: Either PRDMP could not access the control block in the dump data set or the dump data set was incomplete and did not contain the control block. The control block and those control blocks anchored in it are not formatted.

SATMAP

SATMAP IS INACCESSIBLE IN THE DUMP DATA SET.
FORMATTING OF THE SATMAP IS TERMINATED.

Explanation: Either PRDMP could not access the control block in the dump data set or the dump data set was incomplete and did not contain the control block. The control block and those control blocks anchored in it are not formatted.

SCCW AND 480 BYTES PRECEDING AND FOLLOWING THE SCCW IN HEX AND
EBCDIC

SCCW DOES NOT CONTAIN THE SCCWID - POSSIBLE STORAGE OVERLAY.
FORMATTING OF THE SCCW QUEUE AND THE AIAS ANCHORED IN EACH
SCCW IS TERMINATED.

SCCW DOES NOT CONTAIN THE SCCWID - POSSIBLE STORAGE OVERLAY.
WILL FORMAT SCCW AND 480 BYTES PRECEDING AND FOLLOWING
SCCW IN HEX AND EBCDIC. FORMATTING OF SCCW QUEUE IS
TERMINATED. NO ATTEMPT TO FORMAT AIAS ANCHORED IN SCCW.

Explanation: The control block ID is invalid. It is possible that all or portions of the control block was overlaid. In addition to formatting the control block, the unformatted control block plus 480 bytes preceding and following the area are printed in hex and in EBCDIC. No attempt is made to format those control blocks anchored in the control block.

SCCW IS INACCESSIBLE IN THE DUMP DATA SET.
FORMATTING OF THE SCCW QUEUE AND THE AIAS ANCHORED IN EACH
SCCW IS TERMINATED.

Explanation: Either PRDMP could not access the control block in the dump data set or the dump data set was incomplete and did not contain the control block. The control block and those control blocks anchored in it are not formatted.

SDCT AND 480 BYTES PRECEDING AND FOLLOWING THE CONTROL BLOCK IN
HEX AND EBCDIC. NO ATTEMPT TO FORMAT SDCT ENTRIES.

SDCT DOES NOT CONTAIN THE SDCTID - POSSIBLE STORAGE OVERLAY.
FORMATTING OF THE SDCT HEADER AND THE SDCT ENTRIES IS
TERMINATED.

SDCT DOES NOT CONTAIN THE SDCTID - POSSIBLE STORAGE OVERLAY.
WILL FORMAT AND PRINT SDCT AND 480 BYTES PRECEDING AND
FOLLOWING SDCT IN HEX AND EBCDIC. NO ATTEMPT TO FORMAT SDCT
ENTRIES.

Explanation: The control block ID is invalid. It is possible that all or portions of the control block was overlaid. In addition to formatting the control block, the unformatted control block plus 480 bytes preceding and following the area are printed in hex and in EBCDIC. No attempt is made to format those control blocks anchored in the control block.

SDCT ENTRY IS INACCESSIBLE IN THE DUMP DATA SET.
FORMATTING OF ALL SDCT ENTRIES IS TERMINATED.

SDCT HEADER IS INACCESSIBLE IN THE DUMP DATA SET.
FORMATTING OF THE SDCT HEADER AND THE SDCT ENTRIES IS
TERMINATED.

Explanation: Either PRDMP could not access the control block in the dump data set or the dump data set was incomplete and did not contain the control block. The control block and those control blocks anchored in it are not formatted.

SRB AND 480 BYTES PRECEDING AND FOLLOWING THE SRB IN HEX AND
EBCDIC.

SRB DOES NOT CONTAIN THE SRBID - POSSIBLE STORAGE OVERLAY.
FORMATTING OF SRB AND AIA QUEUE ANCHORED IN THE SRB IS
TERMINATED.

SRB DOES NOT CONTAIN THE SRBID - POSSIBLE STORAGE OVERLAY.
WILL FORMAT AND PRINT SRB AND 480 BYTES PRECEDING AND
FOLLOWING SRB IN HEX AND EBCDIC.

Explanation: The control block ID is invalid. It is possible that all or portions of the control block was overlaid. In addition to formatting the control block, the unformatted control block plus 480 bytes preceding and following the area are printed in hex and in EBCDIC. No attempt is made to format those control blocks anchored in the control block.

SRB IS INACCESSIBLE IN THE DUMP DATASET.
FORMATTING OF SRB AND THE AIA QUEUE ANCHORED
IN THE SRB IS TERMINATED.

Explanation: Either PRDMP could not access the control block in the dump dataset or the dump dataset was incomplete and did not contain the control block. The control block and those control blocks anchored in it are not formatted.

SWAP DATASET WAS MARKED BAD

Explanation The swap dataset was marked bad.

THE ASMSWRQR AND ASMSWRQC COUNTERS INDICATE THERE ARE NO
SWAP AIAS WAITING FOR SWAP RESOURCES, HOWEVER,
THE SARWAITF PTR IS NOT ZERO - POSSIBLE STORAGE OVERLAY.
FORMATTING OF THE SWAP AIAS IS TERMINATED.

Explanation: The ASMSWRQR and ASMSWRQC counters are located in the ASMVT and ASMSWRQR - ASMSWRQC indicates the number of AIAs which have not completed processing. If the difference is nonzero and the SARWAITF pointer, that points to the AIAs, is nonzero then the SART or ASMVT was overlaid with invalid data.

THE FOLLOWING AIAS (ANCHORED IN SARWAITQ) ARE
WAITING FOR AVAILABLE SWAP RESOURCES

Explanation: These AIAS are waiting for SCCWS to become available so the swap ins/swap outs they represent can be performed.

THE FOLLOWING AIAS ARE ANCHORED IN THE ABOVE SCCW.

Explanation: This is a title statement indicating that the following AIAS, representing swap ins/swap outs, are associated with the above SCCW, containing the channel programs needed to perform the swap ins/swap outs.

THE FOLLOWING AIAS ARE ERROR AIAS SENT TO ILRCMSRB TO BE PROCESSED.

Explanation: These AIAs are anchored in an SRB that gives control to ILRCMSRB, which in turn calls ILRPAGCM to resume normal processing.

THE FOLLOWING AIAS ARE FOR LSQA SWAPOUT PAGES.

Explanation: These AIAs are anchored in an SRB scheduled by ILRSLSQA which gives control to ILRSWLIO which begins the swap out.

THE FOLLOWING AIA IS ANCHORED IN THE ABOVE PCCW.

Explanation: This is a title statement indicating that the following AIA represents a page in/page out, and that the PCCW, which anchors the AIA, represents the channel program performing the page in/page out.

THE FOLLOWING AVAILABLE PCCWS ARE ANCHORED IN ASMPCCWQ:

Explanation This message precedes the formatting of the available PCCW queue.

THE FOLLOWING AVAILABLE SCCWS ARE ANCHORED IN SARSCCWQ:

Explanation This message precedes the formatting of the available SCCW queue.

THE FOLLOWING ERROR AIAS ARE BEING RETURNED TO RSM.

Explanation: ASM encountered unrecoverable errors in these AIAs and ASM is returning them to RSM.

THE FOLLOWING NO PCCW AIAS ARE ANCHORED IN PARTNPCW.

Explanation: This is a title statement indicating that the AIAs to be formatted are waiting for PCCWs to become available so the page ins/page outs these AIAs represent can be started.

THE FOLLOWING PCCW QUEUE IS ANCHORED IN THE ABOVE IORB.

Explanation: The following PCCWS represent page ins/page outs that have started.

THE FOLLOWING REDRIVE AIAS ARE SENT TO ILREDRV TO BE PROCESSED.

Explanation: These AIAs are anchored in an SRB scheduled by ILRPAGCM which gives control to ILREDRV so the AIAs can be redriven.

THE FOLLOWING SCCWS (ANCHORED IN IORSCCW) ARE BEING PROCESSED

Explanation: These SCCWS, anchored in the IORB that is associated with the above SARTE, have already been started. That is, the swapping is currently in process.

THE FOLLOWING SCCWS (ANCHORED IN SRESCCW) ARE WAITING TO BE STARTED.

Explanation: These SCCWS are anchored in the above SARTE and are the channel programs to perform swap ins/swap outs directed towards the swap data set represented by the SARTE. These SCCWS are waiting for other swapping activity directed towards the swap data set to complete so they can be started.

THE FOLLOWING SRB (ANCHORED IN ASMPGSRB) IS USED TO SCHEDULE ILRCMSRB.

Explanation: Because an error occurred, an SRB is scheduled. ILRCMSRB gets control and calls ILRPAGCM, which resumes normal processing.

THE FOLLOWING SRB (ANCHORED IN ASMPSRB) IS USED BY ILRPAGCM TO SCHEDULE ILREDRV.

Explanation: ILRPAGCM is scheduling an SRB so ILREDRV will get control. The SRB is scheduled for either of the following reasons:

- When PCCWs are lacking
- If AIA previously received I/O errors

THE FOLLOWING SRB (ANCHORED IN ASMRSRB) IS USED TO PASS ERROR AIAS TO RSM.

Explanation: In the processing of these AIAs, ASM encountered errors in the AIAs that ASM could not correct. The AIAs are passed back to RSM by scheduling an SRB with the AIAs anchored in the SRB.

THE FOLLOWING SRB (ANCHORED IN ASMSWSRB) IS USED BY ILRSLSQA TO SCHEDULE ILRSWLIO.

Explanation: Before a swap out occurs, the working set of pages is trimmed (some pages are sent to the local page data sets which reduces the number of pages swapped out). When the pages are trimmed, ILRSLSQA schedules an SRB so ILRSWLIO can begin the swap out.

THE FOLLOWING SRB IS ANCHORED IN THE IOSB AT LOCATION

Explanation: This SRB is built by ASM and used by the I/O subsystem (IOS).

THE FOLLOWING SRB IS ANCHORED IN THE IORB AT LOCATION

Explanation: This SRB is used to schedule ILRCPBLD to issue a start I/O.

THE IOSUSE FIELD OF THE IOSB DOES NOT POINT BACK TO THE IORB IT IS
ANCHORED IN. POSSIBLE STORAGE OVERLAY. FORMATTING OF IOSB AND
SRB ANCHORED IN IOSB IS TERMINATED.

THE IOSUSE FIELD OF THE IOSB DOES NOT POINT BACK TO THE IORB IT IS
ANCHORED IN. POSSIBLE STORAGE OVERLAY. WILL FORMAT AND PRINT
480 BYTES PRECEDING AND FOLLOWING IOSB IN HEX AND EBCDIC. NO
ATTEMPT TO FORMAT SRB ANCHORED IN IOSB.

Explanation: The IOSB does not contain a control block ID. Therefore, to verify that the IOSB is valid, the IOSUSE field is checked to see if it points back to the IORB that the IOSB is anchored in. In this instance, it does not. It is possible that all or portions of the control block was overlaid with other data. In addition to formatting the control block, the unformatted control block plus 480 bytes preceding and following the area are printed in hex and in EBCDIC. No attempt is made to format those control blocks anchored in the control block.

THE LENGTH OF THE PATMAP WAS CALCULATED TO BE ZERO -
POSSIBLE STORAGE OVERLAY IN THE PATMAP.
FORMATTING OF THE PATMAP IS TERMINATED.

THE LENGTH OF THE PCT SECTOR VALUE TABLE IS ZERO -
POSSIBLE STORAGE OVERLAY IN THE PCT.
FORMATTING OF THE PCT SECTOR VALUE TABLE IS TERMINATED.

THE LENGTH OF THE SATMAP WAS CALCULATED TO BE ZERO -
POSSIBLE STORAGE OVERLAY IN THE SAT HEADER.
FORMATTING OF THE SATMAP IS TERMINATED.

Explanation: Informational.

Unnumbered Messages Issued By Virtual Storage Manager (VSM)

ACCESS REQUEST FAILED FOR ADDRESS SPACE EXTENSION
BLOCK (ASXB)

Explanation: The requestor received a nonzero return code from the access service routine while accessing the indicated control block.

ACCESS REQUEST FAILED FOR DESCRIPTOR QUEUE ELEMENT
DESCRIPTOR QUEUE ELEMENT QUEUE (VIRTUAL BELOW, REAL BELOW)
DATA FOLLOWS

ACCESS REQUEST FAILED FOR ADDRESS QUEUE ANCHOR TABLE
ENTRY

ACCESS REQUEST FAILED FOR CELL POOL EXTENT

ACCESS REQUEST FAILED FOR DOUBLE FREE ELEMENT

ACCESS REQUEST FAILED FOR DEFERRED RELEASE FBQE

ACCESS REQUEST FAILED FOR FREE BLOCK QUEUE ELEMENT

ACCESS REQUEST FAILED FOR FREE QUEUE ELEMENT

ACCESS REQUEST FAILED FOR LOCAL DATA AREA

ACCESS REQUEST FAILED FOR LOCAL VSM WORK AREA

ACCESS REQUEST FAILED FOR LSQA ALLOCATED ELEMENT

ACCESS REQUEST FAILED FOR SUBPOOL QUEUE ANCHOR

ACCESS REQUEST FAILED FOR SUBPOOL QUEUE ELEMENT

ACCESS REQUEST FAILED FOR TASK CONTROL BLOCK

ACCESS REQUEST FAILED FOR SIZE QUEUE ANCHOR TABLE

Explanation: The requestor received a nonzero return code from the access service routine while accessing the indicated control block.

ADDRESS QUEUE IS INVALID - POSSIBLE STORAGE OVERLAY

Explanation: The DFE address queue is not in ascending address order.

ADDRESS SPACE REGION DESCRIPTOR DATA FOLLOWS

Explanation: Informational

ADDRESS SPACE VECTOR TABLE ERROR DETECTED

Explanation: An error was detected in the indicated control block. The messages that were previously issued provide more specific information.

ALL ASIDS WERE REQUESTED

Explanation: Message describing the selection criteria for the ASID.

ALLOCATED ELEMENT AT ADDRESS XXXXXXXX

AQAT ENTRY FOR THE 64K AREA STARTING AT ADDRESS XXXXXXXX -
DFE EXISTS FOR THIS ENTRY ALLOCATION BITS: XXXX

AQAT ENTRY FOR THE 64K AREA STARTING AT ADDRESS XXXXXXXX -
NO DFE RESIDES AT XXXXXXXX ALLOCATION BITS: XXXX

AQAT IDENTIFIER IS INVALID - POSSIBLE STORAGE OVERLAY

Explanation: The indicated control block identifier is invalid.

ASCB AT ADDRESS XXXXXXXX IS NOT ACCESSIBLE

Explanation: The requestor received a nonzero return code from the access service routine while accessing the indicated control block.

ASCB IDENTIFIER INVALID - POSSIBLE STORAGE OVERLAY

Explanation: The indicated control block identifier is invalid.

ASVT ENTRY NOT ACCESSIBLE

Explanation: The requestor received a nonzero return code from the access service routine while accessing the indicated control block.

ASVT IDENTIFIER INVALID - POSSIBLE STORAGE OVERLAY

ASXB IDENTIFIER IS INVALID - POSSIBLE STORAGE OVERLAY

Explanation: The indicated control block identifier is invalid.

AUTHORIZED USER KEY (SUBPOOLS 229 AND 230) DATA FOLLOWS

BELOW 16 MEGABYTE DATA FOLLOWS

Explanation: Informational

CELL POOL EXTENT IS INVALID - POSSIBLE STORAGE OVERLAY

Explanation: The pointer to the cell pool extent (LDACPADR) is zero, which it should not be.

CELL POOL IDENTIFIER IS INVALID - POSSIBLE STORAGE OVERLAY

Explanation: The indicated control block identifier is invalid.

COMMON SERVICE AREA DATA FOLLOWS

CSA (BELOW) REGION DESCRIPTOR DATA FOLLOWS

CSA (EXTENDED) REGION DESCRIPTOR DATA FOLLOWS

CSA DEFERRED RELEASE CONDITION DETECTED

CSA SUBPOOL TABLE ERROR DETECTED

Explanation: An error was detected in the indicated control block. The messages that were previously issued provide more specific information.

CSA SUBPOOL TABLE IS NOT ACCESSIBLE

Explanation: The requestor received a nonzero return code from the access service routine while accessing the indicated control block.

DATA FOLLOWS FOR CSA SUBPOOL XXX, KEY XX

DATA FOLLOWS FOR THE ADDRESS QUEUE ANCHOR TABLE

DEFERRED RELEASE CONDITION DETECTED

DEFERRED RELEASE FBQE

DEFERRED RELEASE FBQE AT ADDRESS XXXXXXXX

DESCRIPTOR QUEUE ELEMENT QUEUE (VIRTUAL ANYWHERE)
DATA FOLLOWS

DESCRIPTOR QUEUE ELEMENT QUEUE (VIRTUAL BELOW, REAL ANYWHERE)
DATA FOLLOWS

DFE AT ADDRESS XXXXXXXX

DFE AT ADDRESS XXXXXXXX - THIS IS A DUMMY DFE

DOUBLE FREE ELEMENTS IN ADDRESS QUEUE ORDER FOLLOW

DOUBLE FREE ELEMENTS IN SIZE QUEUE ORDER FOLLOW

Explanation: Informational

DQE AREA ADDRESS IS INVALID - POSSIBLE STORAGE OVERLAY

Explanation: The field named DQEAREA is not on a page boundary.

DQE AT ADDRESS XXXXXXXX

DQE SIZE IS INVALID - POSSIBLE STORAGE OVERLAY

Explanation: The field named DQESIZE is not a page multiple.

DUMMY DFE AT ADDRESS XXXXXXXX

END OF VIRTUAL STORAGE MANAGEMENT DUMP FORMAT ROUTINE
EXTENDED AREA DATA FOLLOWS

EXTENDED ADDRESS SPACE REGION DESCRIPTOR DATA FOLLOWS

Explanation: Informational

EXTENDED SYSTEM REGION DESCRIPTOR IS INVALID - POSSIBLE STORAGE OVERLAY

EXTENDED V=R REGION DESCRIPTOR IS INVALID - POSSIBLE STORAGE OVERLAY

Explanation: Certain fields in the indicated control block are nonzero; they should be zero.

FBQE AREA ADDRESS IS INVALID - POSSIBLE STORAGE OVERLAY

Explanation: The field named FQEAREA is not on a page boundary.

FBQE SIZE IS INVALID - POSSIBLE STORAGE OVERLAY

Explanation: The field named FQESIZE is not on a page multiple.

FIXED GLOBAL VSWK IS NOT ACCESSIBLE

Explanation: The requestor received a nonzero return code from the access service routine while accessing the control block.

FORMAT REQUEST FAILED FOR CELL POOL EXTENT

FORMAT REQUEST FAILED FOR DEFERRED RELEASE FBQE

FORMAT REQUEST FAILED FOR DESCRIPTOR QUEUE ELEMENT

FORMAT REQUEST FAILED FOR DOUBLE FREE ELEMENT

FORMAT REQUEST FAILED FOR FREE BLOCK QUEUE ELEMENT

FORMAT REQUEST FAILED FOR FREE QUEUE ELEMENT

FORMAT REQUEST FAILED FOR GLOBAL DATA AREA

FORMAT REQUEST FAILED FOR LOCAL VSM WORK AREA

FORMAT REQUEST FAILED FOR LOCAL DATA AREA <===

FORMAT REQUEST FAILED FOR LSQA ALLOCATED ELEMENT

FORMAT REQUEST FAILED FOR REGION DESCRIPTOR

FORMAT REQUEST FAILED FOR REGION REQUEST ELEMENT

FORMAT REQUEST FAILED FOR SUBPOOL QUEUE ANCHOR

FORMAT REQUEST FAILED FOR SUBPOOL QUEUE ELEMENT

FORMAT REQUEST FAILED FOR VSM CELL POOL HEADER

FORMAT REQUEST FAILED FOR VSM WORK AREA

Explanation: The requestor received a nonzero return code from the format service routine while formatting the indicated control block.

FQE AT ADDRESS XXXXXXXX

FQE LINK POINTER IS INVALID - POSSIBLE STORAGE OVERLAY

Explanation: The field named FQENEXT contains zero, and it should not.

FQE SIZE IS INVALID - POSSIBLE STORAGE OVERLAY

Explanation: The value in the field named FQESIZE is larger than 8K; the largest valid size is 8K-16.

FREE BLOCK QUEUE ELEMENT (FBQE) AT ADDRESS XXXXXXXX

FREE QUEUE ELEMENT DQE ADDRESS IS INVALID - POSSIBLE STORAGE OVERLAY

Explanation: The field named FQEDQE does not point to the owning DQE.

GLOBAL DATA AREA IDENTIFIER INVALID

Explanation: The indicated control block identifier is invalid.

GLOBAL DATA AREA INACCESSIBLE - GLOBAL VSM DATA CANNOT BE DISPLAYED

Explanation: The requestor received a nonzero return code from the access service routine while accessing the indicated control block.

GLOBAL FIXED VSWK ERROR DETECTED

GLOBAL PAGEABLE VSWK ERROR DETECTED

GLOBAL VSM CELL POOL ERROR DETECTED

Explanation: An error was detected in the indicated control block. The messages that were previously issued provide more specific information.

LDA IDENTIFIER IS INVALID - POSSIBLE STORAGE OVERLAY

Explanation: The indicated control block identifier is invalid.

LSQA ALLOCATED ELEMENTS FOLLOW FOR THE BELOW 16-MEG AREA

LSQA ALLOCATED ELEMENTS FOLLOW FOR THE EXTENDED AREA

LSQA (BELOW) DATA FOLLOWS

LSQA (EXTENDED) DATA FOLLOWS

NO ASCBS ARE ACCESSIBLE - LOCAL VSM DATA CANNOT BE DISPLAYED

PAGEABLE GLOBAL VSWK IS NOT ACCESSIBLE

REGION REQUEST ELEMENT IS NOT ACCESSIBLE

Explanation: The requestor received a nonzero return code from the access service routine while accessing the indicated control block.

REGION REQUEST ERROR DETECTED

Explanation: An error was detected in the indicated control block. The messages that were previously issued provide more specific information.

RGR AT ADDRESS XXXXXXXX

SCHEDULER WORK AREA (SUBPOOLS 236 AND 237) DATA FOLLOWS

SIZE QUEUE ANCHOR TABLE AT ADDRESS XXXXXXXX

SIZE QUEUE ANCHOR TABLE ENTRY FOR SIZE XXXXXXXX ADDRESSES THE

SIZE QUEUE IS INVALID - POSSIBLE STORAGE OVERLAY

Explanation: The DFE size queue is not in ascending size order.

SPQA AT ADDRESS XXXXXXXX

SPQE AT ADDRESS XXXXXXXX

SPQE TCB POINTER IS INVALID - POSSIBLE STORAGE OVERLAY

Explanation: The bit named SPQEOWN indicates that the TCB owns the SPQE, but SPQETCB does not point to this TCB.

SPTT IDENTIFIER IS INVALID - POSSIBLE STORAGE OVERLAY

SUBPOOL TABLE IDENTIFIER IS INVALID - POSSIBLE STORAGE OVERLAY

Explanation: The indicated control block identifier is invalid.

SUBPOOL TRANSLATION TABLE RESIDES AT ADDRESS XXXXXXXX

SYSTEM QUEUE AREA SUBPOOL 226 DATA FOLLOWS

SYSTEM QUEUE AREA SUBPOOL 239 DATA FOLLOWS

SYSTEM QUEUE AREA SUBPOOL 245 DATA FOLLOWS

SYSTEM REGION DESCRIPTOR DATA FOLLOWS

TASK-RELATED SUBPOOL INFORMATION FOLLOWS

TCB IDENTIFIER IS INVALID - POSSIBLE STORAGE OVERLAY

Explanation: The indicated control block identifier is invalid.

THE AMOUNT OF VIRTUAL STORAGE ALLOCATED TO SUBPOOL XXX IS XXXXXXXX

THE AMOUNT OF VIRTUAL STORAGE ALLOCATED TO SUBPOOL XXX, KEY XX
IS XXXXXXXX

THE FOLLOWING KEYWORDS ARE IN EFFECT:

ALL	JOBLIST
ASIDLIST	GLOBAL
CURRENT	NOASIDS
ERROR	NOGLOBAL
GLOBAL	

THE TOTAL NUMBER OF CELLS AVAILABLE FOR THIS ADDRESS SPACE
IS XXXXXXXX

THE TOTAL NUMBER OF GLOBAL VSM CELLS IS XXXXXXXX

THE VSM WORK AREA COMMUNICATION SECTION (VSWKCOMA) FOLLOWS

THIS SUBPOOL IS NOT OWNED - THE SPQA WAS FORMATTED ABOVE

Explanation: Informational.

THIS ASID IS A CURRENT ASID

THIS ASID IS A TCBERROR ASID

THIS ASID IS AN ERROR ASID

THIS ASID IS IN THE ASID LIST

THIS ASID IS IN THE JOB LIST

THIS ASID WAS SELECTED BECAUSE:

Explanation: These messages describe the selection criteria for the ASID.

USER REGION (SUBPOOLS 0 - 127, 251, AND 252) DATA FOLLOWS

VIRTUAL STORAGE MANAGEMENT DUMP FORMAT ROUTINE

V=R REGION DESCRIPTOR DATA FOLLOWS

VSM CELL POOL AT ADDRESS XXXXXXXX

VSM CELL POOL EXTENT AT ADDRESS XXXXXXXX

VSM CELL POOL EXTENT DATA FOLLOWS

VSM CELL POOL EXTENT IS INVALID - POSSIBLE STORAGE OVERLAY

Explanation: The pointer to the cell pool extent (LDACPADR) is zero, which it should not be.

VSM DATA FOLLOWS FOR ADDRESS SPACE NUMBER XXXX.
JOBNAME IS XXXXXXXX

VSM DATA FOLLOWS FOR TCB AT ADDRESS XXXXXXXX

VSM GLOBAL DATA AREA AT ADDRESS XXXXXXXX

VSM GLOBAL DATA AREA ERROR DETECTED

Explanation: An error was detected in the indicated control block. The messages that were previously issued provide more specific information.

VSM LOCAL DATA AREA AT ADDRESS XXXXXXXX

Explanation: Informational

VSM SUBPOOL TRANSLATION TABLE ERROR DETECTED

Explanation: An error was detected in the indicated control block. The messages that were previously issued provide more specific information.

VSM SUBPOOL TRANSLATION TABLE IS NOT ACCESSIBLE

Explanation: The requestor received a nonzero return code from the access service routine while accessing the indicated control block.

VSM WORK AREA AT ADDRESS XXXXXXXX

VSM WORK AREA FOR GLOBAL FIXED STORAGE REQUESTS FOLLOWS

VSM WORK AREA FOR GLOBAL PAGEABLE STORAGE REQUESTS FOLLOWS

VSM WORK AREA IDENTIFIER IS INVALID - POSSIBLE STORAGE OVERLAY

VSWK IDENTIFIER IS INVALID - POSSIBLE STORAGE OVERLAY

Explanation: The indicated control block identifier is invalid.

Problem Determination

Problem determination is the activity required to identify a failing hardware unit or program and determine who is responsible for support.

Problem determination is accomplished by using procedures specified by IBM. In some cases, these procedures may be initiated by a message or code which requires operator or programmer response. The response may include the requirement for additional problem-related data to be collected and will attempt, where possible, to indicate "probable" failure responsibility.

Problem determination information is included for applicable messages under the heading "Problem Determination." Standard problem determination actions are identified as items of Tables I and II. Unique actions are identified following the list of standard actions to be taken. In any case, it is intended that the specified actions be taken before calling IBM for support.

TABLE I

If the problem recurs, follow the problem determination aids specified by the associated message or code before calling IBM for support.

1. If MSGLEVEL=(1,1) was not specified in the JOB statement, specify it and rerun the job.

2. Save the console sheet from the primary console. For systems with remote consoles, save the remote console sheet. In systems with Multiple Console Support (MCS), save a copy of the hard copy log.

3. Save the job stream associated with the job.

4. Save the system output (SYSOUT) associated with the job.

5. Make sure that the failing job step includes one of the following:

   a. SYSABEND DD statement.
   b. SYSUDUMP DD statement.
   c. PL1DUMP DD statement.
   d. SYSMDUMP DD statement.

6. Make sure that the PARM parameter of the EXEC statement specifies the following:

   a. MAP
   b. LIST
   c. DIAG
   d. MSG=AP
   e. CORE, if applicable
   f. XREF
   g. DUMP

7. If SMP is used to make all changes to the system, execute the LIST CDS and LIST PTFBY functions of SMP to obtain a list of the current maintenance from the SMP control data set (CDS).

   If any changes are made to the system without using SMP, execute the LISTIDR function of the AMBLIST service aid program to obtain a list of all members with a PTF or local fix, and save the output. Execute the program against the:

   a. SYS1.LINKLIB data set.
   b. SYS1.SVCLIB data set.
   c. library containing the program that issued the message.
   d. SYS1.LPALIB data set.

8. Execute the IMCJOBQD (stand-alone) or IMCOSJQD (system-assisted) service aid program to obtain a formatted copy of the contents of the SYS1.SYSJOBQE or SYS1.SYSWADS data sets, SWADS or the resident job list. (Not applicable for VS2 MVS.)

9. Execute the AMBLIST service aid program to obtain:

   a. an object module listing, specifying the LISTOBJ function.
   b. a load module map and cross-reference listing, specifying the OUTPUT=BOTH option of the LISTLOAD function.

10. Have a copy of the Message Control Program (MCP) available.

11. Execute the AMDSADMP service aid program to dump the contents of real storage and page data sets on magnetic tape.

    After restarting the system, execute the appropriate function of the AMDPRDMP service aid program to print the required portion of the dump tape produced by AMDSADMP.

    Save both the tape from AMDSADMP (should further information from the tape be required) and the listing from AMDPRDMP.

12. Execute the SEREP program, and save the resulting output.

    (Note: The SEREP program is not supported on processors in the 4300 series.)

13. Save all the associated output.

14. The normal response to this message requests the programmer/operator to execute a specific program. Save all output from that program.

15. Save the program listing associated with the job.

16. Save the dump.

17. Have the system generation (SYSGEN) output available from:

    a. Stage I
    b. Stage II

18. Execute the EREP service aid, to dump the SYS1.LOGREC data set and save the resulting output.

    For MSS, execute the following program to dump the SYS1.LOGREC data set:

    a. Service aid IFCISDA0
    b. Program ISDASDA0 with the DETAIL(ALL) parameter.

19. Save the assembly listing associated with the job.

20. Save the control cards associated with the job.

21. Save the compiler output associated with the job.

22. Save the source input associated with the job.

23. Save the source program listing associated with the job.

24. Run OLTEP diagnostics for the problem device and save the output.

25. Execute the IEHLIST system utility program to obtain a list of the:

    a. volume table of contents of the associated volume, specifying the FORMAT option.
    b. volume table of contents of the associated volume, specifying the DUMP option.
    c. directory of the associated data set
    d. (Not applicable for MVS.)

26. Execute the IEBPTPCH data set utility to print the:

    a. directory of the applicable data set.
    b. applicable data set.
    c. applicable member.
    d. applicable procedure.

27. Have the linkage editor/loader map available.

28. Save the associated volume.

29. Contact your programming support personnel.

30. Contact your hardware support personnel.

31. Save the trace output data set.

32. Print the GTF trace data set with the AMDPRDMP service aid program using the EDIT statement.

33. Print the associated SVC Dump data set, using the AMDPRDMP service aid with the GO statement.

34. Execute the access method services LISTCAT command to:

    a. list the contents of the applicable catalog.
    b. list the catalog entries for the applicable objects and any related objects.

35. Execute the following access method services command:

    a. The MSS LISTMSF command for mountable volumes.
    b. The MSS LISTMSF command with the CARTRIDGES parameter.
    c. The PRINT command to list the contents of the mass storage volume control inventory data set.
    d. The LISTMSVI command.
    e. The LISTMSF command with the ALL parameter.

36. Execute the access method services PRINT command to print the repair work file.

37. Execute the AMASPZAP service aid program using the ABSDUMP statement to print the contents of the applicable:

    a. Data set.
    b. Track.

38. Execute the access method services AUDITMSS command with the following parameter:

    a. The CHECK parameter.
    b. The MAP parameter.
    c. The READLABEL parameter.

39. Execute the access method services CHECKMSS command.

40. Execute the access method services COMPARET command.

41. Execute the access method services DUMPMSS command to dump the following:

    a. Formatted mass storage control storage.
    b. Mass storage control main storage.
    c. Mass storage control extended storage.
    d. Formatted Staging Adapter storage.
    e. Staging Adapter main storage.
    f. Staging Adapter extended storage.
    g. Mass storage control tables.

42. Save the latest output from the Mass Storage Control Table Create program.

43. Display units for units associated with the problem area. If specific unit(s) is not known, display range of all virtual units. See your configuration path chart for address ranges.

44. Obtain the RACF profile of the associated data set, where applicable.

45. Stop the processor and use the hardware ALTER/DISPLAY facility to display:

    a. all general purpose registers.
    b. the PSW.
    c. main storage locations 0 through 200 (hexadecimal) and 7000 through 7080 (hexadecimal).

46. If the AMDSADMP program resides on tape, save the tape. If the AMDSADMP program resides on disk, use the DUMP feature of IEHDASDR to print the SYS1.PAGEDUMP data set and cylinder 0 track 0 of this residence disk.

47. Save the output (listings) of the stage 1 and stage 2 AMDSADMP initialization jobs.

48. Follow the procedures for item 9b of this table for load modules AMDSAPGE, AMDSAPRO and AMDSALDR of SYS1.LINKLIB. Use IEBUPDTE or IEBPTCH to print the AMDSADMP and AMDSADM2 macros from SYS1.MACLIB.

49. Save the AMDSADMP dump output (tape or listing).

50. If the program seems to be looping, use the display PSW feature of the hardware ALTER/DISPLAY facility along with the hardware instruction Step facility to trace the loop, instruction by instruction.

51. If there is an error in the contents of a page data set dump, restart the system using a different page data set, then dump the original page data set.

52. Use IEBCOPY to unload SYS1.IMAGELIB to tape.

53. Have a list of RACF-defined entities available.

54. Contract your IBM system engineer.

55. Save the console sheets from all active global resource serialization systems, and from any systems that are restarting or joining the global resource serialization complex.

TABLE II

GTF for Problem Determination

Format 1: Tracing Without Prompting for Event Keywords

Before reproducing the problem, have the system operator issue a START GTF command specifying tape output, MODE = EXT and TIME = YES. In response to message AHL100A the operator should type TRACE = opt, where opt is the trace option indicated for the particular message or code, within the text of his reply.

When data for the problem has been recorded, run the AMDPRDMP service aid program using the EDIT statement to format the trace output, specifying DDNAME = (ddname of the trace data set).

Format 2: Tracing With Prompting for Event Keywords

Before reproducing the problem, have the system operator issue a START GTF command specifying tape output, MODE = EXT and TIME = YES. In response to the message AHL100A the operator should specify the trace options indicated for the associated message or code within the text of his reply. Then, in response to the message AHL101A, he should specify the event keywords also indicated with the associated message or code.

When data for the problem has been recorded, run the AMDPRDMP service aid program using the EDIT statement to format the trace output, specifying DDNAME = (ddname of the trace data set).

Format 3: Specialized Tracing Action

Before reproducing the problem, have the system operator issue a START GTF command specifying tape output, MODE = EXT and TIME = YES. In response to message AHL100A the operator should type 'TRACE = SYS,USR,SLIP'. The DD statement for a data set in error should specify DCB = DIAGNS = TRACE.

When data for the problem has been recorded, execute the EDIT function of AMDPRDMP specifying the options SYS, USR = FFF, and SLIP.

Format 4: Specialized Tracing Action for VSAM

Before reproducing the problem, have the system operator issue a START GTF command specifying tape output, MODE = EXT and TIME = YES. In response to message AHL100A the operator should type 'TRACE = SYS,USR'. The DD statement for a data set in error should specify AMP = TRACE.

When data for the problem has been recorded, execute the EDIT function of AMDPRDMP specifying the options SYS and USR = (FFF,FF5).

TABLE III

If a problem occurs in JES3, one or more of the following steps may be taken to assist in determining the cause:

1. Take a stand-alone dump of the system by specifying DUMP=PRDMP on the OPTIONS card in the initialization deck and save the output (SYS1.DUMPnn).

2. Take a standard dump of the system by specifying DUMP=JES on the OPTIONS card in the initialization deck and save the output (JESABEND).

3. Take an operating system dump including the nucleus and SQA by specifying DUMP=MVS on the OPTIONS card in the initialization deck and save output (SYSABEND).

4. Save the MLOG listing or get a print of DLOG.

5. Provide listing of initialization deck (JES3OUT).

6. Provide console log from initialization.

7. Ascertain OS/VS level and JES3 PTF level.

8. Provide OS/VS nucleus LOADMOD map.

9. Issue *F T,L=linename, SNAPON and *X RJPSNPS.

10. Issue *F T,L-linename,TRCEON. This will give an RJP event trace on the MLOG console. Save MLOG output.

11. Take a system dump by placing an INTDEBUG,n, message-text$$ card in the initialization deck. The message-text field is compared for occurrences of the chosen message. The n field specifies the number of message occurrences before the system is dumped.

12. Issue *X DISPLAY and save output.

13. Issue *X DISPLAY,SNAPS and save the output.

14. Rerun job with //*PROCESS CBPRNT and save output.

    a. After Interpreter DSP
    b. After Main Service
    c. After Input Service

15. Rerun job with EXEC PGM=JCLTEST and save output.

16. Rerun job with EXEC PGM=JSTTEST and save output.

17. Rerun job with TYPRUN=SCAN specified on JOB card and save output.

18. Issue *X DISPDJC when problem occurs and save output.

19. Restart system with specifying a start type of WA (Warm start with queue analysis) and save output (JES3SNAP).

20. Check SYSMSG data set for error indications.

21. Provide a listing of the JES3 start up procedure, containing all JCL used to start the subsystem.

22. Save the IOERR trace that will be printed.

23. Rerun job with DEBUG=All immediately following PROCESS CI or PROCESS RI card.

TABLE IV

This section contains details for dumping interactive problem control system (IPCS) problem directory records and data set directory records.

Problem Directory Records

The problem directory contains information about each problem and the data sets associated with each problem. It is organized as a VSAM key-sequenced cluster containing four types of records: status records, data set association records, description records, and a seed value record. (For an explanation of a seed value record, see the *IPCS User's Guide and Reference*). The length of the key for the problem directory records is 14 bytes and the key is organized as follows:

Key Contents	Length in Bytes
xx	2
000	3
prob-id-suffix	5
rec-seq-num	4

where:

xx
defines the type of record as follows:

- **ST** indicates a problem status record. There should be one status record for each problem.
- **DS** indicates a data set association record. There can be zero to 10000 data set association records for each problem.
- **DE** indicates a description record. There can be zero to 10000 description records for each problem.
- **SV** indicates the seed value record. There is one seed value record for each cluster.

000
represents three EBCDIC zeros.

prob-id-suffix
represents the last five digits of the problem identifier. The prob-id-suffix is recorded in EBCDIC with the leading zeros specified. For the seed value record, this suffix contains zeros.

rec-seq-num
represents the record sequence number for the record. The number is recorded in EBCDIC with the leading zeros specified. For both the status record and the seed value record, the record sequence number is always 0000; for the data set association records and the description records, the number ranges from 0000 to 9999.

If the problem determination section for a message indicates the need to dump the appropriate problem directory records, all four record types should be requested. That is, the seed value record, the status record for the problem identifier in question and for the next problem identifier in sequence, and all the data set association records and description records for the problem identifier in question.

For example, if message BLS04000I is issued for the problem identifier XYZ00067, the following four IDCAMS PRINT statements satisfy the problem determination requirements.

```
PRINT INFILE(dname) OUTFILE(dname)
      FROMKEY(SV000000000000)
      TOKEY(SV000000000000) DUMP
PRINT INFILE(dname) OUTFILE(dname)
      DUMP FROMKEY(ST000000670000)
      TOKEY(ST000000680000)
PRINT INFILE(dname) OUTFILE(dname)
      FROMKEY(DS000000670000)
      TOKEY(DS000000679999) DUMP
PRINT INFILE(dname) OUTFILE(dname)
      FROMKEY(DE000000670000)
      TOKEY(DE000000679999) DUMP
```

For further information on using IDCAMS PRINT statements, see *access method services Reference*.

Data Set Directory Records

The data set directory contains information about each data set known to IPCS. It is organized as a VSAM key-sequenced cluster containing two types of records: base records and problem association records. The length of the key for the data set directory records is 61 bytes and the organization of the key depends on the type of record.

For base records:

Key Contents	Length in Bytes
B	1
dsname	44
membername	8
DSNbBASE	8

For problem association records:

Key Contents	Length in Bytes
B	1
dsname	44
membername	8
000	3
prob-id-suffix	5

where:

B
indicates a base record.

dsname
is the name of the data set, in EBCDIC, left justified with trailing blanks.

membername
is the name of the member, in EBCDIC, left justified with trailing blanks. If there is no membername, this field contains 8 blanks.

DSNbBASE
is the identifier for the data set base record (the 'b' represents a blank).

000
represents three EBCDIC zeros.

prob-id-suffix
represents the last five digits of the problem identifier with which this data set is associated. The prob-id-suffix is recorded in EBCDIC, with the leading zeros specified.

If the problem determination section for a message indicates the need to dump the appropriate data set directory records, both record types should be dumped for the particular data set name in question. That is, the data set base record and all problem association records for that data set.

For example, if message BLS04005I is issued for data set name ABC.DUMP.DEC7 (no membername), the following two IDCAMS PRINT statements satisfy the problem determination requirements (the lowercase b is used to represent a blank):

```
PRINT INFILE(dname) OUTFILE(dname) DUMP
  FROMKEY('BABC.DUMP.DEC7bbbbbbbbbbbbbb
  bbbbbbbbbbbbbbbbbbbbbbbbbbDSNbBASE')
  TOKEY('BABC.DUMP.DEC7bbbbbbbbbbbbbbbb
  bbbbbbbbbbbbbbbbbbbbbbbbDSNbBASE')

PRINT INFILE(dname) OUTFILE(dname) DUMP
  FROMKEY('PABC.DUMP.DEC7bbbbbbbbbbbbbb
  bbbbbbbbbbbbbbbbbbbbbbbb00000000')
  TOKEY('PABC.DUMP.DEC7bbbbbbbbbbbbbbbb
  bbbbbbbbbbbbbbbbbbbbbbbb00099999')
```

For further information on using IDCAMS PRINT statements, see *Access Method Services Reference*.

Message to Module Table

This table correlates message IDs with module numbers. For each message, three module numbers are listed: the module that detects the need for the message, the module that issues the message, and the module that contains the message.

If the message ID you require is not listed in this table, see the diagnostic aids documentation for the component that issues the message.

Message ID	Detecting Module	Issuing Module	Containing Module
AMD150I	AMDPRCTL	AMDPRCTL	AMDPRCTL
AMD151I	AMDPRRDC	AMDPRRDC	AMDPRRDC
	AMDPRRDC	AMDPRCOM	AMDPRRDC
	AMDPRFRM	AMDPRCOM	AMDPRFRM
	AMDPRCTL	AMDPRCOM	AMDPRPMG
AMD153I	AMDPRCTL	AMDPRCTL	AMDPRCTL
AMD162I	AMDPRPJB	AMDPRPJB	AMDPRPJB
	AMDPRPJB	AMDPRCOM	AMDPRPJB
AMD163I	AMDPRCTL	AMDPRCOM	AMDPRCTL
AMD164I	AMDPRCTL	AMDPRCOM	AMDPRCTL
AMD165I	AMDPRCTL	AMDPRCOM	AMDPRCOM
	AMDPRUIM	AMDPRCOM	AMDPRUIM
	AMDPRUIM	AMDPRUIM	AMDPRUIM
	AMDPRPMS	AMDPRPMS	AMDPRPMS
	AMDPRPJB	AMDPRCOM	AMDPRPJB
	AMDPRIUM	AMDPRCOM	AMDPRUIM
	AMDPRPMS	AMDPRCOM	AMDPRPMS
	AMDPRCTL	AMDPTCOM	AMDPRPMG
AMD166I	AMDPRPJB	AMDPRPJB	AMDPRPJB
	AMDPRPJB	AMDPRCOM	AMDPRPJB
AMD168I	AMDPRCTL	AMDPRCOM	AMDPRCTL
AMD170I	AMDPRCTL	AMDPRCOM	AMDPRCTL
AMD171I	AMDPRCOM	AMDPRCOM	AMDPRCOM
AMD172I	AMDPRCOM	AMDPRCOM	AMDPRCOM
AMD173I	AMDPRRDC	AMDPRCOM	AMDPRRDC
AMD174I	AMDPRLOD	AMDPRCOM	AMDPRLOD
AMD175I	AMDPRRDC	AMDPRCOM	AMDPRRDC
AMD177I	AMDPRSEG	AMDPRCOM	AMDPRPMG
AMD178	AMDPRSEG	AMDPRCOM	AMDPRPMG
AMD180I	AMDPRSEG	AMDPRCOM	AMDPRPMG
AMD181I	AMDPRSEG	AMDPRCOM	AMDPRPMG
AMD187I	AMDPRCTL	AMDPRCOM	AMDPRCTL
AMD201I	AMDPRSCN	AMDPRCOM	AMDPRSMG
AMD202I	AMDPRSCN	AMDPRCOM	AMDPRSMG
AMD203I	AMDPRSN2	AMDPRCOM	AMDPRSMG
AMD204I	AMDPRSN2	AMDPRCOM	AMDPRSMG
AMD205I	AMDPRSN2	AMDPRCOM	AMDPRSMG
AMD206I	AMDPRSN2	AMDPRCOM	AMDPRSMG
AMD207I	AMDPRSN2	AMDPRCOM	AMDPRSMG
AMD208I	AMDPRSCN	AMDPRCOM	AMDPRSMG
AMD209I	AMDPRSCN	AMDPRCOM	AMDPRSMG
AMD211I	AMDPRSN3	AMDPRCOM	AMDPRSMG
AMD212I	AMDPREXT	AMDPRCOM	AMDPRFMG
	AMDPRAPP	AMDPRCOM	AMDPRFMG
AMD213I	AMDPREXT	AMDPRCOM	AMDPRFMG
	AMDPRAPP	AMDPRCOM	AMDPRFMG
AMD214I	AMDPRAPP	AMDPRCOM	AMDPRFMG
AMD215I	AMDPRAPP	AMDPRCOM	AMDPRFMG
AMD216I	AMDPRAPP	AMDPRCOM	AMDPRFMG
AMD217I	AMDPRFRM	AMDPRCOM	AMDPRFMG
AMD220I	AMDPRGET	AMDPRCOM	AMDPRFMG
AMD225I	AMDPRGET	AMDPRCOM	AMDPRFMG
AMD226I	AMDPRFRM	AMDPRCOM	AMDPRFMG
AMD227I			AMDPRFMG
AMD228I	AMDPRGET	AMDPRCOM	AMDPRFMG
	AMDPRAPP	AMDPRCOM	AMDPRFMG
AMD251I	AMDPRPMS	AMDPRPMS	AMDPRPMS
AMD252I	AMDPRPMS	AMDPRPMS	AMDPRPMS
AMD254I	AMDPRRDC	AMDPRRDC	AMDPRRDC
	AMDPRRDC	AMDPRCOM	AMDPRPMG
AMD258I	AMDPRSLI	AMDPRSLI	AMDPRSLI
AMD260I	AMDPRMST	AMDPRCOM	AMDPRMST
	AMDPRMST	AMDPRMST	AMDPRMST
AMD261I	AMDPRMST	AMDPRMST	AMDPRMST
AMD263I	AMDPRMST	AMDPRMST	AMDPRMST
AMD264I	AMDPRMST	AMDPRMST	AMDPRMST
AMD267I	AMDPRFOR	AMDPRFOR	AMDPRFOR
AMD269I	AMDPRFOR	AMDPRFOR	AMDPRFOR
AMD270I	AMDPRFOR	AMDPRFOR	AMDPRFOR
AMD271I	AMDPRFOR	AMDPRFOR	AMDPRFOR
AMD272I	AMDPRFOR	AMDPRFOR	AMDPRFOR
AMD273I	AMDPRFOR	AMDPRFOR	AMDPRFOR
AMD274I	AMDPRFOR	AMDPRFOR	AMDPRFOR
AMD275I	AMDPRFOR	AMDPRFOR	AMDPRFOR
AMD276I	AMDPRFOR	AMDPRFOR	AMDPRFOR
AMD277I	AMDPRFOR	AMDPRFOR	AMDPRFOR
AMD278I	AMDPRMST	AMDPRMST	AMDPRMST
AMD280I	AMDPRLOD	AMDPRCOM	AMDPRLOD
AMD281I	AMDPRUIM	AMDPRCOM	AMDPRUIM
	AMDPRUIM	AMDPRUIM	AMDPRUIM
AMD282I	AMDPRPMS	AMDPRPMS	AMDPRPMS
AMD284I	AMDPRMST	AMDPRMST (WTO)	AMDPRMST
AMD285I	AMDPRRDC	AMDPRCOM	AMDPRRDC
AMD286I	AMDPRMST	AMDPRCOM	AMDPRMST
	AMDPRMST	AMDPRMST	AMDPRMST
AMD287I	AMDPRUIM	AMDPRCOM	AMDPRUIM
AMD288I	AMDPRUIM	AMDPRUIM	AMDPRUIM
AMD289I	AMDPRUIM	AMDPRUIM	AMDPRUIM
AMD290I	AMDPRMST	AMDPRCOM	AMDPRMST
	AMDPRMST	AMDPRMST	AMDPRMST
AMD291I	AMDPRLOD	AMDPRCOM	AMDPRLOD
AMD292I	AMDPRRDC	AMDPRCOM	AMDPRRDC
AMD293I	AMDPRRDC	AMDPRCOM	AMDPRRDC
AMD294I	AMDPRMST	AMDPRCOM	AMDPRMST
	AMDPRMST	AMDPRMST	AMDPRMST
AMD295I	AMDPRPMS	AMDPRPMS	AMDPRPMS
AMD296I	AMDPRPMS	AMDPRPMS	AMDPRPMS
AMD297I	AMDPRUIM	AMDPRUIM	AMDPRUIM
AMD298I	AMDPRSLI	AMDPRSLI	AMDPRSLI

Message ID	Detecting Module	Issuing Module	Containing Module
AMD299I	AMDPRSLI	AMDPRSLI	AMDPRSLI
AMD300I	AMDPRUIM	AMDPRUIM	AMDPRUIM
AMD301I	AMDPRUIM	AMDPRUIM	AMDPRUIM
AMD302I	AMDPREAD	AMDPREAD	AMDPREAD
AMD303I	AMDPRABS	AMDPRABS	AMDPRABS
AMD304I	AMDPRABS	AMDPRABS	AMDPRABS
AMD305I	AMDPRCTL	AMDPRCTL	AMDPRCTL
BLS01000I	BLSACGTH	BLSACGTH	BLSACGTH
BLS01001I	BLSACQEC	BLSACQEC	BLSACQEC
BLS01002I	BLSACRPT	BLSACRPT	BLSACRPT
BLS01003I	BLSACRPT	BLSACRPT	BLSACRPT
BLS01004I	BLSACGTH	BLSACGTH	BLSACGTH
BLS01005I	BLSACRPT	BLSACRPT	BLSACRPT
BLS01006I	BLSACGTH	BLSACGTH	BLSACGTH
BLS01040I	BLSACBST	BLSACBST	BLSACBST
BLS01041I	BLSACBST	BLSACBST	BLSACBST
BLS01042I	BLSACBSS	BLSACBSS	BLSACBSS
BLS01043I	BLSACBST	BLSACBST	BLSACBST
BLS01060I	BLSADSYM	BLSADSYM	BLSADSYM
BLS01061I	BLSADSYM	BLSADSYM	BLSADSYM
BLS01062I	BLSADSYM	BLSADSYM	BLSADSYM
BLS01063I	BLSADSYM	BLSADSYM	BLSADSYM
BLS01064I	BLSADSYM	BLSADSYM	BLSADSYM
BLS01065I	BLSADSYM	BLSADSYM	BLSADSYM
BLS01066I	BLSADSYM	BLSADSYM	BLSADSYM
BLS01067I	BLSADSYM	BLSADSYM	BLSADSYM
BLS03100I	BLSCAINT	BLSUTRMV	BLSDMSGS
BLS03101I	BLSCAINT	BLSUTRMV	BLSDMSGS
BLS03102I	BLSCABLD	BLSUTRMV	BLSDMSGS
BLS03103I	BLSCABLD	BLSUTRMV	BLSDMSGS
BLS03104I	BLSCABLD	BLSUTRMV	BLSDMSGS
BLS03105I	BLSCAMRE	BLSUTRMV	BLSDMSGS
BLS03106I	BLSCABLD	BLSUTRMV	BLSDMSGS
	BLSCAMRE	BLSUTRMV	BLSDMSGS
BLS03107I	BLSCABLD	BLSUTRMV	BLSDMSGS
	BLSCAMRE	BLSUTRMV	BLSDMSGS
BLS03108I	BLSCAAMS	BLSUTRMV	BLSDMSGS
BLS03109I	BLSCAAMS	BLSUTRMV	BLSDMSGS
	BLSCANAL	BLSUTRMV	BLSDMSGS
	BLSCANLI	BLSUTRMV	BLSDMSGS
	BLSCFAMS	BLSUTRMV	BLSDMSGS
BLS03110I	BLSCAAMS	BLSUTRMV	BLSDMSGS
	BLSCANAL	BLSUTRMV	BLSDMSGS
	BLSCANLE	BLSUTRMV	BLSDMSGS
	BLSCANLI	BLSUTRMV	BLSDMSGS
	BLSCFAMS	BLSUTRMV	BLSDMSGS
BLS03111I	BLSCAAMS	BLSUTRMV	BLSDMSGS
	BLSCFAMS	BLSUTRMV	BLSDMSGS
BLS03112I	BLSCADYN	BLSUTRMV	BLSDMSGS
	BLSCFDYN	BLSUTRMV	BLSDMSGS
BLS03113I	BLSCAMRE	BLSUTRMV	BLSDMSGS
BLS03114I	BLSCENDD	BLSUTRMV	BLSDMSGS
	BLSCREAS	BLSUTRMV	BLSDMSGS
	BLSCGETT	BLSUTRMV	BLSDMSGS
	BLSCPDIN	BLSUTRMV	BLSDMSGS
	BLSCPUTT	BLSUTRMV	BLSDMSGS
	BLSCSETT	BLSUTRMV	BLSDMSGS
BLS03115I	BLSCENDD	BLSUTRMV	BLSDMSGS
	BLSCREAS	BLSUTRMV	BLSDMSGS
	BLSCGETT	BLSUTRMV	BLSDMSGS
	BLSCPOIN	BLSUTRMV	BLSDMSGS
	BLSCPUTT	BLSUTRMV	BLSDMSGS
BLS03116I	BLSCENDD	BLSUTRMV	BLSDMSGS
	BLSCREAS	BLSUTRMV	BLSDMSGS
	BLSCGETT	BLSUTRMV	BLSDMSGS
	BLSCPOIN	BLSUTRMV	BLSDMSGS
	BLSCPUTT	BLSUTRMV	BLSDMSGS
	BLSCSETT	BLSUTRMV	BLSDMSGS
BLS03117I	BLSCFAMS	BLSUTRMV	BLSDMSGS
	BLSCFDYN	BLSUTRMV	BLSDMSGS
BLS03118I	BLSCFDYN	BLSUTRMV	BLSDMSGS
BLS03119I	BLSCFAMS	BLSUTRMV	BLSDMSGS
BLS04000I	BLSEMDPR	BLSUTRMV	BLSDMSGS
	BLSFAV00	BLSUTRMV	BLSDMSGS
	BLSFMD00	BLSUTRMV	BLSDMSGS
BLS04001I	BLSEDEL0	BLSUTRMV	BLSDMSGS
	BLSELPDR	BLSUTRMV	BLSDMSGS
BLS04002I	BLSEDEL0	BLSUTRMV	BLSDMSGS
	BLSELPDR	BLSUTRMV	BLSDMSGS
BLS04003I	BLSEDEL0	BLSUTRMV	BLSDMSGS
	BLSELPDR	BLSUTRMV	BLSDMSGS
	BLSEMDPR	BLSUTRMV	BLSDMSGS
	BLSFPARS	BLSUTRMV	BLSDMSGS
	BLSFPS00	BLSUTRMV	BLSDMSGS
BLS04004I	BLSEMDPR	BLSUTRMV	BLSDMSGS
	BLSFMD00	BLSUTRMV	BLSDMSGS
	BLSFVRFY	BLSUTRMV	BLSDMSGS
BLS04005I	BLSFLD00	BLSUTRMV	BLSDMSGS
	BLSFMD00	BLSUTRMV	BLSDMSGS
BLS04008I	BLSFDS00	BLSUTRMV	BLSDMSGS
	BLSFPARS	BLSUTRMV	BLSDMSGS
BLS04009I	BLSFMD00	BLSUTRMV	BLSDMSGS
BLS04010I	BLSFAV00	BLSUTRMV	BLSDMSGS
	BLSFMD00	BLSUTRMV	BLSDMSGS
BLS04011I	BLSEADP1	BLSUTRMV	BLSDMSGS
BLS04012I	BLSEADP1	BLSUTRMV	BLSDMSGS
BLS04014I	BLSELPFM	BLSUTRMV	BLSDMSGS
BLS04015I	BLSELPFR	BLSUTRMV	BLSDMSGS
	BLSFLONE	BLSUTRMV	BLSDMSGS
BLS04016I	BLSDIFP0	BLSUTRMV	BLSDIFP0

Message ID	Detecting Module	Issuing Module	Containing Module
BLS04017I	BLSDIFP0	BLSUTRMV	BLSDIFP0
	BLSDIFP1	BLSDIFP1	BLSDIFP1
BLS04018I	BLSDIFP0	BLSUTRMV	BLSDIFP0
	BLSDIFP1	BLSDIFP1	BLSDIFP1
BLSS04019I	BLSDIFP0	BLSUTRMV	BLSDIFP0
	BLSDIFP1	BLSDIFP1	BLSDIFP1
BLS04040I	BLSDAPDR	BLSUTRMV	BLSDMSGS
BLS04041I	BLSEADPR	BLSUTRMV	BLSDMSGS
	BLSEMDPR	BLSUTRMV	BLSDMSGS
	BLSFMD00	BLSUTRMV	BLSDMSGS
BLS04042I	BLSEADPR	BLSUTRMV	BLSDMSGS
	BLSEDELC	BLSUTRMV	BLSDMSGS
	BLSEDEL0	BLSUTRMV	BLSDMSGS
	BLSELPCT	BLSUTRMV	BLSDMSGS
	BLSELPDR	BLSUTRMV	BLSDMSGS
	BLSEMDPR	BLSUTRMV	BLSDMSGS
	BLSFGP00	BLSUTRMV	BLSDMSGS
	BLSFOPEN	BLSUTRMV	BLSDMSGS
	BLSFUP00	BLSUTRMV	BLSDMSGS
	BLSFVRFY	BLSUTRMV	BLSDMSGS
BLS04043I	BLSEADPR	BLSUTRMV	BLSDMSGS
	BLSEDEL0	BLSUTRMV	BLSDMSGS
	BLSELPDR	BLSUTRMV	BLSDMSGS
	BLSELPFR	BLSUTRMV	BLSDMSGS
	BLSELPOP	BLSUTRMV	BLSDMSGS
	BLSEMDPR	BLSUTRMV	BLSDMSGS
	BLSEMDP1	BLSUTRMV	BLSDMSGS
	BLSFGP00	BLSUTRMV	BLSDMSGS
	BLSFMV00	BLSUTRMV	BLSDMSGS
	BLSFPREA	BLSUTRMV	BLSDMSGS
	BLSFPRES	BLSUTRMV	BLSDMSGS
	BLSFUP00	BLSUTRMV	BLSDMSGS
	BLSFVRFY	BLSUTRMV	BLSDMSGS
BLS04044I	BLSEADP1	BLSUTRMV	BLSDMSGS
	BLSEMDPR	BLSUTRMV	BLSDMSGS
	BLSEMDP1	BLSUTRMV	BLSDMSGS
	BLSFREAS	BLSUTRMV	BLSDMSGS
	BLSFUP00	BLSUTRMV	BLSDMSGS
BLS04045I	BLSEADPR	BLSUTRMV	BLSDMSGS
	BLSEDELC	BLSUTRMV	BLSDMSGS
	BLSEDEL0	BLSUTRMV	BLSDMSGS
	BLSELPCT	BLSUTRMV	BLSDMSGS
	BLSELPDR	BLSUTRMV	BLSDMSGS
	BLSEMDPR	BLSUTRMV	BLSDMSGS
	BLSFCLOS	BLSUTRMV	BLSDMSGS
	BLSFGP00	BLSUTRMV	BLSDMSGS
	BLSFUP00	BLSUTRMV	BLSDMSGS
BLS04046I	BLSDFPDR	BLSUTRMV	BLSDMSGS
BLS04047I	BLSDFPDR	BLSUTRMV	BLSDMSGS
BLS04050I	BLSDADSD	BLSUTRMV	BLSDMSGS
BLS04051I	BLSFLD00	BLSUTRMV	BLSDMSGS
	BLSFLP00	BLSUTRMV	BLSDMSGS
	BLSFMD00	BLSUTRMV	BLSDMSGS
BLS04052I	BLSFGD00	BLSUTRMV	BLSDMSGS
	BLSFGG00	BLSUTRMV	BLSDMSGS
	BLSFLD00	BLSUTRMV	BLSDMSGS
	BLSFOPEN	BLSUTRMV	BLSDMSGS
	BLSFUD00	BLSUTRMV	BLSDMSGS
BLS04053I	BLSFDEL	BLSUTRMV	BLSDMSGS
	BLSFDRES	BLSUTRMV	BLSDMSGS
	BLSFGD00	BLSUTRMV	BLSDMSGS
	BLSFGG00	BLSUTRMV	BLSDMSGS
	BLSFLD00	BLSUTRMV	BLSDMSGS
	BLSFLV00	BLSUTRMV	BLSDMSGS
	BLSFMV00	BLSUTRMV	BLSDMSGS
	BLSFUD00	BLSUTRMV	BLSDMSGS
BLS04054I	BLSFREAS	BLSUTRMV	BLSDMSGS
	BLSFUD00	BLSUTRMV	BLSDMSGS
BLS04055I	BLSFCLOS	BLSUTRMV	BLSDMSGS
	BLSFGD00	BLSUTRMV	BLSDMSGS
	BLSFGG00	BLSUTRMV	BLSDMSGS
	BLSFLD00	BLSUTRMV	BLSDMSGS
	BLSFUD00	BLSUTRMV	BLSDMSGS
BLS04060I	BLSFSD00	BLSUTRMV	BLSDMSGS
BLS04061I	BLSFOD00	BLSUTRMV	BLSDMSGS
BLS04062I	BLSFOD00	BLSUTRMV	BLSDMSGS
BLS04063D	BLSFOD00	BLSUTRMV	BLSDMSGS
BLS04064I	BLSFOD00	BLSUTRMV	BLSDMSGS
BLS04065I	BLSFPS00	BLSUTRMV	BLSDMSGS
BLS04066I	BLSFCN00	BLSUTRMV	BLSDMSGS
BLS04067D	BLSFCN00	BLSUTRMV	BLSDMSGS
BLS04068I	BLSFAD00	BLSUTRMV	BLSDMSGS
BLS04069I	BLSEADPR	BLSUTRMV	BLSDMSGS
	BLSEMDPR	BLSUTRMV	BLSDMSGS
	BLSFAD00	BLSUTRMV	BLSDMSGS
	BLSFMD00	BLSUTRMV	BLSDMSGS
BLS04070I	BLSEADPR	BLSUTRMV	BLSDMSGS
	BLSEADP1	BLSUTRMV	BLSDMSGS
	BLSEMDPR	BLSUTRMV	BLSDMSGS
BLS04071I	BLSCALOC	BLSUTRMV	BLSDMSGS
	BLSCRTCB	BLSUTRMV	BLSDMSGS
	BLSDFPDR	BLSUTRMV	BLSDMSGS
BLS04072I	BLSCALOC	BLSUTRMV	BLSDMSGS
	BLSCRTCB	BLSUTRMV	BLSDMSGS
	BLSDFPDR	BLSUTRMV	BLSDMSGS
BLS04073I	BLSFSCRT	BLSUTRMV	BLSDMSGS
BLS04074I	BLSFSCRT	BLSUTRMV	BLSDMSGS
BLS04075I	BLSFSCRT	BLSUTRMV	BLSDMSGS
BLS04076I	BLSFSCRT	BLSUTRMV	BLSDMSGS
BLS04077I	BLSFSCRT	BLSUTRMV	BLSDMSGS

Message ID	Detecting Module	Issuing Module	Containing Module
BLS04078I	BLSFSCRT	BLSUTRMV	BLSDMSGS
BLS04079I	BLSFSCRT	BLSUTRMV	BLSDMSGS
BLS04080I	BLSDC128	BLSUTRMV	BLSDMSGS
	BLSDC600	BLSUTRMV	BLSDMSGS
BLS04081I	BLSDC070	BLSUTRMV	BLSDMSGS
BLS04082I	BLSFUD00	BLSUTRMV	BLSDMSGS
	BLSFUP00	BLSUTRMV	BLSDMSGS
BLS04083I	BLSFAD00	BLSUTRMV	BLSDMSGS
BLS04084I	BLSFDEL	BLSUTRMV	BLSDMSGS
BLS04085I	BLSFDEL	BLSUTRMV	BLSDMSGS
BLS04086I	BLSFDEL	BLSUTRMV	BLSDMSGS
BLS04087I	BLSFDEL	BLSUTRMV	BLSDMSGS
BLS04088I	BLSFDEL	BLSUTRMV	BLSDMSGS
BLS04089I	BLSFDALL	BLSUTRMV	BLSDMSGS
BLS04090I	BLSFLALL	BLSUTRMV	BLSDMSGS
BLS04091I	BLSFLP00	BLSUTRMV	BLSDMSGS
BLS05100I	BLSEADPR	BLSUTRMV	BLSDMSGS
BLS05101I	BLSEADPR	BLSUTRMV	BLSDMSGS
BLS05104I	BLSEADPR	BLSUTRMV	BLSDMSGS
BLS05301I	BLSELPDR	BLSUTRMV	BLSDMSGS
BLS05400I	BLSEAUTH	BLSUTRMV	BLSDMSGS
BLS05401I	BLSEDELC	BLSUTRMV	BLSDMSGS
BLS05402D	BLSEDEL1	BLSUTRMV	BLSDMSGS
	BLSFCONF	BLSUTRMV	BLSDMSGS
BLS06401I	BLSFCONF	BLSUTRMV	BLSDMSGS
BLS06402I	BLSFVRFY	BLSUTRMV	BLSDMSGS
BLS07001I	BLSGX	BLSUTRMV	BLSGX
BLS07002I	BLSGX	BLSUTRMV	BLSGX
BLS17001I	BLSQEXTI	BLSQEXTI	BLSQEXTI
BLS17002I	BLSQCFMT	BLSQCFMT	BLSQCFMT
	BLSQIFMT	BLSQIFMT	BLSQIFMT
BLS17003I	BLSQCFMT	BLSQCFMT	BLSQCFMT
BLS17004I	BLSQCFMT	BLSQCFMT	BLSQCFMT
BLS17005I	BLSQCFMT	BLSQCFMT	BLSQCFMT
BLS17006I	BLSQECT	BLSQECT	BLSQECT
BLS17007I	BLSQECT	BLSQECT	BLSQECT
BLS17009I	BLSQECT	BLSQECT	BLSQECT
BLS17013I	BLSQECT	BLSQECT	BLSQECT
BLS17014I	BLSQTRAP	BLSQTRAP	BLSQTRAP
BLS17015I	BLSQPRMA	BLSQPRMA	BLSQPRMA
BLS17016I	BLSQTRAP	BLSQTRAP	BLSQTRAP
BLS17017I	BLSQPRMR	BLSQPRMR	BLSQPRMR
BLS17018I	BLSQPRMR	BLSQPRMR	BLSQPRMR
BLS17019I	BLSQPRMR	BLSQPRMR	BLSQPRMR
BLS17040I	BLSQFORM	BLSQFORM	BLSQFORM
BLS18000I	BLSRASM7	BLSUTRMV	BLSRASM7
BLS18001I	BLSRASM7	BLSUTRMV	BLSRASM7
BLS18002I	BLSRASM7	BLSUTRMV	BLSRASM7
BLS18003I	BLSRASM7	BLSUTRMV	BLSRASM7
BLS18004I	BLSRASM7	BLSUTRMV	BLSRASM7
BLS18005I	BLSRASM7	BLSUTRMV	BLSRASM7
BLS18006I	BLSRCOMK	BLSUTRMV	BLSRCOMK
BLS18007I	BLSRCOMK	BLSUTRMV	BLSRCOMK
BLS18008I	BLSRCOMK	BLSUTRMV	BLSRCOMK
BLS18009I	BLSRCOMK	BLSUTRMV	BLSRCOMK
BLS18010I	BLSRCOMK	BLSUTRMV	BLSRCOMK
BLS18011I	BLSRDUSC	BLSUTRMV	BLSRDUSC
BLS18012I	BLSRDUSC	BLSUTRMV	BLSRDUSC
BLS18013I	BLSRDUSC	BLSUTRMV	BLSRDUSC
BLS18014I	BLSRDUSC	BLSUTRMV	BLSRDUSC
BLS18015I	BLSRFMOD	BLSUTRMV	BLSRFMOD
BLS18016I	BLSRFMOD	BLSUTRMV	BLSRFMOD
BLS18025I	BLSRFIND	BLSUTRMV	BLSRFIND
BLS18026I	BLSRFIND	BLSUTRMV	BLSRFIND
BLS18027I	BLSRFIND	BLSUTRMV	BLSRFIND
BLS18028I	BLSRFIND	BLSUTRMV	BLSRFIND
BLS18029I	BLSRCOMP	BLSUTRMV	BLSRCOMP
BLS18030I	BLSRCOMP	BLSUTRMV	BLSRCOMP
BLS18031I	BLSRCOMP	BLSUTRMV	BLSRCOMP
BLS18038I	BLSRSCAN	BLSUTRMV	BLSRSCAN
BLS18039I	BLSRSCAN	BLSUTRMV	BLSRSCAN
BLS18040I	BLSRENQA	BLSUTRMV	BLSRENQA
BLS18041I	BLSRENQA	BLSUTRMV	BLSRENQA
BLS18042I	BLSRENQA	BLSUTRMV	BLSRENQA
BLS18043I	BLSRFIND	BLSUTRMV	BLSRFIND
BLS18046I	BLSRAMDX	BLSUTRMV	BLSRAMDX
BLS18048I	BLSRSALI	BLSUTRMV	BLSRSALI
BLS18050I	BLSRADDR	BLSUTRMV	BLSRADDR
BLS18051I	BLSRADD1	BLSUTRMV	BLSRADD1
BLS18052I	BLSRADD1	BLSUTRMV	BLSRADD1
BLS18053I	BLSRADD1	BLSUTRMV	BLSRADD1
BLS18054I	BLSRADDP	BLSUTRMV	BLSRADDP
BLS18058I	BLSRSAPC	BLSUTRMV	BLSRSAPC
BLS18059I	BLSRSAPC	BLSUTRMV	BLSRSAPC
BLS18060I	BLSRDUIS	BLSUTRMV	BLSRDUIS
BLS18061I	BLSRVSLD	BLSUTRMV	BLSRVSLD
BLS18063I	BLSRENQK	BLSUTRMV	BLSRENQK
BLS18064I	BLSRESRV	BLSUTRMV	BLSRESRV
BLS18065I	BLSRESRV	BLSUTRMV	BLSRESRV
BLS18066I	BLSRESRV	BLSUTRMV	BLSRESRV
BLS18067I	BLSRESRV	BLSUTRMV	BLSRESRV
BLS18068I	BLSRESRV	BLSUTRMV	BLSRESRV
BLS18069I	BLSRESRV	BLSUTRMV	BLSRESRV
BLS18070I	BLSRADD1	BLSUTRMV	BLSRADD1
BLS18071I	BLSQECT	BLSQECT	BLSQECT
BLS18072I	BLSQECT	BLSQECT	BLSQECT
BLS18073I	BLSQECT	BLSQECT	BLSQECT
BLS18074I	BLSRADDP	BLSUTRMV	BLSRADDP
BLS18075I	BLSRADD1	BLSUTRMV	BLSRADD1
BLS18076I	BLSRSCAN	BLSUTRMV	BLSRSCAN

Message ID	Detecting Module	Issuing Module	Containing Module
BLS18077I	BLSRM077	BLSUTRMV	BLSRM077
BLS18078I	BLSRM011	BLSRM077	BLSRM077
BLS18079I	BLSRM011	BLSRM077	BLSRM077
BLS18081I	BLSR3270	BLSUTRMV	BLSR3270
BLS18082I	BLSR3270	BLSUTRMV	BLSR3270
BLS18087I	BLSRCOMP	BLSUTRMV	BLSRCOMP
BLS18090I	BLSRVPAS	BLSUTRMV	BLSRVPAS
BLS18092I	BLSRVPCP	BLSUTRMV	BLSRVPCP
BLS18094I	BLSRRNCH	BLSUTRMV	BLSRRNCH
BLS18095I	BLSRFMOD	BLSUTRMV	BLSRFMOD
BLS18100I	BLSRACC	BLSUTRMV	BLSRACC
BLS18102I	BLSRESCK	BLSUTRMV	BLSRESCK
BLS18104I	BLSRESGU	BLSUTRMV	BLSRESGU
BLS18114I	BLSRSADE	BLSUTRMV	BLSRSADE
BLS18116I	BLSRENQA	BLSUTRMV	BLSRENQA
BLS18122I	BLSRDUAL	BLSUTRMV	BLSRDUAL
BLS18123I	BLSRDUAL	BLSUTRMV	BLSRDUAL
BLS18124I	BLSRDUIH	BLSUTRMV	BLSRDUIH
BLS18125I	BLSRDUIC	BLSUTRMV	BLSRDUIC
BLS18126I	BLSRDUIC	BLSUTRMV	BLSRDUIC
BLS18127I	BLSRDUIH	BLSUTRMV	BLSRDUIH
BLS18128I	BLSRDUIC	BLSUTRMV	BLSRDUIC
BLS18130I	BLSRDUAL	BLSUTRMV	BLSRDUAL
BLS18132I	BLSRDUAL	BLSUTRMV	BLSRDUAL
BLS18133I	BLSRDUOP	BLSUTRMV	BLSRDUOP
BLS18134I	BLSRDUSC	BLSUTRMV	BLSRDUSC
BLS18135I	BLSRDUDE	BLSRDUDE	BLSRDUDE
BLS18150I	BLSRRNCH	BLSUTRMV	BLSRRNCH
BLS18151I	BLSRRNCH	BLSUTRMV	BLSRRNCH
BLS18154I	BLSRDUOP	BLSUTRMV	BLSRDUOP
BLS18155I	BLSRCOTA	BLSUTRMV	BLSDMSGS
BLS18156I	BLSRCOIN	BLSUTRMV	BLSDMSGS
BLS18160D	BLSRDUIS	BLSUTRMV	BLSDMSGS
BLS18161I	BLSRST01	BLSUTRMV	BLSRST01
BLS18162I	BLSRCOIN	BLSUTRMV	BLSDMSGS
BLS18163I	BLSRCOPY	BLSUTRMV	BLSDMSGS
BLS18164I	BLSRCOTA	BLSUTRMV	BLSDMSGS
BLS18165I	BLSRCOTA	BLSUTRMV	BLSDMSGS
BLS18166I	BLSRCOPY	BLSUTRMV	BLSDMSGS
BLS18167I	BLSRCOHD	BLSUTRMV	BLSDMSGS
BLS18168D	BLSRCOHD	BLSUTRMV	BLSDMSGS
BLS18169I	BLSRCOPY	BLSUTRMV	BLSDMSGS
BLS18170I	BLSRCOPY	BLSUTRMV	BLSDMSGS
BLS18171I	BLSRCOPY	BLSUTRMV	BLSDMSGS
BLS18172I	BLSRCOIN	BLSUTRMV	BLSDMSGS
BLS18173I	BLSRCOHD	BLSUTRMV	BLSDMSGS
BLS18174I	BLSRST01	BLSUTRMV	BLSRST01
BLS18175I	BLSRCOMP	BLSUTRMV	BLSRCOMP
BLS18176I	BLSRDUAL	BLSUTRMV	BLSRDUAL
BLS18177I	BLSRDUAL	BLSUTRMV	BLSRDUAL
BLS18178I	BLSRDUAL	BLSUTRMV	BLSRDUAL
BLS18179I	BLSRDUAL	BLSUTRMV	BLSRDUAL
BLS18180I	BLSRDUAL	BLSUTRMV	BLSRDUAL
BLS18181I	BLSRDUAL	BLSUTRMV	BLSRDUAL
BLS18182I	BLSRDUAL	BLSUTRMV	BLSRDUAL
BLS18183I	BLSRDUBF	BLSUTRMV	BLSRDUBF
BLS18185I	BLSRDUOP	BLSUTRMV	BLSRDUOP
BLS18186I	BLSRDUO2	BLSUTRMV	BLSRDUO2
BLS18187I	BLSRDUO3	BLSUTRMV	BLSRDUO3
BLS18188I	BLSRDUO4	BLSUTRMV	BLSRDUO4
BLS18189I	BLSRDUO5	BLSUTRMV	BLSRDUO5
BLS18190I	BLSRDUO5	BLSUTRMV	BLSRDUO5
BLS18191I	BLSRVAL	BLSUTRMV	BLSRVAL
BLS18192I	BLSRM192	BLSUTRMV	BLSRM192
BLS18193I	BLSRM193	BLSUTRMV	BLSRM193
BLS18194I	BLSRVALC	BLSUTRMV	BLSRVALC
BLS18195I	BLSRVALC	BLSUTRMV	BLSRVALC
BLS18196I	BLSRVALC	BLSUTRMV	BLSRVALC
BLS18197I	BLSRVALP	BLSRVALP	BLSRVALP
BLS18198I	BLSRVALP	BLSRVALP	BLSRVALP
BLS18199I	BLSRM199	BLSRM199	BLSRM199
BLS18200I	BLSRM200	BLSRM200	BLSRM200
BLS18205I	BLSRVALX	BLSUTRMV	BLSRVALX
BLS18206I	BLSRDUDE	BLSRDUDE	BLSRDUDE
BLS18210I	BLSRACCL	BLSUTRMV	BLSRACCL
BLS18211I	BLSRAMDX	BLSUTRMV	BLSRAMDX
BLS18212I	BLSRDUAL	BLSRDUAL	BLSRDUAL
BLS18213I	BLSRFIND	BLSRFIND	BLSRFIND
BLS18214I	BLSRM214	BLSRM214	BLSRM214
BLS18215I	BLSRESST	BLSRESST	BLSRESST
BLS18216I	BLSRCOPY BLSRDUIN	BLSRM216	BLSRM216
BLS18300I	BLSRM300	BLSUTRMV	BLSRM300
BLS18301I	BLSRM301	BLSUTRMV	BLSRM301
BLS18302I	BLSRM302	BLSUTRMV	BLSRM302
BLS18400I	BLSRSTFR	BLSRSTFR	BLSRSTFR
BLS18401I	BLSRSTFR	BLSRSTFR	BLSRSTFR
BLS18402I	BLSRSTFR	BLSRSTFR	BLSRSTFR
BLS18405I	BLSRSTWS	BLSRSTWS	BLSRSTWS
BLS18451I	BLSRWHRS	BLSRWHER	BLSRWHER
BLS18452I	BLSRWHRS	BLSRWHRS	BLSRWHRS
BLS19000I	BLSSASCB	BLSUTRMV	BLSSASCB
BLS19001I	BLSSASCB	BLSUTRMV	BLSSASCB
BLS19002I	BLSSLCCA	BLSUTRMV	BLSSLCCA
BLS19003I	BLSSPCCA	BLSUTRMV	BLSSPCCA
BLS20000I	BLST08	BLSUTRMV	BLST08
BLS21002I	BLSUMONA	BLSUTRMV	BLSUMONA
BLS21003I	BLSUMONA	BLSUTRMV	BLSUMONA
BLS21006I	BLSUMONL	BLSUTRMV	BLSUMONL
BLS21007I	BLSUMONL	BLSUTRMV	BLSUMONL

Message ID	Detecting Module	Issuing Module	Containing Module
BLS21020I	BLSUVSOP	BLSUTRMV	BLSUVSOP
BLS21021I	BLSUVSOP	BLSUTRMV	BLSUVSOP
BLS21022I	BLSUVSAR	BLSUTRMV	BLSUVSAR
BLS21024I	BLSUVSOQ	BLSUTRMV	BLSUVSOQ
BLS21025D	BLSUMON	BLSUTRMV	BLSUMON
BLS21030I	BLSUPARU	BLSUTRMV	BLSUPARU
BLS21031I	BLSUPROP	BLSUTRMV	BLSUPROP
BLS21032I	BLSUBLDL	BLSUTRMV	BLSUBLDL
BLS21033I	BLSUBLDL	BLSUTRMV	BLSUBLDL
BLS21034I	BLSUBLDL	BLSUTRMV	BLSUBLDL
BLS21043I	BLSUPROP	BLSUTRMV	BLSUPROP
BLS21044I	BLSUPROP	BLSUTRMV	BLSUPROP
BLS21045I	BLSUPUT	BLSUTRMV	BLSUPUT
BLS21050I	BLSUTOC	BLSUTOC	BLSUTOC
BLS21051I	BLSUVSCR	BLSUTRMV	BLSUVSCR
BLS21052I	BLSUVSMR	BLSUTRMV	BLSUVSMR
BLS21053I	BLSU	BLSUTRMV	BLSU
BLS21054I	BLSU	BLSUTRMV	BLSU
BLS21055I	BLSUMON2	BLSUTRMV	BLSUMON2
BLS21056I	BLSUMON2	BLSUTRMV	BLSUMON2
BLS21058I	BLSUTSO	BLSUTRMV	BLSUTSO
BLS21061I	BLSUVSWB	BLSUTRMV	BLSUVSWB
BLS21062I	BLSUMOLD	BLSUTRMV	BLSUMOLD
	BLSUMNOL	BLSUMNOL	BLSUMNOL
BLS21066I	BLSUVSOQ	BLSUTRMV	BLSUVSOQ
BLS21067I	BLSUVSOQ	BLSUTRMV	BLSUVSOQ
BLS21068I	BLSUVSOQ	BLSUTRMV	BLSUVSOQ
BLS21069I	BLSUPGML	BLSUTRMV	BLSUPGML
BLS21070I	BLSUPGMU	BLSUTRMV	BLSUPGMU
BLS21071I	BLSUVSCA	BLSUTRMV	BLSUVSCA
BLS21072I	BLSUSTKS	BLSUSTKS	BLSUSTKS
BLS21075I	BLSUCLOS	BLSUTRMV	BLSUCLOS
BLS21077I	BLSUCLOS	BLSUTRMV	BLSUCLOS
BLS21078I	BLSUOPEN	BLSUTRMV	BLSUOPEN
BLS22002I	BLSVASVT	BLSUTRMV	BLSVASVT
BLS22003I	BLSVASVT	BLSUTRMV	BLSVASVT
BLS22004I	BLSVASVT	BLSUTRMV	BLSVASVT
BLS22007I	BLSVRB	BLSUTRMV	BLSVRB
BLS22008I	BLSVRB	BLSUTRMV	BLSVRB
BLS22012I	BLSVUCB	BLSUTRMV	BLSVUCB
BLS22018I	BLSVASCB	BLSUTRMV	BLSVASCB
BLS22020I	BLSVASCB	BLSUTRMV	BLSVASCB
BLS22021I	BLSVASCB	BLSUTRMV	BLSVASCB
BLS22022I	BLSVASCB	BLSVASCB	BLSVASCB
BLS22025I	BLSVASSB	BLSVASSB	BLSVASSB
BLS22029I	BLSVLCCA	BLSUTRMV	BLSVLCCA
BLS22033I	BLSVCVT	BLSUTRMV	BLSVCVT
BLS22036I	BLSVCVT	BLSUTRMV	BLSVCVT
BLS22037I	BLSVCVT	BLSUTRMV	BLSVCVT
BLS22038I	BLSVCVT	BLSUTRMV	BLSVCVT
BLS22039I	BLSVCVT	BLSUTRMV	BLSVCVT
BLS22040I	BLSVCVT	BLSUTRMV	BLSVCVT
BLS22041I	BLSVCVT	BLSUTRMV	BLSVCVT
BLS22042I	BLSVCVT	BLSUTRMV	BLSVCVT
BLS22043I	BLSVCVT	BLSUTRMV	BLSVCVT
BLS22052I	BLSVTCB	BLSUTRMV	BLSVTCB
BLS22054I	BLSVTCB	BLSUTRMV	BLSVTCB
BLS22055I	BLSVTCB	BLSUTRMV	BLSVTCB
BLS22073I	BLSVPCCA	BLSUTRMV	BLSVPCCA
BLS22080I	BLSVPGTE	BLSUTRMV	BLSVPGTE
BLS22083I	BLSVPSA	BLSUTRMV	BLSVPSA
BLS22084I	BLSVPSA	BLSUTRMV	BLSVPSA
BLS22085I	BLSVPSA	BLSUTRMV	BLSVPSA
BLS22107I	BLSVSGTE	BLSUTRMV	BLSVSGTE
BLS22111I	BLSVXTLS	BLSUTRMV	BLSVXTLS
BLS22112I	BLSVXTLS	BLSUTRMV	BLSVXTLS
BLS22114I	BLSVXTLS	BLSUTRMV	BLSVXTLS
BLS22117I	BLSVASXB	BLSUTRMV	BLSVASXB
BLS22118I	BLSVASXB	BLSUTRMV	BLSVASXB
BLS22141I	BLSVRTCT	BLSUTRMV	BLSVRTCT
BLS22142I	BLSVCVT2	BLSUTRMV	BLSVCVT2
BLS22144I	BLSVCVT2	BLSUTRMV	BLSVCVT2
BLS22148I	BLSVASTE	BLSUTRMV	BLSVASTE
BLS22151I	BLSVGVT	BLSUTRMV	BLSVGVT
BLS22155I	BLSVGVTX	BLSUTRMV	BLSVGVTX
BLS22159I	BLSVQCB	BLSUTRMV	BLSVQCB
BLS22167I	BLSVQHT	BLSUTRMV	BLSVQHT
BLS22170I	BLSVRSV	BLSUTRMV	BLSVRSV
BLS22187I	BLSVUCM	BLSUTRMV	BLSVUCM
BLS22189I	BLSVUCM	BLSUTRMV	BLSVUCM
BLS22192I	BLSVORE	BLSUTRMV	BLSVORE
BLS22303I	BLSVCSD	BLSUTRMV	BLSVCSD
BLS22304I	BLSVCSD	BLSUTRMV	BLSVCSD
BLS22305I	BLSVCSD	BLSUTRMV	BLSVCSD
BLS22306I	BLSVCSD	BLSUTRMV	BLSVCSD
BLS22307I	BLSVCSD	BLSUTRMV	BLSVCSD
BLS22308I	BLSVCSD	BLSUTRMV	BLSVCSD
BLS22309I	BLSVCSD	BLSUTRMV	BLSVCSD
IAR10001I	IARRD	IARRD	IARRD
	IARR3	IARR3	IARR3
IAR10002I	IARRD	IARRD	IARRD
	IARR3	IARR3	IARR3
IAR10003I	IARRD	IARRD	IARRD
	IARR3	IARR3	IARR3
IAR10004I	IARRD	IARRD	IARRD
IAR10005I	IARQM	IARQM	IARQM
	IARR7	IARR7	IARR7
IAR10006I	IARQM	IARQM	IARQM
	IARR7	IARR7	IARR7
IAR10007I	IARRO	IARRO	IARRO

Message ID	Detecting Module	Issuing Module	Containing Module
IAR10008I	IARRU	IARRU	IARRU
	IARR6	IARR6	IARR6
IAR10009I	IARRU	IARRU	IARRU
	IARR6	IARR6	IARR6
IAR10010I	IARRM	IARRM	IARRM
	IARR7	IARR7	IARR7
IAR10011I	IARR5	IARR5	IARR5
IAR10012I	IARR1	IARR1	IARR1
IAR10013I	IARR1	IARR1	IARR1
IAR10014I	IARR5	IARR5	IARR5
IAR10015I	IARR5	IARR5	IARR5
IAR10016I	IARRF	IARRF	IARRF
IAR10017I	IARRD	IARRD	IARRD
IAR10018I	IARQM	IARQM	IARQM
IAR10019I	IARQM	IARQM	IARQM
IAR10020I	IARQM	IARQM	IARQM
IAR10021I	IARR1	IARR1	IARR1
IAR10022I	IARRu	IARRU	IARRU
IAR10023I	IARR5	IARR5	IARR5
IAR10024I	IARRM	IARRM	IARRM
IAR10025I	IARRM	IARRM	IARRM
IAR10026I	IARRO	IARRO	IARRO
IAR10027I	IARRO	IARRO	IARRO
IAR10028I	IARRM	IARRM	IARRM
	IARR8	IARR8	IARR8
IAR10029I	IARRM	IARRM	IARRM
IAR10030I	IARRM	IARRM	IARRM
	IARR8	IARR8	IARR8
IAR10031I	IARR3	IARR3	IARR3
IAR10032I	IARR3	IARR3	IARR3
IAR10033I	IARR8	IARR8	IARR8
IEA11001I	IEAVESLX	IEAVESLX	IEAVESLX
IEA21001I	IEAVTRCA	IEAVTRCA	IEAVTRCA
IEA21002I	IEAVTRCA	IEAVTRCA	IEAVTRCA
IEA21003I	IEAVTRCA	IEAVTRCA	IEAVTRCA
IEA21004I	IEAVTRCA	IEAVTRCA	IEAVTRCA
IEA21005I	IEAVTRCA	IEAVTRCA	IEAVTRCA
IEA21006I	IEAVTRCA	IEAVTRCA	IEAVTRCA
IEA21007I	IEAVTRCA	IEAVTRCA	IEAVTRCA
IEA21010I	IEAVTRCA	IEAVTRCA	IEAVTRCA
IEA24001I	IEAVTREF	IEAVTREF	IEAVTREF
IEA24002I	IEAVTREF	IEAVTREF	IEAVTREF
IEA24003I	IEAVTREF	IEAVTREF	IEAVTREF
IEA24004I	IEAVTREF	IEAVTREF	IEAVTREF
IEA24005I	IEAVTREF	IEAVTREF	IEAVTREF
IEA24006I	IEAVTREF	IEAVTREF	IEAVTREF
IEA24007I	IEAVTREF	IEAVTREF	IEAVTREF
IEA24008I	IEAVTREF	IEAVTREF	IEAVTREF
IEA24009I	IEAVTREF	IEAVTREF	IEAVTREF
IEA24010I	IEAVTREF	IEAVTREF	IEAVTREF
IEA24011I	IEAVTREF	IEAVTREF	IEAVTREF
IEA24012I	IEAVTREF	IEAVTREF	IEAVTREF
IEA24050I	IEAVTREF	IEAVTREF	IEAVTREF
IEA24060I	IEAVTREF	IEAVTREF	IEAVTREF
IEA31001I	IEAVG702	IEAVG702	IEAVG702
IEA31002I	IEAVG702	IEAVG702	IEAVG702
IEA31003I	IEAVG702	IEAVG702	IEAVG702
IEA31004I	IEAVG702	IEAVG702	IEAVG702
IEA31005I	IEAVG702	IEAVG702	IEAVG702
IEA31006I	IEAVG705	IEAVG705	IEAVG705
IEA31007I	IEAVG704	IEAVG704	IEAVG704
IEA31009I	IEAVG701	IEAVG701	IEAVG701
IEE30001I	IEEMB817	IEEMB817	IEEMB817
IEE30002I	IEEMB817	IEEMB817	IEEMB817
IEE30003I	IEEMB817	IEEMB817	IEEMB817
IEE30004I	IEEMB817	IEEMB817	IEEMB817
IEE30005I	IEEMB817	IEEMB817	IEEMB817
IEE30006I	IEEMB817	IEEMB817	IEEMB817
IEE30007I	IEEMB817	IEEMB817	IEEMB817
IEE30008I	IEEMB817	IEEMB817	IEEMB817
IEE30009I	IEEMB817	IEEMB817	IEEMB817
IEE30010I	IEEMB817	IEEMB817	IEEMB817
IEE30011I	IEEMB817	IEEMB817	IEEMB817
IEE30012I	IEEMB817	IEEMB817	IEEMB817
IOS10001I	IOSVFMTH	IOSVFMTH	IOSVFMTH
IOS10101I	IOSVFMTH	IOSVFMTH	IOSVFMTH
	IOSVFMTR	IOSVFMTR	IOSVFMTR
IOS10104I	IOSVFMTH	IOSVFMTH	IOSVFMTH
IOS10105I	IOSVFMTH	IOSVFMTH	IOSVFMTH
IOS10106I	IOSVFMTH	IOSVFMTH	IOSVFMTH
IOS10107I	IOSVFMTH	IOSVFMTH	IOSVFMTH
IOS10108I	IOSVFMTH	IOSVFMTH	IOSVFMTH
IOS10110I	IOSVFMTH	IOSVFMTH	IOSVFMTH
	IOSVFMTR	IOSVFMTR	IOSVFMTR
	IOSVFMTY	IOSVFMTY	IOSVFMTY
IOS10111I	IOSVFMTH	IOSVFMTH	IOSVFMTH
IOS10112I	IOSVFMTH	IOSVFMTH	IOSVFMTH
IOS10113I	IOSVFMTH	IOSVFMTH	IOSVFMTH
IOS10114I	IOSVFMTH	IOSVFMTH	IOSVFMTH
IOS10201I	IOSVFMTY	IOSVFMTY	IOSVFMTY
IOS10202I	IOSVFMTY	IOSVFMTY	IOSVFMTY
IOS10302I	IOSVFMTR	IOSVFMTR	IOSVFMTR
IOS10303I	IOSVFMTR	IOSVFMTR	IOSVFMTR
IOS10304I	IOSVFMTR	IOSVFMTR	IOSVFMTR
IOS10305I	IOSVFMTR	IOSVFMTR	IOSVFMTR
IRA10101I	IRARMCBS	IRARMCBS	IRARMCBS
IRA10102I	IRARMCBS	IRARMCBS	IRARMCBS
IRA10103I	IRARMCBS	IRARMCBS	IRARMCBS
IRA10104I	IRARMCBS	IRARMCBS	IRARMCBS
IRA10105I	IRARMCBS	IRARMCBS	IRARMCBS

Message ID	Detecting Module	Issuing Module	Containing Module
IRA10106I	IRARMCBS	IRARMCBS	IRARMCBS
IRA10107I	IRARMCBS	IRARMCBS	IRARMCBS
IRA10108I	IRARMCBS	IRARMCBS	IRARMCBS
IRA10109I	IRARMCBS	IRARMCBS	IRARMCBS
IRA10110I	IRARMCBS	IRARMCBS	IRARMCBS
IRA10111I	IRARMCBS	IRARMCBS	IRARMCBS
IRA10112I	IRARMCBS	IRARMCBS	IRARMCBS
IRA10113I	IRARMCBS	IRARMCBS	IRARMCBS
IRA10114I	IRARMCBS	IRARMCBS	IRARMCBS
IRA10201I	IRARMCBV	IRARMCBV	IRARMCBV
IRA10202I	IRARMCBV	IRARMCBV	IRARMCBV
IRA10203I	IRARMCBV	IRARMCBV	IRARMCBV
ISG10001I	ISGDCONT	ISGDCONT	ISGDCONT
	ISGDPDMP	ISGDPDMP	ISGDPDMP
ISG10002I	ISGDCONT	ISGDCONT	ISGDCONT
	ISGDPDMP	ISGDPDMP	ISGDPDMP
ISG10003I	ISGDCONT	ISGDCONT	ISGDCONT
	ISGDPDMP	ISGDPDMP	ISGDPDMP
ISG10004I	ISGDCONT	ISGDCONT	ISGDCONT
	ISGDPDMP	ISGDPDMP	ISGDPDMP
ISG10005I	ISGDPDMP	ISGDPDMP	ISGDPDMP
ISG10006I	ISGDPDMP	ISGDPDMP	ISGDPDMP
ISG10007I	ISGDCONT	ISGDCONT	ISGDCONT
	ISGDPDMP	ISGDPDMP	ISGDPDMP
ISG10008I	ISGDCONT	ISGDCONT	ISGDCONT
	ISGDPDMP	ISGDPDMP	ISGDPDMP
ISG10009I	ISGDCONT	ISGDCONT	ISGDCONT
	ISGDPDMP	ISGDPDMP	ISGDPDMP
ISG10010I	ISGDCONT	ISGDCONT	ISGDCONT
	ISGDPDMP	ISGDPDMP	ISGDPDMP
ISG10011I	ISGDCONT	ISGDCONT	ISGDCONT
	ISGDPDMP	ISGDPDMP	ISGDPDMP
ISG10012I	ISGDCONT	ISGDCONT	ISGDCONT
	ISGDPDMP	ISGDPDMP	ISGDPDMP
ISG10013I	ISGDCONT	ISGDCONT	ISGDCONT
	ISGDPDMP	ISGDPDMP	ISGDPDMP
ISG10014I	ISGDCONT	ISGDCONT	ISGDCONT
	ISGDPDMP	ISGDPDMP	ISGDPDMP
ISG10015I	ISGDCONT	ISGDCONT	ISGDCONT
	ISGDPDMP	ISGDPDMP	ISGDPDMP
ISG10016I	ISGDCONT	ISGDCONT	ISGDCONT
	ISGDPDMP	ISGDPDMP	ISGDPDMP
ISG10017I	ISGDCONT	ISGDCONT	ISGDCONT
	ISGDPDMP	ISGDPDMP	ISGDPDMP
ISG10018I	ISGDCONT	ISGDCONT	ISGDCONT
	ISGDPDMP	ISGDPDMP	ISGDPDMP
ISG10019I	ISGDCONT	ISGDCONT	ISGDCONT
	ISGDPDMP	ISGDPDMP	ISGDPDMP
ISG10020I	ISGDCONT	ISGDCONT	ISGDCONT
	ISGDPDMP	ISGDPDMP	ISGDPDMP
ISG10021I	ISGDCONT	ISGDCONT	ISGDCONT
	ISGDPDMP	ISGDPDMP	ISGDPDMP
ISG10022I	ISGDCONT	ISGDCONT	ISGDCONT
	ISGDPDMP	ISGDPDMP	ISGDPDMP
ISG10023I	ISGDCONT	ISGDCONT	ISGDCONT
	ISGDPDMP	ISGDPDMP	ISGDPDMP
ISG10024I	ISGDCONT	ISGDCONT	ISGDCONT
	ISGDPDMP	ISGDPDMP	ISGDPDMP
ISG10025I	ISGDCONT	ISGDCONT	ISGDCONT
	ISGDPDMP	ISGDPDMP	ISGDPDMP
ISG10026I	ISGDCONT	ISGDCONT	ISGDCONT
	ISGDPDMP	ISGDPDMP	ISGDPDMP
ISG10027I	ISGDCONT	ISGDCONT	ISGDCONT
	ISGDPDMP	ISGDPDMP	ISGDPDMP
ITV10001I	Data in Virtual Component		
ITV10002I	Data in Virtual Component		
ITV10003I	Data in Virtual Component		
ITV10004I	Data in Virtual Component		
ITV10005I	Data in Virtual Component		
ITV10006I	Data in Virtual Component		
ITV10007I	Data in Virtual Component		
ITV10008I	Data in Virtual Component		
ITV10009I	Data in Virtual Component		
ITV10011I	Data in Virtual Component		
ITV10012I	Data in Virtual Component		
ITV10013I	Data in Virtual Component		
ITV10014I	Data in Virtual Component		
ITV10101I	Data in Virtual Component		
ITV10102I	Data in Virtual Component		
ITV10103I	Data in Virtual Component		
ITV10104I	Data in Virtual Component		
ITV10105I	Data in Virtual Component		
ITV10106I	Data in Virtual Component		
ITV10107I	Data in Virtual Component		
ITV10108I	Data in Virtual Component		
ITV10111I	Data in Virtual Component		
ITV10112I	Data in Virtual Component		
ITV10113I	Data in Virtual Component		
ITV10114I	Data in Virtual Component		
ITV10115I	Data in Virtual Component		
ITV10116I	Data in Virtual Component		

MVS/Extended Architecture Message Library: Dump Output Messages

GC28-1336-0

S370-40

Printed in U.S.A.

MVS/Extended Architecture
Message Library:
Dump Output Messages

READER'S
COMMENT
FORM

GC28-1336-0

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. 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.

Note: ***Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.***

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, and date:

_____

_____

_____

_____

What is your occupation? _____

How do you use this publication? _____

Number of latest Newsletter associated with this publication: _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

MVS/Extended Architecture Message Library: Dump Output Messages

GC28-1336-0 S370-40

Reader's Comment Form

Cut or Fold 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
Department D58, Building 921-2
PO Box 390
Poughkeepsie, New York 12602

Fold and tape Please Do Not Staple Fold and tape

Printed in U.S.A.

IBM®

GC28-1336-00