

IBM Data Language/I Disk Operating System/
Virtual Storage (DL/I DOS/VS)



Messages and Codes

Version 1 Release 8

IBM Data Language/I Disk Operating System/
Virtual Storage (DL/I DOS/VS)



Messages and Codes

Version 1 Release 8

Note !

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

Eleventh Edition (December 2002)

This edition applies to Version 1 Release 8 of IBM Data Language/I Disk Operating System/Virtual Storage (DL/I DOS/VS), Program Number 5746-XX1, and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the addresses given below.

A form for readers' comments is provided at the back of this publication. If the form has been removed, address your comments to:

IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany

You may also send your comments by FAX or via the Internet:

Internet: s390id@de.ibm.com
FAX (Germany): 07031-16-3456
FAX (other countries): (+49)+7031-16-3456

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1973, 2002. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	v
Trademarks and Service Marks	v
Preface	vii
Related Publications:	vii
DL/I Message Format	viii
DL/I Message Directory	ix
Chapter 1. DL/I System Messages	1
DLZnnnt (System Execution Messages)	2
DL/I System/Task Abnormal Termination	61
CICS/VS Transaction ABEND Codes	62
CICS/VS Dump Codes	63
Chapter 2. Data Base Description Generation (DBDGEN) Error Messages	65
ACCnnn (ACCESS Macro)	66
DBDnnn (DBD Macro)	69
DGENnnn (DBDGEN Macro)	71
DMANnnn (DATASET Macro)	77
EXTnnn (LCHILD/DBDGEN Macro)	80
FINnnn (FINISH Macro)	80
FLDnnn (FIELD Macro)	80
LCHDnnn (LCHILD Macro)	83
SEGMnnn (SEGM Macro)	85
XDFLDnnn (XDFLD Macro)	90
Chapter 3. Program Specification Block Generation (PSBGEN) Error Messages	93
PCBnnn (PCB Macro)	94
PGENnnn (PSBGEN Macro)	95
SEGnnn (SENSEG Macro)	96
SFLDnnn (SENFLD Macro)	97
VFLDnnn (VIRFLD Macro)	98
Chapter 4. Online Nucleus Generation Error Messages	101
DLZ1nnn (DLZACT Macro)	102
DLZ3nnn (CALLDLI Macro)	106
Chapter 5. LLC/CC Messages and Return Codes	107
LLC/CC Messages	108
LLC0nnt (Execution Program Generation DLZNN Macro)	108
LLC1nnt (Control Data Base Initialization)	108
LLC/CC Return Codes	109
Chapter 6. DL/I Status Codes	111
DL/I CICS/VS Return Codes	111
DL/I PCB Status Codes	113

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785, U.S.A.

Any pointers in this publication to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement. IBM accepts no responsibility for the content or use of non-IBM Web sites specifically mentioned in this publication or accessed through an IBM Web site that is mentioned in this publication.

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

IBM Deutschland Informationssysteme GmbH
Department 0215
Pascal Str. 100
70569 Stuttgart
Germany

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

Trademarks and Service Marks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

CICS
IBM

Preface

This manual applies to the following releases of IBM Data Language/I Disk Operating System/Virtual Storage (DL/I DOS/VS):

- Version 1, Release 8, Modification Level 0 and
- Version 1, Release 7, Modification Level 1 and
- Version 1, Release 7, Modification Level 0 with SPE PTF UL90011.

It provides information on:

- DL/I messages (except for dialog messages which are in the *DL/I DOS/VS Interactive Resource Definition and Utilities*) and
- Status codes

for:

- Operators,
- Application programmers, and
- Data base administrators.

Related Publications:

DL/I VSE Release Guide, SC33-6211-05

DL/I DOS/VS Release Guide, SC33-6211-04

DL/I DOS/VS Data Base Administration, SH24-5011

DL/I DOS/VS Resource Definition and Utilities, SH24-5021

DL/I DOS/VS Interactive Resource Definition and Utilities, SH24-5029.

DL/I DOS/VS Recovery/Restart Guide, SH24-5030.

Other DL/I publications:

DL/I DOS/VS General Information, GH20-1246

DL/I DOS/VS Library Guide and Master Index, GH24-5008

DL/I DOS/VS Application Programming: High Level Programming Interface, SH24-5009

DL/I DOS/VS Application Programming: CALL and RQDLI Interface, SH12-5411

DL/I DOS/VS Guide for New Users, SH24-5001

DL/I DOS/VS Messages and Codes, SH12-5414

DL/I DOS/VS Diagnostic Guide, SH24-5002

DL/I Message Format

The following conventions are used in showing the format of DL/I messages described in this manual:

- Uppercase letters, stand-alone numbers, and punctuation marks are used to show the actual message text. The only exceptions to this convention are brackets `[]`; braces `{}`; and logical or symbols `|`.
- Lowercase letters and words and associated numbers shown as part of a message text are used to indicate variables. Specific values are substituted for the variables when the message is printed.
- Brackets are used to indicate optional message text. Depending on the circumstances, an item or group of items within brackets may or may not appear when the message is printed.
- Braces are used to indicate alternative items. Depending on the circumstances, only one of the stacked items (or items separated by a logical or symbol) enclosed in braces appears when the message is printed.

DL/I Message Directory

The following DL/I message directory is provided to help you locate DL/I messages in this publication. The directory presents the message formats in alphabetical order, identifies the function and/or macro that issues the messages, and shows the page number where the chapter/section describing these messages start.

Message Format	Function and/or Macro	Page Number
ACCnnn	DBD generation - ACCESS macro	Page 66
DBDnnn	DBD generation - DBD macro	Page 69
DGENnnn	DBD generation - DBDGEN macro	Page 71
DLZnnn	DBD, PSB, and ACT generation using IMF and utility job stream generation using IUG	See Note below.
DLZnnnt	DL/I system messages	Page 2
DLZ1nnn	Nucleus generation - DLZACT macro	Page 102
DLZ3nnn	Nucleus generation - CALLDLI macro	Page 106
DMANnnn	DBD generation - DATASET macro	Page 77
EXTnnn	DBD generation - LCHILD/DBDGEN macro	Page 80
FINnnn	DBD generation - FINISH macro	Page 80
FLDnnn	DBD generation - FIELD macro	Page 80
LCHDnnn	DBD generation - LCHILD macro	Page 83
LLCnnnt	LLC/CC error messages	Page 108
PCBnnn	PSB generation - PCB macro	Page 94
PGENnnn	PSB generation - PSBGEN macro	Page 95
SEGnnn	PSB generation - SENSEG macro	Page 96
SEGMnnn	DBD generation - SEGM macro	Page 85
SFLDnnn	PSB generation - SENFLD macro	Page 97
VFLDnnn	PSB generation - VIRFLD macro	Page 98
XDFLDnnn	DBD generation - XDFLD macro	Page 90

Note: The error messages that may be encountered while using the Interactive Macro Facility (IMF) to generate DBDs, PSBs, or ACTs and while using the Interactive Utility Generation (IUG) facility to generate utility job streams are described in Chapter "Error Messages" of the IBM publication *DL/I DOS/VS Interactive Resource Definition and Utilities*.

Chapter 1. DL/I System Messages

This chapter contains messages issued during DL/I batch system execution, CICS/VS-DL/I online system execution, and DL/I system utility execution. The format of these messages is:

DLZnnnt text

where:

DLZ	Indicates a DL/I message.
nnn	Is the message number.
t	If I, indicates an information message. If A, indicates an operator response is required.
text	Is a brief description of the problem.

Messages during online initialization appear at the system operator's console. Messages issued during online application program execution appear at the CICS/VS transient data destination CSMT.

If a DL/I task abnormal termination occurs during online processing, the transaction is terminated with a CICS/VS message that contains a code following the word ABEND. For an explanation of the ABEND code refer to "CICS/VS Transaction ABEND Codes" on page 62.

DLZnnnt (System Execution Messages)

DLZ000I NO TEXT FOR MESSAGE DLZnnnt

Explanation: When trying to write message DLZnnnt, the DL/I message writer module could not find any text for it. There may have been a bad pointer causing the message writer to find a non-existent message, the module containing the messages may not be correct, or a module may be at the wrong level.

User Response: Look up the message indicated by DLZnnnt in this chapter of the manual and, if found, follow the explanation and response information given. If it is not found, save the log and dump and contact the IBM Support Center.

DLZ001I ABNORMAL TERMINATION ENCOUNTERED

Explanation: An error has occurred that caused batch DL/I to terminate abnormally. DL/I ABEND exit has received control and the system log file is closed. This message is followed by DLZ002I unless the UNLD call fails.

User Response: If the error was detected by DL/I, the reason is indicated by a preceding message.

If DLZ001I was not preceded by a DL/I termination message, the error was not detected by DL/I and the termination routine received control from the

system supervisor. The only way to determine the reason for the error, is to analyze the dump to find the DL/I abend save area. The register and pseudo abend save area contents are the same as those described in message DLZ002I.

DLZ002I BATCH DL/I ABNORMAL TERMINATION COMPLETE

Explanation: This message follows DLZ001I if the buffer pool records were successfully purged (written) and the data bases closed. A storage dump is produced.

The following should be noted from the dump:

Register	Contents
Register 2:	Addressable part of SCD (address of SCD plus 96 bytes).
Register 3:	Address of user save area if abnormal end was entered with a pseudo-abend. If entered for a program check or abnormal end, register 3 value is not set.
Register 4:	Contains original 2-byte error code at entry to pseudo-abend.
Register 5:	Address of STXIT AB save area if abend indicator shows that STXIT AB processing has been entered.
Register 6:	Address of the application program. This is also the DL/I high address rounded up to page boundary.
Register 7:	Address of STXIT PC or pseudo-abend save area.
Register 9:	Address of PST.

At label SCDABSAV in the SCD is a pointer to the STXIT PC or pseudo abend save area which contains:

Dec	(Hex)	Length	Description
-32	(-20)	32	Constant: ***** DL/I ABEND SAVE AREA *****
0	(00)	8	Program old PSW (Program check only)
8	(08)	64	Registers 0 to 15
72	(48)	17	Constant: ABEND INDICATORS*
89	(59)	1	Abend reason indicator: X'80' - Entered by STXIT AB' X'40' - Entered by STXIT PC' X'20' - Abend in progress' X'10' - Buffer pool or data base damage' X'08' - Buffer pool unload and close complete' X'02' - Abend in progress'
90	(5A)	1	STXIT AB reason code as returned by the system (low order byte of register 0)
91	(5B)	25	Constant: *PSTFNCTN,RTCDE,OFFST-XXH
116	(74)	1	PSTFNCTN (function code)
117	(75)	1	PSTRTCDE (return code)
118	(76)	2	PSTOFFST (offset)
120	(78)	24	Constant: PSTBLKNM,BYTNM,DATA-FFF-
144	(90)	4	PSTBLKNM (relative block number)
148	(94)	4	PSTBYTNM (RBA or relative record number)
152	(98)	4	PSTDATA (address of requested data)
156	(9C)	24	Constant: PSTSV1 THROUGH PSTSV3---
180	(B4)	72	PSTSV1 (save area)
252	(FC)	72	PSTSV2 (save area)
324	(144)	72	PSTSV3 (save area)

Notes:

1. If the abend indicator shows that STXIT AB processing has been entered, the STXIT AB save area may be located as follows:

- Find the address of the SCD extension (label SCDEXTBA).
- Find the address of the STXIT AB save area (label SCDEABSV).

This address points to the STXIT AB save area which has the following content:

Dec	(Hex)	Length	Description
0	(00)	8	Program old PSW
8	(08)	64	Registers 0-15

2. The load address of the application program is in the SCDDLIUP field of the SCD. This is also the DL/I high address rounded up to page boundary.

User Response: Take appropriate action.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VIS Diagnostic Guide*.

DLZ003I *file name* STOPPED, I/O ERROR

Explanation: An I/O error occurred during an attempt to load an HSAM data base where *file name* is the DTF file name.

User Response: Reexecute the application program.

DLZ004I ERROR ACCESSING FILE *file name*, VSAM CODE=*xyy*

Explanation: An I/O error occurred during an attempt to read or write to the file *file name*.

For an explanation of the VSAM return code *xx* and the VSAM error code *yy* refer to *VSE/SP Messages and Codes*, Section "VSE/VSAM Return and Error Codes."

This message can also occur on an I/O error either when accessing a data base data set or when writing to a disk log file.

User Response: Remove the error condition, perform data base recovery, and resubmit the job.

DLZ005I *file name* STOPPED, OUT OF SPACE, VSAM CODE=*xyy*

Explanation: An attempt was made to add a record to the file *file name* and there was no more EXTENT allocation.

For an explanation of the VSAM return code *xx* and the VSAM error code *yy* refer to *VSE/SP Messages and Codes*, Section "VSE/VSAM Return and Error Codes."

User Response: Execute the data base data set recovery utility.

DLZ006I UNABLE TO ATTACH DL/I LOGGER MODULE

Explanation: During DL/I initialization, the DL/I asynchronous logger was not successfully attached. No logging will be performed.

User Response: If logging is desired, increase the virtual address area for the partition prior to reinitializing DL/I online.

DLZ007I ERROR DURING CREATION OF WORKFILE, RETURN CODE=*rc*

Explanation: A possible error was detected by DLZDSEH0 while attempting to create WORKFIL output. Refer to message number DLZ952I for the meaning of *rc*.

User Response: If *rc* is 04 and no DL/I abend message occurs, the data base load continues. This happens when logical parent segments are loaded and SYS013 (WORKFIL) has been assigned IGN. In all other cases, the error must be corrected (see DLZ952I message description) and the job reexecuted.

DLZ009I SUBPOOL COUNT CONFLICTS WITH NUMBER OF HDBFR STMTS, SET BY DL/I

Explanation: The number of buffer subpools (subpool count) specified in the parameter statement was inconsistent with the number of HDBFR statements. One of the following occurred:

- The subpool count is smaller than the number of HDBFR statements. The number of subpools is increased by DL/I to the number of HDBFR statements. If necessary, an additional subpool is allocated for unassigned DBDs, including INDEX and HISAM DBDs with load/insert sensitivity.
- The subpool count is equal to the number of HDBFR statements. However, an additional subpool is allocated by DL/I for unassigned DBDs, including INDEX and HISAM DBDs with load/insert sensitivity.
- The subpool count is greater than the number of HDBFR statements plus the number of subpools necessary for unassigned DBDs, including INDEX and HISAM DBDs with load/insert sensitivity. If necessary, a remaining subpool will be assigned as a delete work area.

See *DL/I DOS/VS Resource Definition and Utilities* for details on the HDBFR parameter. See *DL/I DOS/VS Data Base Administration* for details on buffer subpool assignment.

User Response: None.

DLZ010A ENTER DL/I PARAMETER INFORMATION

Explanation: One of the following occurred:

- Bit 0 of the UPSI byte was 1 indicating the parameter information should be entered via SYSLOG.
- Bit 0 of the UPSI byte was 0 indicating the input should come from SYSIPT, but SYSIPT reached end-of-file.

User Response: Enter the correct DL/I parameter information via SYSLOG. Refer to the *DL/I DOS/VS Resource Definition and Utilities* for the correct format.

If Bit 0 of the UPSI byte was 0, this message is preceded by message DLZ014A.

DLZ011I REQUIRED NUCLEUS MODULE MISSING, NAME = *modname*

Explanation: One of the following occurred:

- While building the DL/I nucleus load list, module *modname* was not found in any of the libraries defined in the search chain.
- DLZEIPB0 attempted to load DLZEIPB1 but the phase was not found in any of the libraries defined in the search chain.

The program is canceled.

User Response: Catalog the module in a library and ensure that this library is defined in the search chain. Reexecute the application program.

DLZ012I APPLICATION PROGRAM OR DL/I CONTROL BLOCK *name* NOT FOUND

Explanation: While building a load list entry, the phase *name* was not found in any of the libraries defined in the search chain.

User Response: Catalog the named module in a library and ensure that this library is defined in the search chain. Reexecute the application program.

DLZ013I PSB(S) AND DMB DO NOT MATCH, DBD *dbdname* CANNOT BE USED

Explanation: The data base described by Data Base Description (DBD) *dbdname* cannot be initialized. Its Data Base Management Block (DMB) does not match the DMB used when the PSB(s) was built. In particular, the DMB for *dbdname* is linked (by secondary index or logical relationships) to a data base to which the PSB(s) is not linked. Either the DMB was changed since the PSB was generated or the DMB was not updated during the ACBGEN process. A PSB scheduling error will be returned whenever a program attempts to schedule a PSB which refers to this data base.

User Response: If message DLZ040A has not been stopped by a previous RUN response, it is issued after this message.

Respond to DLZ040A as follows:

- To continue with DL/I and CICS/VS initialization, enter either:
 - GO This data base will be disabled.
 - RUN This data base will be disabled. (DL/I will stop issuing DLZ040A messages.)
- To stop DL/I and CICS/VS initialization, enter either:
 - CANCEL To end DL/I and CICS/VS initialization.
 - DUMP To end DL/I and CICS/VS initialization and DUMP the contents of storage at this point.

To correct this problem, rebuild all PSBs which refer to this DBD using the ACBGEN procedure with DMB=YES specified on the BUILD statement.

Rerun the CICS/VS and DL/I initialization job.

DLZ014A PARAMETER STMT MISSING, REPLY CANCEL OR EOB TO READ FROM SYSLOG

Explanation: Bit 0 of the UPSI byte indicated the DL/I parameter information was to be entered via SYSIPT and an end-of-file was reached.

User Response:

- Reply CANCEL to end the job.
- Enter null line to read the required parameter statement from SYSLOG.

DLZ015I PARAMETER STATEMENT DATA INVALID

Explanation: A DL/I parameter statement contains one of the following error conditions:

- Data did not start in column 1.
- A field length was invalid.
- A required field was omitted.
- An expected continuation statement was not found.
- A continuation statement was specified for SYSLOG input.
- The LOG parameter was specified a second time.
- The TRACE parameter was specified a second time.
- ULR used without DLZURGU0.
- PLU used without DLZURGU0.
- PLU used and the specified PSB references a logical DBD or an INDEX DBD.

The parameter or subparameter causing the error, plus all parameter information following it that was already read, is listed before the message appears.

User Response: Correct the error and reexecute the job.

DLZ016I *file name* MODCB ERROR, VSAM CODE = *xx*

Explanation: An error occurred during execution of the VSAM MODCB macro for *file name*.

For an explanation of the VSAM error code *xx* refer to *VSE/SP Messages and Codes*, Section "VSE/VSAM Return and Error Codes."

User Response: Take the appropriate VSAM action for the error code and reexecute the application program.

DLZ017I PSB VERSION INVALID, DL/I TERMINATED

Explanation: While loading the requested PSB, initialization detected that the block was created prior to Version 1.1.

User Response: Rebuild the PSB using the application control block utility DLZUACB0 provided with the current level of DL/I.

DLZ018I DMB *dmbname* VERSION INVALID, DL/I TERMINATED

Explanation: While loading the named DMB, initialization detected that the block was created prior to Version 1.1.

User Response: Rebuild the block using the application control block utility DLZUACB0 provided with the current level of DL/I.

DLZ019I **BUFFER NUMBER IN {HDBFR|HSBFR} STATEMENT TOO {LOW|HIGH}. SET TO {MINIMUM|MAXIMUM}**

Explanation: One of the buffer number specifications did not fulfill the minimum requirements, or the specification was too high. DL/I resets the values to the minimum (if it was too low) or the maximum. See *DL/I DOS/VS Resource Definition and Utilities* for information on the requirements.

User Response: None.

DLZ020I *file name* **OPEN ERROR, VSAM CODE = xx**

Explanation: An error occurred during an attempt to open a DL/I file *file name*.

For an explanation of the VSAM error code *xx* refer to *VSE/SP Messages and Codes*, Section "VSE/VSAM Return and Error Codes."

User Response: Take the appropriate VSAM action for the error code.

If *file name* indicates the log file, the error condition should be removed. Depending on the VSAM error code, data base recovery can be done and the job resubmitted.

If *file name* does not indicate the log file, it is possible that either the data base data set recovery utility or data base backout utility should be executed if the data set was not previously closed. The message also occurs when loading a data base with secondary indexes and the indexes are to be created by one of the logical relationship utilities (there are no DLBL and EXTENT cards for the secondary indexes). The VSAM error code is X'80'. Loading continues.

DLZ021I *file name* **CLOSE ERROR, VSAM CODE = xx**

Explanation: An error occurred during an attempt to close the DL/I file *file name*.

For an explanation of the VSAM error code *xx* refer to *VSE/SP Messages and Codes*, Section "VSE/VSAM Return and Error Codes."

User Response: If *file name* indicates a data base data set, take the appropriate VSAM action for the error code and reexecute the application program. If *file name* indicates the log file, the following steps should be taken:

- Remove the error condition.
- Execute AMS VERIFY against the log file.
- Perform data base recovery.
- Resubmit the job.

DLZ022I *file name* **OPEN ERROR WRITING DL/I CONTROL RECORD**

Explanation: An error occurred during an attempt to write the first VSAM control interval containing the DL/I control record on *file name*.

User Response: Execute the VSAM access method services utility command VERIFY for the file and then reexecute the application program.

DLZ023I *file name* **OPEN ERROR INVALID PROCOPT**

Explanation: One of the following occurred:

- A PROCOPT≠L was specified for the PCB accessing the file and the file was empty.
- A PROCOPT≠L was specified for the PCB accessing the file and the parameter statements start with PLS/PLC (partial data base load).
- A PROCOPT=L was specified for the PCB accessing the file and the file was not empty.
- The parameter statement contains PLS/PLC (partial data base load) and the data base is completely loaded with a load step using either 'PLC' or 'DLI'.

User Response: You should:

- Verify that the PROCOPT is correctly specified and that the correct PCB is used.
- Prior to loading a data base, execute the VSAM access method services utility commands DELETE and DEFINE.

DLZ024I *file name* **SHOWCB ERROR, VSAM CODE = xx**

Explanation: An error occurred during execution of the VSAM SHOWCB macro for *file name*.

For an explanation of the VSAM error code *xx* refer to *VSE/SP Messages and Codes*, Section "VSE/VSAM Return and Error Codes."

User Response: Take the appropriate VSAM action for the error code and reexecute the application program.

DLZ025I *file name* **CONTROL INTERVAL SIZE MISMATCH**

Explanation: The control interval size specification in the VSAM catalog is not the same value as the corresponding value in the DMB for *file name*.

User Response: You should perform one or more of the following prior to reexecuting the application program:

- Execute a new DBD generation.
- Delete the DMB and execute the ACB creation and maintenance utility.
- Execute the VSAM access method services utility DEFINE command.

DLZ026I **GETVIS FAILED IN TRACE MODULE, TRACING HAS BEEN DEACTIVATED**

Explanation: The trace module could not get space needed to initialize. The DL/I job continues to run without the trace facility.

User Response: If tracing is wanted, resubmit the job with more GETVIS space available in the partition.

DLZ027I *file name* **KEY LENGTH MISMATCH**

Explanation: The key length specification in the VSAM catalog is not the same value as the corresponding value in the DMB for *file name*.

User Response: You should perform one or more of the following prior to reexecuting the application program:

- Execute a new DBD generation.
- Delete the DMB and execute the ACB creation and maintenance utility.
- Execute the VSAM access method services utility DELETE and DEFINE commands.

DLZ028I *file name* **RELATIVE KEY POSITION MISMATCH**

Explanation: The relative key position specification in the VSAM catalog is not the same value as the corresponding value in the DMB for *file name*.

User Response: You should perform one or more of the following prior to reexecuting the application program:

- Execute a new DBD generation.
- Delete the DMB and execute the ACB creation and maintenance utility.
- Execute the VSAM access method services utility DELETE and DEFINE commands.

DLZ030I **STORAGE LAYOUT CONTROL TABLE IS INVALID, WILL NOT BE USED**

Explanation: The module specified in the SLC parameter of the DLZACT TYPE=CONFIG statement does not match the format of a storage layout control table.

User Response: If message DLZ040A has not been stopped by a previous RUN response, it is issued after this message. Respond to it as follows:

- To continue with DL/I and CICS/VS initialization, enter either:
GO DL/I will select the storage layout.
RUN DL/I will select the storage layout.
(DL/I will stop issuing DLZ040A messages.)
- To stop DL/I and CICS/VS initialization, enter either:
CANCEL To end DL/I and CICS/VS initialization.
DUMP To end DL/I and CICS/VS initialization and DUMP the contents of storage at this point.

To correct this problem, rebuild the SLC table using the DLZSLC macro and the SLC generation procedure.

Then rerun the CICS/VS and DL/I initialization job.

DLZ031I **CICS JOURNAL NOT ACTIVE BUT REQUIRED FOR DL/I PROGRAM ISOLATION**

Explanation: Program isolation requires CICS/VS dynamic transaction backout to be active which in turn requires CICS/VS journaling. This message is issued if CICS/VS journaling was suppressed (by UPSI bits 6 and 7) in the CICS/VS initialization job.

User Response: Set UPSI bits 6 and 7 in the CICS/VS initialization job to 0. Rerun the job.

DLZ032A **THE SYSTEM LOG JCT ENTRY IS MISSING**

Explanation: Logging on the CICS/VS system journal was specified, but no JCT entry was found for the system log.

User Response: If logging is not desired, change the UPSI byte to indicate no logging. If logging is required on the CICS/VS system log, ensure that the JCT entry specified is cataloged to the system and restart the job.

DLZ033I **INTERSYSTEM COMMUNICATION FAILURE, REASON IS xxxxxxxx**

Explanation: An abnormal condition was encountered during an attempt to process a DL/I call on a remote system. xxxxxxxx defines the specific condition as follows:

xxxxxxx	Explanation
ISC MOD	CICS/VS intercommunication support not generated.
INVPARM	ISC detected invalid call parameter list. Task terminated.
NO LINK	Link to remote system does not exist.
BADLINK	Remote system link out of service.
DLIDOWN	DL/I on remote system is not active.
NO XECB	MPS cross-partition communication XECB not found. Task terminated.
CHKPERR	Remote PSB failed to reschedule after CHKP taken. Task terminated.
INT ERR	Internal ISC program error. Task terminated.
User Response: If xxxxxxxx is:	
ISC MOD	Consult the <i>CICS/DOS/VS Installation and Operations Guide</i> to determine how to generate CICS/VS with intersystem communication support. Resubmit the job after bringing up CICS/VS with ISC support.
INVPARM	Data base call was attempted with insufficient arguments. Check call format and provide

additional arguments as necessary.

- NO LINK Check that the remote system name was correctly specified in DLZACT TYPE=RPSB statement for PSB requested to be scheduled. Also, check that correct remote system name is defined in the CICS/VS Terminal Control Table (TCT).
- BADLINK Use CICS/VS master terminal commands to bring the link to the remote system back into service.
- DLIDOWN Use CICS/VS initialization to bring up DL/I in remote system.
- NO XECB Check that the Multiple Partition Support (MPS) batch job has not been canceled. Check that no other batch job is using XECB names reserved for DL/I MPS.
- CHKPERR Restart failing task from the last CHKP taken.
- INT ERR Save the log and dump and contact the IBM Support Center.

Depending on the CI size, the following restrictions are valid for specific CI sizes:

CI size in bytes	Number of allowed secondary indexes
512	53
1024	117
1536	181
2048	245
>2048	254

User Response: Load the data base with one step (DLI in parameter statement) or increase the CI size.

DLZ034I ERROR BUILDING SECONDARY INDEX *dbdname* DURING PARTIAL DATA BASE LOAD

Explanation: During the steps of partial data base load, you have changed the method to build the named INDEX data base or you have added or deleted a secondary index definition for the named INDEX data base.

User Response: Choose the same method to build the named INDEX data base, or add the deleted or remove the added secondary index definition, and re-do DBDGEN and ACBGEN before re-running this job.

DLZ035I INVALID DATA BASE ORGANIZATION FOR PARTIAL DATA BASE LOAD

Explanation: PLS/PLC used in the parameter statement and the specified PSB references a DBD with HSAM/SHSAM or HISAM/SHISAM organization.

User Response: Load the data base with one step (DLI in parameter statement).

DLZ036I TOO MANY SECONDARY INDEXES DEFINED FOR PARTIAL DATA BASE LOAD

Explanation: During the several partial load steps DL/I has to recognize the status of a secondary index. Therefore an additional eight bytes of control information are used for each defined secondary index in the DL/I control record (1.CI of the data base ESDS).

DLZ037I DL/I STATUS 'xx' RETURNED, STMT 'yyyyyyyy', PROGRAM TERMINATED

Explanation: A DL/I status code (xx) indicating an abnormal condition was returned to the application program and the program was abnormally terminated. See Chapter 6, "DL/I Status Codes" on page 111 for an explanation of the status code. The yyyyyyyy is the statement number of the command that generated the status code, if available. For PL/I, the statement number will be taken from columns 73 to 80 of the listing produced by the CICS/VS translator. For COBOL, the statement number will be six characters taken from columns 1 to 6.

User Response: Correct the error and rerun application program.

DLZ038I ERROR OCCURRED EXECUTING {GETVIS|FREEVIS} MACRO, RETURN CODE = rc

Explanation: A DL/I module encountered an error while executing a GETVIS or FREEVIS request. The program terminated. The GETVIS or FREEVIS return code rc is given in decimal format.

For an explanation of the GETVIS or FREEVIS return code rc refer to *VSE/SP Messages and Codes*, Section "VSE/Advanced Functions Return Codes."

User Response: Correct the error and rerun the program.

DLZ039I FIELD EXIT ROUTINE xxxxxxxx NOT FOUND

Explanation: During DL/I online initialization, field exit routine xxxxxxxx could not be found in any of the libraries defined in the search chain. This message is followed by DLZ056I, which identifies the PSB containing the field exit routine xxxxxxxx, in the RTNAME= parameter of a SENFLD or VIRFLD statement.

User Response: The PSB named in DLZ056I will not be usable until the DL/I online system is terminated, the error is corrected, and DL/I is reinitialized.

DLZ040A ENTER CANCEL, DUMP, GO TO CONTINUE, OR RUN TO END DLZ040A MESSAGES

Explanation: This message is written as the second message of a pair.

User Response: Take the appropriate action based on the previous error message printed:

- To continue with the program enter either:

GO	Causes the program to continue.
RUN	Causes the program to continue. No further DLZ040A messages will be written. The first message of the pair will still be written.
- To stop the program, enter either:

CANCEL	Causes the CANCEL macro to be invoked for the partition.
DUMP	Causes the DUMP macro to be issued for the partition.

DLZ041I INVALID RESPONSE TO MESSAGE DLZ040A

Explanation: The response to message DLZ040A was not GO, RUN, CANCEL, or DUMP.

User Response: Reply with GO, RUN, CANCEL, or DUMP as explained in message DLZ040A and in the message preceding DLZ040A.

DLZ042I HSAM DATA BASE NOT ALLOWED ONLINE, PSB *psbname* CANNOT BE USED

Explanation: HSAM data bases are not supported by DL/I in an online environment. PSB *psbname* refers to an HSAM data base and, therefore, cannot be initialized.

User Response: If message DLZ040A has not been stopped by a previous RUN response, it is issued after this message.

Respond to DLZ040A as follows:

- To continue with DL/I and CICS/VS initialization, enter either:

GO	PSB <i>psbname</i> will be disabled.
RUN	PSB <i>psbname</i> will be disabled. (DL/I will stop issuing DLZ040A messages.)
- To stop DL/I and CICS/VS initialization, enter either:

CANCEL	To end DL/I and CICS/VS initialization.
DUMP	To end DL/I and CICS/VS initialization and DUMP the contents of storage at this point.

To correct this problem, remove PSB *psbname* from your DL/I online nucleus definition by updating the DLZACT statements used to build this nucleus. This is done by removing *psbname* from all DLZACT TYPE=PROGRAM statements. If the program cannot be run without this PSB, the TYPE=PROGRAM statement must be removed.

Rebuild the DL/I nucleus (ACT) using the ACTGEN procedure.

Rerun the CICS/VS and DL/I initialization job.

DLZ043I LOAD OPTION NOT ALLOWED ONLINE, PSB *psbname* CANNOT BE USED

Explanation: DL/I data bases must be loaded in a batch environment. PSB *psbname* was built with a LOAD processing option (PROCOPT=L or LS on the PCB statement). Therefore, it cannot be initialized in an online environment.

User Response: If message DLZ040A has not been stopped by a previous RUN response, it is issued after this message.

Respond to DLZ040A as follows:

- To continue with DL/I and CICS/VS initialization, enter either:

GO	PSB <i>psbname</i> will be disabled.
RUN	PSB <i>psbname</i> will be disabled. (DL/I will stop issuing DLZ040A messages.)
- To stop DL/I and CICS/VS initialization, enter either:

CANCEL	To end DL/I and CICS/VS initialization.
DUMP	To end DL/I and CICS/VS initialization and DUMP the contents of storage at this point.

To correct this problem, load the data base referred to by PSB *psbname* in the batch environment, if it has not already been loaded.

Rebuild this PSB using the PSBGEN procedure with PROCOPT other than L or LS specified on the PCB statement.

Then rerun the CICS/VS and DL/I initialization job.

DLZ044I PSB *psbname* NOT FOUND IN LIBRARY

Explanation: PSBs defined in the DL/I online nucleus are loaded from one of the libraries defined in the search chain during DL/I online initialization. PSB *psbname* was not found in any of the libraries defined in the search chain and, therefore, was not loaded.

User Response: Respond to DLZ040A as follows:

- To continue with DL/I and CICS/VS initialization, enter either:

GO	PSB <i>psbname</i> will be disabled.
RUN	PSB <i>psbname</i> will be disabled. (DL/I will stop issuing DLZ040A messages.)
- To stop DL/I and CICS/VS initialization, enter either:

CANCEL	To end DL/I and CICS/VS initialization.
DUMP	To end DL/I and CICS/VS initialization and DUMP the contents of storage at this point.

To correct this problem, check that:

- The correct PSB name was used when defining the DL/I online nucleus (ACTGEN procedure).

- The correct PSB name was used during the PSB definition step of the PSBGEN procedure.
- The correct library was assigned during the PSB generation step (ACBGEN procedure) of the PSBGEN procedure and that the LIBDEF sequence is correct.
- PSB *psbname* was not removed from any of the libraries defined in the search chain.

Rebuild the PSB or online nucleus.

Rerun the CICS/VS and DL/I initialization job.

DLZ045I CICS/VS FILE CONTROL TABLE NOT LOCATED

Explanation: A required CICS/VS table, the file control table (FCT), could not be found during DL/I online initialization. In particular, the field CSAFACTBA in the CICS/VS Common System Area (CSA) control block does not contain an address of an FCT. DL/I cannot be initialized.

User Response: If message DLZ040A has not been stopped by a previous RUN response, it is issued after this message.

Respond to DLZ040A as follows:

- To continue with CICS/VS initialization only, enter either:
GO or RUN DL/I will not be initialized. A dummy DL/I nucleus will be used which returns control to the application program when DL/I is called without performing any DL/I function.
- To stop DL/I and CICS/VS initialization, enter either:
CANCEL To end DL/I and CICS/VS initialization.
DUMP To end DL/I and CICS/VS initialization and DUMP the contents of storage at this point.

To correct this problem, rerun the CICS/VS and DL/I initialization job with a CICS/VS file control table specified. If the problem persists, contact the IBM Support Center.

DLZ046I DBD *dbdname* NOT DEFINED IN CICS/VS FILE CONTROL TABLE

Explanation: The data base described by Data Base Description (DBD) *dbdname* cannot be initialized because its name is not defined in the CICS/VS file control table (FCT). A PSB scheduling error will be returned whenever a program attempts to schedule a PSB which refers to this data base.

User Response: If message DLZ040A has not been stopped by a previous RUN response, it is issued after this message.

Respond to DLZ040A as follows:

- To continue with DL/I and CICS/VS initialization, enter either:
GO This data base will be disabled.

RUN This data base will be disabled. (DL/I will stop issuing DLZ040A messages.)

- To stop DL/I and CICS/VS initialization, enter either:

CANCEL To end DL/I and CICS/VS initialization.

DUMP To end DL/I and CICS/VS initialization and DUMP the contents of storage at this point.

To correct this problem, add this data base to the CICS/VS file control table. The DATASET name in the CICS/VS file control table must be the same name specified in the NAME operand of the DBDGEN procedure for this data base.

Rerun the CICS/VS and DL/I initialization job.

DLZ047I DBD *dbdname* NOT FOUND IN LIBRARY

Explanation: Data Base Description (DBD) *dbdname* is needed by a PSB(s) defined in the DL/I online nucleus. DBD *dbdname* was not found in any of the libraries defined in the search chain and, therefore, was not loaded.

User Response: If message DLZ040A has not been stopped by a previous RUN response, it is issued after this message.

Respond to DLZ040A as follows:

- To continue with DL/I and CICS/VS initialization, enter either:
GO This data base will be disabled.
RUN This data base will be disabled. (DL/I will stop issuing DLZ040A messages.)
- To stop DL/I and CICS/VS initialization, enter either:
CANCEL To end DL/I and CICS/VS initialization.
DUMP To end DL/I and CICS/VS initialization and DUMP the contents of storage at this point.

To correct this problem, catalog and link-edit the DBD as a phase in the correct library using the DBDGEN procedure.

Rerun the CICS/VS and DL/I initialization job.

DLZ048I RANDOMIZING MODULE *randname* FOR DBD *dbdname* NOT FOUND

Explanation: Randomizing module *randname* is required by the HDAM Data Base Description (DBD) *dbdname*. Module *randname* was not found in any of the libraries defined in the search chain and, therefore, was not loaded.

User Response: If message DLZ040A has not been stopped by a previous RUN response, it is issued after this message.

Respond to DLZ040A as follows:

- To continue with DL/I and CICS/VS initialization, enter either:
GO This data base will be disabled.

RUN This data base will be disabled.
(DL/I will stop issuing DLZ040A messages.)

- To stop DL/I and CICS/VS initialization, enter either:
CANCEL To end DL/I and CICS/VS initialization.
- DUMP To end DL/I and CICS/VS initialization and DUMP the contents of storage at this point.

To correct this problem, catalog and link-edit the randomizing module as a phase in a library using the system link-edit procedure and ensure that this library is defined in the search chain.

Rerun the CICS/VS and DL/I initialization job.

DLZ049I NO VALID DMBS FOUND

Explanation: No DL/I data bases were successfully initialized during DL/I online initialization. Therefore, DL/I will not be initialized.

User Response: If message DLZ040A has not been stopped by a previous RUN response, it is issued after this message.

Respond to DLZ040A as follows:

- To continue with CICS/VS initialization only, enter either:
GO or RUN DL/I will be initialized. A dummy DL/I nucleus will be used which returns control to the application program when DL/I is called without performing any DL/I function.
- To stop DL/I and CICS/VS initialization, enter either:
CANCEL To end DL/I and CICS/VS initialization.
- DUMP To end DL/I and CICS/VS initialization and DUMP the contents of storage at this point.

Then correct the problems described by DL/I messages issued during this attempt to initialize CICS/VS and DL/I.

Rerun the CICS/VS and DL/I initialization job.

DLZ052I DL/I INITIALIZATION FAILED, DL/I NOT ACTIVE

Explanation: DL/I online initialization encountered an error and DL/I will continue with a dummy nucleus without executing any DL/I calls.

User Response: This is a summary message: refer to the explanation of the previous error message printed for specific action.

DLZ053I DL/I INITIALIZATION COMPLETE

Explanation: DL/I online initialization was successful and no errors were encountered.

User Response: None required.

DLZ054I DL/I INITIALIZATION ERROR(S) DETECTED

Explanation: DL/I online initialization was successful, but one or more errors were detected.

User Response: Refer to explanation(s) of the previous error message(s) printed.

DLZ055I DL/I ACTION MODULE *modname* NOT FOUND IN LIBRARY

Explanation: DL/I action module *modname* is required by DL/I. It could not be found in any of the libraries defined in the search chain and, therefore, was not loaded.

User Response: If message DLZ040A has not been stopped by a previous RUN response, it is issued after this message.

Respond to DLZ040A as follows:

- To continue with CICS/VS initialization only, enter either:
GO or RUN DL/I will not be initialized. A dummy DL/I nucleus will be used which returns control to the application program when DL/I is called without performing any DL/I function.
- To stop DL/I and CICS/VS initialization only, enter either:
CANCEL To end DL/I and CICS/VS initialization.
- DUMP To end DL/I and CICS/VS initialization and DUMP the contents of storage at this point.

To correct this problem, check that:

- The correct procedure was used to install DL/I.
- The DL/I action module was not removed.
- The LIBDEF search chain includes the DL/I library.

Catalog and link-edit the DL/I action module as a phase in the correct library using the DL/I installation procedure or correct the LIBDEF search chain.

Rerun the CICS/VS and DL/I initialization job.

DLZ056I FIELD EXIT ROUTINE MISSING, PSB *psbname* CANNOT BE USED

Explanation: PSB *psbname* could not be initialized because a required field exit routine could not be found (see message DLZ039I which precedes this message).

User Response: If message DLZ040A has not been stopped by a previous RUN response, it is issued after this message.

Respond to DLZ040A as follows:

- To continue with DL/I and CICS/VS initialization, enter either:
GO This data base will be disabled.

RUN This data base will be disabled.
(DL/I will stop issuing DLZ040A messages.)

- To stop DL/I and CICS/VS initialization, enter either:
CANCEL To end DL/I and CICS/VS initialization.
- DUMP To end DL/I and CICS/VS initialization and DUMP the contents of storage at this point.

To correct this problem, check that:

- The correct field exit name was used in the RTNAME= parameter of the SENFLD or VIRFLD statement when building this PSB. If this caused the problem, rebuild the PSB using the PSBGEN procedure.
- The correct library was assigned when the field exit routine was cataloged and link-edited. Ensure that this library is defined in the search chain.
- The field exit routine was not removed from the library.

Rebuild the PSB if necessary or catalog and link-edit the field exit routine in the correct library using the system link-edit procedure.

Rerun the CICS/VS and DL/I initialization job.

DLZ0571 OPEN ERROR, DBD *dbdname* CANNOT BE USED

Explanation: DBD *dbdname* could not be initialized because of an open error on a data set required by this DBD. The open error is described by a preceding open error message (see messages DLZ0201 to DLZ0281). A PSB scheduling error will be returned whenever a program attempts to schedule a PSB which refers to this data base.

Note: The only exception is if the open error was reported for an indirectly-built secondary index DBD belonging to a data base within partial data base load. In that case you can ignore this message, if you do not want to use a PCB with an alternate processing sequence using this index.

User Response: If message DLZ040A has not been stopped by a previous RUN response, it is issued after this message.

Respond to DLZ040A as follows:

- To continue with DL/I and CICS/VS initialization, enter either:
GO This data base will be disabled.
RUN This data base will be disabled.
(DL/I will stop issuing DLZ040A messages.)
- To stop DL/I and CICS/VS initialization, enter either:
CANCEL To end DL/I and CICS/VS initialization.
- DUMP To end DL/I and CICS/VS initialization and DUMP the contents of storage at this point.

Refer to the preceding open error message for help in correcting the problem.

Rerun the CICS/VS and DL/I initialization job.

DLZ0581 INSUFFICIENT STORAGE TO START DL/I

Explanation: The virtual storage area is too small to hold the DL/I modules or buffer pool.

User Response:

- Batch:

This message is followed by an immediate abnormal termination and dump.

Check the following:

- The HDBFR parameter in the DL/I parameter statement can be used to decrease the number of buffers required. It is possible for the parameter not to be read because of a missing continuation character in column 72. If this caused the problem, correct the DL/I parameter and rerun the job.
- The eligible DL/I action modules have not been loaded in the shared virtual area (SVA). (For a complete list of SVA-eligible DL/I phases refer to the *DL/I DOS/VS 1.8 Release Guide* supplied with the program materials.) If this caused the problem, install the DL/I action modules in the SVA and rerun the job.

Increase the SIZE parameter in the EXEC statement and resubmit the job or increase the partition size and the SIZE parameter in the EXEC statement and resubmit the job.

- Online:

If message DLZ040A has not been stopped by a previous RUN response, it is issued after this message.

Respond to DLZ040A as follows:

- To continue with CICS/VS initialization only, enter either:
GO or RUN DL/I will not be initialized. A dummy DL/I nucleus will be used which returns control to the application program when DL/I is called without performing any DL/I function.
- To stop DL/I and CICS/VS initialization, enter either:
CANCEL To end DL/I and CICS/VS initialization.
- DUMP To end DL/I and CICS/VS initialization and DUMP the contents of storage at this point.

To correct this problem, check that:

- The HDBFR parameter in the DLZACT TYPE=BUFFER statement can be used to decrease the number of buffers required. It is possible for the parameter not to have

an effect because of an error in the DLZACT statements used to build this nucleus. If this caused the problem, correct the DLZACT TYPE=BUFFER statements and rebuild the DL/I nucleus (ACT) using the ACTGEN procedure.

- The eligible DL/I action modules have not been loaded in the shared virtual area (SVA). (For a complete list of SVA-eligible DL/I phases refer to the *DL/I DOS/VS 1.8 Release Guide* supplied with the program materials.) If this caused the problem, install the DL/I action modules in the SVA.

Increase the SIZE parameter in the EXEC statement and rerun the CICS/VS and DL/I initialization job or increase the partition size and the SIZE parameter and resubmit the job.

DLZ059A ENTER CANCEL TO END, OR GO TO CONTINUE

Explanation: This message allows you to decide whether the preceding message was serious enough to cause the job to be canceled. There was some condition which could cause a problem, but may allow the job to continue and run successfully.

User Response: Enter CANCEL if the error described in the preceding message was significant enough so that DL/I should not continue. Enter GO if the job should continue.

DLZ060I BUFFER POOL ALLOCATION MISSING OR INVALID

Explanation: During DL/I initialization the allocation for buffer pool count was zero or invalid. Message DLZ061A follows.

User Response: Respond to message DLZ061A as specified in that message.

DLZ061A ENTER POOL COUNT, GO, CANCEL, OR DUMP

Explanation: This message follows message DLZ060I.

User Response: Respond with one of the following entries:

<i>pool-count</i>	If you enter the number of subpools needed ($1 \leq \text{pool-count} \leq 255$), DL/I continues normal online initialization.
GO	Loads a dummy DL/I nucleus and continues online initialization. Use GO only when you do not want to define a DL/I environment.
CANCEL	Cancels DL/I online initialization. You must reassemble an ACT with a non-zero buffer pool count and reinitialize CICS/VS.
DUMP	Produces a dump and terminates DL/I online initialization. You must reassemble an ACT with a non-zero buffer pool count and reinitialize CICS/VS.

DLZ062I DL/I HAS BEEN ABNORMALLY TERMINATED

Explanation: A condition has occurred that makes it impossible for DL/I to continue. The active DL/I task has been abnormally terminated with code D062, and no further scheduling of DL/I services will be performed. Field TCAFCTR (TCAFTRC in ANS COBOL) or UIBFCTR (if UIB is used) contains X'08FF'. Application programs should test this field after requesting DL/I services, and further processing should be based on the result. HLPI programs will abend with a DHTJ abend code.

User Response: Previous error messages printed should be acted upon before reinitializing DL/I online.

DLZ063I DL/I NORMAL TERMINATION ENTERED

Explanation: CICS/VS is being terminated and the DL/I terminator has been called by DLZSTP00.

User Response: None required.

DLZ064I DL/I CANNOT BE STARTED, DL/I NUCLEUS

xxxxxxx

Explanation: If xxxxxxx is:

NOT FOUND

The DL/I online nucleus module, DLZNUCxx, was not loaded from any of the libraries defined in the search chain during DL/I online initialization. In particular, the field CSADLI in the CICS/VS Common System Area (CSA) optional features list control block does not contain an address of the DL/I nucleus. DL/I cannot be initialized.

HAS WRONG ENTRY POINT

The ENTRY statement identifying DLZNUC as the entry point was not provided to the linkage editor when generating the DL/I online nucleus module, DLZNUCxx.

NOT AT CURRENT LEVEL

The DL/I online nucleus module, DLZNUCxx, was not generated using the DLZACT macro of current level DL/I system.

In all cases, DL/I will not be initialized. A dummy DL/I nucleus will be used which returns control to the application program when DL/I is called without performing any DL/I function. The system continues with CICS/VS initialization only.

User Response: Determine the name of the DL/I nucleus module being used. The name of the nucleus module is DLZNUCxx, where xx is the suffix provided in the CICS/VS System Initialization Table parameter DLI=xx.

NOT FOUND

Check that:

- The correct suffix was specified in the CICS/VS System Initialization Table DLI= parameter.

- DLZNUCxx was not removed.
- The correct library was assigned during the DL/I nucleus generation step (ACTGEN procedure) and that it is defined in the search chain.

HAS WRONG ENTRY POINT

Check that:

- An ENTRY DLZNUC statement was included when cataloging this nucleus.
- No error occurred during linkage editing of this nucleus.

NOT AT CURRENT LEVEL

Check that DLZNUCxx was generated using the DLZACT macro that is provided with the current level DL/I system.

Rebuild the DL/I nucleus, if required. Catalog and link-edit the DL/I nucleus as a phase in a library using the ACTGEN procedure and ensure that this library is defined in the search chain.

Rerun the CICS/VS and DL/I initialization job.

DLZ065I DL/I ABNORMALLY TERMINATED WHILE DL/I TASK(S) ACTIVE

Explanation: An immediate CICS/VS shutdown was specified while active DL/I tasks were in progress.

User Response: When the DL/I online system has abnormally terminated, it is necessary to determine the status of the system before attempting to restart. Any alterations to the data base being used by the DL/I tasks active at the time of termination must be backed out by using the DL/I data base backout utility. This program must be run once for each active PSB. After the transactions have been backed out, the system may be restarted and the backed-out transactions reentered.

DLZ066I ERROR DURING CLOSE, DL/I ABNORMALLY TERMINATED

Explanation: The data base files as specified in preceding close error message (DLZ020I) could not be closed.

User Response: Use the access method services program to VERIFY the specified data base files prior to next execution of DL/I. The data files must be closed prior to the next execution of DL/I.

DLZ067I INSUFFICIENT STORAGE TO TERMINATE DL/I

Explanation: When attempting to acquire control block storage for DL/I, CICS/VS could not satisfy the request.

User Response: Use the access method services program to VERIFY all data base files prior to the next execution of DL/I.

DLZ068I SYSTEM PREVIOUSLY ABENDED, DL/I ABNORMALLY TERMINATED

Explanation: The message DLZ062I has occurred and DL/I is not available for shutdown.

User Response: When the DL/I online system has abnormally terminated, it is necessary to determine the status of the system before attempting to restart. Any alterations to the data base being used by the DL/I tasks active at the time of termination must be backed out by using the DL/I data base backout utility. This program must be run once for each active PSB. After the transactions have been backed out, the system may be restarted and the backed-out transactions reentered.

DLZ069I DL/I NORMAL TERMINATION COMPLETE

Explanation: DL/I has successfully terminated.

User Response: None required.

DLZ070I ONLINE DL/I ABNORMAL TERMINATION COMPLETE

Explanation: DL/I has been called by the CICS/VS system recovery program because the system cannot continue.

User Response: The DL/I online system has abnormally terminated, the transactions have been backed out by CICS/VS dynamic transaction back out facility, if it is in effect for active transactions. The system may be restarted after the reason for abnormal termination of the system has been identified and problem corrected.

DLZ071I PSB *psbname* VERSION INVALID, CANNOT BE USED

Explanation: A PSB created by an incompatible level of DL/I was attempted to be initialized.

User Response: If message DLZ040A has not been stopped by a previous RUN response, it is issued after this message.

Respond to DLZ040A as follows:

- To continue with DL/I and CICS/VS initialization, enter either:
 GO This data base will be disabled.
 RUN This data base will be disabled.
 (DL/I will stop issuing DLZ040A messages.)
- To stop DL/I and CICS/VS initialization, enter either:
 CANCEL To end DL/I and CICS/VS initialization.
 DUMP To end DL/I and CICS/VS initialization and DUMP the contents of storage at this point.

To correct this problem, rebuild this PSB using the PSBGEN and ACBGEN procedures of the current level of DL/I.

Rerun the CICS/VS and DL/I initialization job.

DLZ072I DBD *dbdname* VERSION INVALID, CANNOT BE USED

Explanation: A DBD created by an incompatible level of DL/I was attempted to be initialized. A PSB scheduling error will be returned whenever a program attempts to schedule a PSB which refers to this data base.

User Response: If message DLZ040A has not been stopped by a previous RUN response, it is issued after this message.

Respond to DLZ040A as follows:

- To continue with DL/I and CICS/VS initialization, enter either:
 GO This data base will be disabled.
 RUN This data base will be disabled.
 (DL/I will stop issuing DLZ040A messages.)
- To stop DL/I and CICS/VS initialization, enter either:
 CANCEL To end DL/I and CICS/VS initialization.
 DUMP To end DL/I and CICS/VS initialization and DUMP the contents of storage at this point.

To correct this problem, rebuild this DBD using the DBDGEN and ACBGEN procedures of the current level of DL/I.

Rerun the CICS/VS and DL/I initialization job.

DLZ073I COMPRESSION MODULE *compname* FOR DBD *dbdname* NOT FOUND

Explanation: Compression module *compname* is required by the Data Base Description (DBD) *dbdname*. Module *compname* was not found in any of the libraries defined in the search chain and, therefore, was not loaded.

User Response: If message DLZ040A has not been stopped by a previous RUN response, it is issued after this message.

Respond to DLZ040A as follows:

- To continue with DL/I and CICS/VS initialization, enter either:
 GO This data base will be disabled.
 RUN This data base will be disabled.
 (DL/I will stop issuing DLZ040A messages.)
- To stop DL/I and CICS/VS initialization, enter either:
 CANCEL To end DL/I and CICS/VS initialization.
 DUMP To end DL/I and CICS/VS initialization and DUMP the contents of storage at this point.

To correct this problem, catalog and link-edit the compression module as a phase in the correct library using the system link-edit procedure.

Rerun the CICS/VS and DL/I initialization job.

DLZ074I INDEX EXIT MODULE *indxname* FOR DBD *dbdname* NOT FOUND

Explanation: Secondary index exit module *indxname* is required by the Data Base Description (DBD) *dbdname*. Module *indxname* was not found in any of the libraries defined in the search chain and, therefore, was not loaded.

User Response: If message DLZ040A has not been stopped by a previous RUN response, it is issued after this message.

Respond to DLZ040A as follows:

- To continue with DL/I and CICS/VS initialization, enter either:
 GO This data base will be disabled.
 RUN This data base will be disabled.
 (DL/I will stop issuing DLZ040A messages.)
- To stop DL/I and CICS/VS initialization, enter either:
 CANCEL To end DL/I and CICS/VS initialization.
 DUMP To end DL/I and CICS/VS initialization and DUMP the contents of storage at this point.

To correct this problem, check that:

- The correct secondary index exit module name was used in the EXTRTN= parameter of the XDFLD statement when building this DBD. If this caused the problem, rebuild the DBD with the correct routine name using the DBDGEN procedure.
- Secondary index exit module *indxname* was not removed from the library. If this caused the problem, catalog and link-edit the secondary index exit module as a phase in the correct library using the system link-edit.

Rerun the CICS/VS and DL/I initialization job.

DLZ075I LOG PARAMETER IS INCORRECTLY SPECIFIED

Explanation: An incorrect subparameter value was specified for the LOG parameter in the DL/I parameter statement. The only valid values for the first subparameter are TAPE, DISK1, DISK2, or nothing specified. The only valid values for the second subparameter are PAUSE, NOPAUSE, or nothing specified. (It is incorrect to omit both subparameters if the LOG parameter is specified.)

User Response: Correct LOG parameter values and resubmit job.

DLZ076A DLI LOG {DSKLOG1|DSKLOG2} IS FULL REPLY 'GO' TO CONTINUE OR 'CANCEL'

Explanation: The currently used DL/I disk log extent is full. Depending on the options specified in the LOG parameter on the DL/I parameter statement, this disk log extent will either be overwritten, or DL/I will switch to the other disk log extent. When this message is issued, the current disk log extent is closed.

User Response: The operator may want to start a program in another partition to copy the data from the disk log extent that is full. When processing can be resumed, enter GO. If you want to terminate DL/I, enter CANCEL. This results in the cancellation of DL/I execution.

DLZ0771 DL/I LOG DATA BEING RECORDED ON {DSKLOG1|DSKLOG2}

Explanation: The log file indicated in the message text has just been opened and is going to be used/reused to record DL/I log data.

User Response: If two log files are being used, and if the contents of the previously used log file are to be preserved, it should be dumped at this time.

DLZ0781 LOG PARAMETER SPECIFIED WITH NO LOGGING BEING USED

Explanation: The LOG parameter was specified on the DL/I parameter statement, but no logging was specified in the UPSI byte. The job is canceled.

User Response: Resubmit the job with UPSI bit 6 set to zero if logging is desired, or remove the LOG parameter from the DL/I statement parameter if logging is not wanted.

DLZ0791 LOG FULL, RUN CANCELLED BY OPERATOR

Explanation: The disk extents specified for logging became full and the operator requested that the run be canceled in reply to message DLZ076A.

User Response: Increase the size of the disk extent, or make arrangements to dump the extent before continuing to log, and resubmit job after restoring data base. Or dump existing log and resubmit the job from point of previous cancel.

DLZ0801 MPS STOP ATTEMPTED WHEN MPS NOT ACTIVE - IGNORED

Explanation: MPS stop transaction processing was attempted when MPS was not active or was already stopping. It has no effect.

User Response: None required.

DLZ0811 MPS BATCH DL/I PARTITION {STARTED|RESTARTED}

Explanation: STARTED indicates that initialization for an MPS batch job has successfully completed and control has been given to the application program. RESTARTED indicates that reinitialization for an MPS batch job which has been restarted using the VSE Restart Facility has successfully completed.

User Response: None required.

DLZ0821 XECBTAB {DEFINE|DELETE|CHECK} FAILED FOR XECB *xecbname* IN [*id*] *routname*, RC=*rc*

Explanation: Return code *rc* was received when the *routname* routine issued an XECBTAB macro for use in DL/I Multiple Partition Support (MPS). *xecbname* is the name of the XECB that caused the unsuccessful DEFINE, DELETE, or CHECK. In the message text, *routname* can be:

DLZMPC00	Master Partition Controller (MPC)
DLZBPC00	Batch Partition Controller (BPC)
DLZMINIT	Batch partition initialization
DLZMPRH	Program request handler

DLZMINIT and DLZMPRH are part of the MPS batch module, DLZMPI00. If *routname* is DLZBPC00, the partition identifier *id* of the batch partition that the BPC is serving appears in the message text. The batch partition terminates abnormally.

For an explanation of the XECBTAB macro return code *rc* for a CHECK, DEFINE or DELETE request refer to *VSE/SP Messages and Codes*, Section "VSE/Advanced Functions Return Codes" and to *DL/I DOS/VS 1.8 Logic Extensions*, "Diagnostic Aids for MPS Messages."

User Response: Correct the error and rerun the program.

DLZ0831 MPS CANNOT BE STARTED, MPC ATTACH FAILED

Explanation: Multiple Partition Support cannot be started because the Master Partition Controller (MPC) could not be attached due to insufficient CICS/VS storage.

Note: For further information refer to *DL/I DOS/VS Diagnostic Guide*, "Appendix A: MPS Diagnostic Aids."

User Response: Increase the SIZE parameter in the EXEC statement prior to reinitialization of CICS/VS.

DLZ0841 {XPOST|XWAIT} FAILED FOR XECB *xecbname* IN [*id*] *routname*, RETURN CODE=*rc*

Explanation: Return code *rc* was received when the *routname* routine issued an XPOST or XWAIT macro for use in DL/I Multiple Partition Support (MPS).

For an explanation of return code *rc* refer to *VSE/SP Messages and Codes*, Section "VSE/Advanced Functions Return Codes."

xecbname is the name of the XECB that caused the unsuccessful XPOST or XWAIT. In the message text, *routname* can be:

DLZMPC00	Master Partition Controller (MPC)
DLZBPC00	Batch Partition Controller (BPC)
DLZMINIT	Batch partition initialization
DLZMPRH	Program request handler

DLZMINIT and DLZMPRH are part of the MPS batch module, DLZMPI00. If *routname* is

DLZBPC00, the partition identifier *id* of the batch partition that the BPC is serving appears in the message text.

Note: For further information refer to *DL/I DOS/VS 1.8 Logic Extensions*, "Diagnostic Aids for MPS Messages."

User Response: The batch partition ends unless the routine is DLZBPC00 or DLZMPC00 for XPOST. If there is no other message concerning the ending of a batch partition, the operator will have to cancel it.

DLZ085I MPS BATCH PARTITION ENDED ABNORMALLY, BPC ATTACH FAILED

Explanation: The MPS batch partition ended abnormally because the Master Partition Controller (MPC) could not attach the BPC (Batch Partition Controller) that was to service it.

Note: For further information refer to *DL/I DOS/VS 1.8 Logic Extensions*, "Diagnostic Aids for MPS Messages."

User Response: Define the BPC (DLZBPC00) in the CICS/VS processing program table (PPT) and define the BPC transaction ID (CSDC) in the CICS/VS program control table (PCT).

DLZ086I MPS STOP REQUEST DELAYED - nn TASKS ACTIVE

Explanation: Multiple Partition Support (MPS) was signaled to stop and will terminate after all currently active (*nn*) MPS batch partitions have terminated. No further MPS jobs are allowed to start.

User Response: None required.

DLZ087A INVALID PARAMETER SPECIFICATIONS, ENTER PARAMETERS OR 'CANCEL'

Explanation: An MPS batch partition read a parameter statement containing one of the following invalid conditions.

- The function code value was not DLI or DLR.
- The program name was DLZBACK0, DLZPRCT2, or was not between one and eight characters.
- The PSB (program specification block) was not between one and seven characters.

This message is followed by the incorrect parameter plus the rest of the input data.

User Response: Reenter the entire parameter statement with the DLI function code if utilities are not to run. If a utility is to be run it must be done in batch (non-MPS). Reply CANCEL to this message if the job is not valid for MPS. If the program name was not between one and eight characters or the PSB name was not between one and seven characters, respecify them if the correct values are known, otherwise enter CANCEL.

DLZ088I MPS BATCH PARTITION CANNOT START, MAXIMUM MPS JOBS ACTIVE

Explanation: The MPS batch partition could not start because the MPS batch initialization routine (DLZMPI00) detected that the maximum number of concurrent MPS jobs (currently 12) were running at the time the batch job attempted to start.

User Response: If the maximum number of concurrent MPS batch jobs were active at the time of failure, restart the job when one of the MPS batch jobs completes.

DLZ089I MPS BATCH PARTITION CANNOT START, MPC NOT ACTIVE OR ENDING

Explanation: The Multiple Partition Support (MPS) batch partition is unable to start because it could not find an online partition with active DL/I MPS support. Either MPS was not started or was terminated, either normally or abnormally.

User Response: Start MPS in the online (CICS/VS) partition and resubmit the batch job.

DLZ090I MPS STOP ATTEMPTED WHEN DL/I NOT ACTIVE - IGNORED

Explanation: Multiple Partition Support was signaled to stop, but DL/I was either not active or was already terminating.

User Response: None required.

DLZ091I MPS BATCH PARTITION ENDED ABNORMALLY, TOO MANY PARAMETERS

Explanation: More than eighteen parameters were passed to DL/I by the application program CALL statement.

User Response: Make sure that no more than eighteen parameters are used on a CALL statement to DL/I.

DLZ092I MPS PARTITION ENDED ABNORMALLY, PARAMETERS NOT WITHIN PARTITION

Explanation: In the MPS batch partition, one of the parameters passed to DL/I by an application program CALL statement is invalid. It is not within the partition boundary. This does not apply to the PCB address which is known to be in the online partition.

User Response: Make sure that all addresses are within the partition.

DLZ093I MPS STARTED

Explanation: The Master Partition Controller (MPC) for DL/I Multiple Partition Support (MPS) started successfully. Batch jobs for MPS can now be started.

User Response: None required.

DLZ094I MPS STOPPED {ABNORMALLY|NORMALLY}

Explanation: Multiple Partition Support (MPS) ended abnormally or normally. If it ended abnormally, the previous message gives the reason. No further batch jobs can be run in an MPS environment.

User Response: None required.

DLZ095I MPS BATCH PARTITION ENDED, SCHEDULING RETURN CODE=yyzz

Explanation: The Multiple Partition Support (MPS) batch partition was signaled to end by the online partition because there is a scheduling error. The scheduling return code yy is the contents of TCAFCTR (or TCAFRCR if ANS COBOL) and zz is the contents of TCADLTR. Refer to "DL/I CICS/VS Return Codes" on page 111 for an explanation of the exit condition return codes.

Note: For further information refer to *DL/I DOS/VS 1.8 Logic Extensions*, "Diagnostic Aids for MPS Messages."

User Response: Resubmit the batch job after the cause for the scheduling error is found and corrected.

DLZ096I STXIT {AB|PC} ENTERED, MPS BATCH PARTITION ENDED ABNORMALLY

Explanation: Control was passed to the STXIT AB or STXIT PC entry point in the MPS batchabend handler routine (DLZMABND).

User Response: Examine the log and dump, if printed, to determine the cause of the error. The corresponding register save areas are immediately preceded in the dump of module DLZMPI00 by the eye-catchers 'AB SAVE' or 'PC SAVE' respectively. The one-byte reason code for AB STXIT entry is preceded by 'AB REASON CODE'. In the dump, Register 3 contains the address of the applicable save area.

Note: Additional aids for interpreting and debugging dumps after this message is issued can be found in the *DL/I DOS/VS 1.8 Logic Extensions*, "Diagnostic Aids for MPS Messages."

DLZ097I MPS START ATTEMPT WHEN DL/I NOT ACTIVE

Explanation: There was an attempt to start Multiple Partition Support (MPS) for DL/I, but DL/I was not active because DL/I initialization failed, or a dummy nucleus was used for initialization. See messages issued during CICS/VS initialization for more information.

User Response: Be sure the CICS/VS system generation macros are coded:

```
DFHSG DL1=YES
DFHSIT DL1={xx|YES}
```

where xx is the suffix used with the nucleus name. Also, be sure that the nucleus specified at

initialization is in any of the libraries defined in the search chain.

Note: Additional aids for interpreting and debugging dumps after this message is issued can be found in the *DL/I DOS/VS Diagnostic Guide*, "Appendix A: MPS Diagnostic Aids."

DLZ098I RETURN CODE nnn FROM DL/I CALL

Explanation: The MPS batch program request handler routine was given a pseudo-abend error return code (nnn) from the online partition processing of a DL/I CALL statement. nnn is the numeric part of another DL/I message that gives more information about the error: Refer to message DLZnnnl, where nnn is the return code.

User Response: Take appropriate action in the application program to prevent abnormal terminations. If the error damaged the data base, take recovery action.

DLZ099I XPCC function FAILED FOR xpccbname IN [id] routname, RC=rc, RS=rs

Explanation: Multiple Partition Support (MPS) uses VSE XPCC support to communicate across address spaces or, for initialization, across partitions.

DL/I was attempting to communicate using XPCC, doing function *function* and received return code *rc* and reason code *rs* from VSE. An explanation of return code *rc* and reason code *rs* can be found in DSECT MAPXPCCB (or in the manual *VSE/Advanced Functions Diagnosis Reference Supervisor*).

xpccbname is the XPCC FROM application for the XPCCB which had the unsuccessful call.

In the message text, *routname* can be:

DLZMPC00	Master Partition Controller (MPC)
DLZBPC00	Batch Partition Controller (BPC)
DLZMINIT	Batch partition initialization
DLZMPRH	Program request handler

DLZMINIT and DLZMPRH are part of the MPS batch module, DLZMPI00.

- If *routname*=DLZBPC00, the partition identifier *id* of the batch partition that the BPC is serving appears in the text. The batch partition will send a message when it terminates.
- If *routname*=DLZMPC00, the message following DLZ099I will give you more information about MPS termination.

If the following message is DLZ086I, the Master Partition Controller encountered an error during CONNECT or DISCONNECT of the Start Partition XPCC block and at least one MPS batch partition was active. In this case MPS will terminate after completion of all running MPS batch partitions.

Note: Additional aids for interpreting and debugging dumps after this message is issued

can be found in the *DL/I DOS/VS 1.8 Logic Extensions*, "Diagnostic Aids for MPS Messages."

User Response: Correct the error as indicated by *rc* and *rs* and restart MPS.

DLZ100I MPS BATCH PARTITION ENDED ABNORMALLY, CODE = xx

Explanation: Depending on the code *xx*:

- | xx | Explanation |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 01 | The MPS batch PRH (program request handler) routine was notified that the BPC (Batch Partition Controller) terminated abnormally while the PRH was waiting for a DL/I call to be processed. |
| 02 | The MPS batch PRH determined that the address of the user I/O area and/or the length of the data to be moved there would cause the data to be outside the valid addresses for the user program, or else the length of the data was negative. |

User Response: If the code is:

- | xx | User Response |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 01 | Check online messages to determine if the BPC gave a reason for terminating. Save the log and dump and contact the IBM Support Center. |
| 02 | If the online partition terminated abnormally, correct the problem and take any recovery action needed. Otherwise, save the log and dump and contact the IBM Support Center. |

Note: Additional aids for interpreting and debugging dumps after this message is issued can be found in the *DL/I DOS/VS 1.8 Logic Extensions*, "Diagnostic Aids for MPS Messages."

DLZ101I MPS START ATTEMPTED WHEN MPS ACTIVE, REQUEST IGNORED

Explanation: Multiple Partition Support (MPS) start transaction processing was initiated when MPS was active or in the process of stopping. It has no effect.

User Response: None required.

DLZ102I BAD RETURN FROM ONLINE PARTITION. RETURN CODE = yyzz

Explanation: The Multiple Partition Support (MPS) batch partition received an error return code from a call to the online partition. This was probably from a PCB or TERM call issued by the MPS application program, although it may have been from another type of call.

Return code *yy* is the contents of TCAFCTR (or TCAFRCR if ANS COBOL) and *zz* is the contents of TCADLTR. Refer to "DL/I CICS/VS Return Codes" on page 111 for an explanation of the exit condition return codes.

Note: For further information refer to *DL/I DOS/VS 1.8 Logic Extensions*, "Diagnostic Aids for MPS Messages."

User Response: Resubmit the batch job after the cause for the return code is found and corrected.

DLZ103I id BPC STOPPED {ABNORMALLY|NORMALLY}

Explanation: The BPC (Batch Partition Controller) ended abnormally or normally. If it ended abnormally and there was a previous message, it gives the reason. The batch job for this BPC also ends; *id* is the partition identifier of the batch partition associated with the BPC. If the BPC ended abnormally and there was no previous message, the batch partition ended abnormally or was canceled.

User Response: If the BPC ended abnormally, correct the cause and resubmit the job.

DLZ104I {MPC|BPC} ABENDED, COMPLETION CODE = code [, PSW = xxxxxxxx xxxxxxxx]

Explanation: Multiple Partition Support (MPS) ended because the Master Partition Controller (MPC) or the Batch Partition Controller (BPC) terminated abnormally. The completion code indicated in the message text is from TCAPCAC in the task communication area of CICS/VS. If the completion code is ASRA, program interrupt PSW information also appears in the message text. The PSW is from TCAPCPSW.

User Response: Correct the cause for the abnormal termination of the MPC or BPC and take appropriate recovery action. Then restart MPS.

DLZ105I DL/I CHECKPOINT TAKEN, PSB = psbname ID = id

Explanation: A program has taken a checkpoint.

User Response: The checkpoint-id should be saved since it may be valuable information to aid in backout and restart processing.

DLZ106I TASK TERMINATION DUE TO UNRESOLVABLE CONFLICT IN DATA USAGE

Explanation: The task being terminated was involved in a loop where each task involved owned (had enqueued) a resource and was waiting (had an enqueued request outstanding) for a resource the next task owned.

User Response: Resource contention problem is very dependent on current mix of tasks. Rerun the task.

Programmer Response: If the rerun is unsuccessful, reevaluate the mix of tasks and the resources they reference. (Refer to *DL/I DOS/VS Logic Manual, Volume 1*, for information on "DLZQUEF0 - Queuing Facility" and the layout of the resource descriptor block (RDB) and the resource request descriptor (RRD).)

DLZ108I TASK TERMINATION DUE TO INSUFFICIENT CICS PARTITION SPACE

Explanation: No additional GETVIS storage was available in the CICS/VS partition when an attempt was made to get some for enqueue control blocks.

User Response: Rerun the task.

If this condition persists, it may be necessary to do at least one of the following:

- Increase the size of the GETVIS area in the CICS/VS partition.
- Decrease the number of concurrently executing tasks.
- Rewrite the tasks to issue checkpoint (CHKP) calls.
- Rewrite the tasks to issue CICS/VS synchronization point (SYNCPPOINT) calls.

DLZ111I RC PARAMETER INVALID. 'NO' ASSUMED

Explanation: Only YES or NO may be specified as RC option. The RC option will be set to NO.

User Response: Correct the parameter in error before running the job again.

DLZ112I HDBFR PARAMETER NOT ALLOWED FOR THE SPECIFIED PSB

Explanation: HDBFR parameter has been specified for a PSB that cannot use the HDBFR parameter.

User Response:

- Batch:
Correct the DL/I parameter statement and rerun the job.
- Online:

If message DLZ040A has not been stopped by a previous RUN response, it is issued after this message.

Respond to DLZ040A as follows:

- To continue with DL/I and CICS/VS initialization, enter either:

GO	This data base will be disabled.
RUN	This data base will be disabled. (DL/I will stop issuing DLZ040A messages.)
- To stop DL/I and CICS/VS initialization, enter either:

CANCEL	To end DL/I and CICS/VS initialization.
DUMP	To end DL/I and CICS/VS initialization and DUMP the contents of storage at this point.

To correct this problem, remove this DBD from your DL/I online nucleus definition by updating the DLZACT statements used to build this nucleus. This is done by removing this DBD

name from all DLZACT TYPE=BUFFER statements.

Rebuild the DL/I nucleus (ACT) using the ACTGEN procedure.

Rerun the CICS/VS and DL/I initialization job.

DLZ113I PROGRAM ISOLATION {ACTIVE|INACTIVE}

Explanation: This is an information message issued during online initialization. It specifies the status of program isolation.

User Response: None required.

DLZ114I UNREFERENCED DBD *dbdname* SPECIFIED IN {HDBFR|HSBFR} PARAMETER

Explanation: Data Base Description (DBD) *dbdname* is not referenced by the PSB(s) being used but is specified as an HDBFR or HSBFR parameter in the DL/I parameter statement (batch) or in the DLZACT TYPE=BUFFER statement (online).

User Response:

- Batch:
Correct the DL/I parameter statement and rerun the job.
- Online:

If message DLZ040A has not been stopped by a previous RUN response, it is issued after this message.

Respond to DLZ040A as follows:

- To continue with DL/I and CICS/VS initialization, enter either:

GO	This data base will be disabled.
RUN	This data base will be disabled. (DL/I will stop issuing DLZ040A messages.)
- To stop DL/I and CICS/VS initialization, enter either:

CANCEL	To end DL/I and CICS/VS initialization.
DUMP	To end DL/I and CICS/VS initialization and DUMP the contents of storage at this point.

To correct this problem, check that:

- The PSB(s) that referenced this DBD were initialized successfully. If this caused the problem, correct the PSB initialization problem.
- The correct PSB names were specified in the DLZACT TYPE=PROGRAM statements used to build this online nucleus. If this caused the problem, correct the PSB names used to build this nucleus. This is done by changing the incorrect PSB name in all DLZACT TYPE=PROGRAM statements. Rebuild the DL/I nucleus (ACT) using the ACTGEN procedure.

- The correct DBD name was specified in the DLZACT TYPE=BUFFER statements used to build this online nucleus. If this caused the problem, remove this DBD from your DL/I online nucleus definition by updating the DLZACT statements used to build this nucleus. This is done by removing this DBD name from all DLZACT TYPE=BUFFER statements.

Rebuild the DL/I nucleus (ACT) using the ACTGEN procedure and rerun the CICS/VS and DL/I initialization job.

DLZ115I DBD *dbdname* HAS DATA BASE TYPE NOT ALLOWED IN {HDBFR|HSBFR} PARAMETER

Explanation: An incorrect specification was made in an HDBFR or HSBFR parameter in the DL/I parameter statement (batch) or in the DLZACT TYPE=BUFFER statement (online) or an INDEX DBD was specified.

User Response:

- Batch:
 - If an HDBFR parameter was specified, ensure that:
 - No HSAM or simple HSAM DBD is specified.
 - No HISAM or simple HISAM or INDEX DBD is specified, when the processing option specified in the PROCOPT= parameter of the SENSEG or PCB parameter of the PSB being used does not include load or insert.
 - If an HSBFR parameter was specified, ensure that no HDAM or HIDAM DBD is specified.

Correct the DL/I parameter statement and rerun the job.

- Online:
 - If an HDBFR parameter was specified, ensure that no INDEX DBD is specified.
 - If an HSBFR parameter was specified, ensure that no HDAM or HIDAM DBD is specified.

If message DLZ040A has not been stopped by a previous RUN response, it is issued after this message.

Respond to DLZ040A as follows:

- To continue with DL/I and CICS/VS initialization, enter either:

GO	This data base will be disabled.
RUN	This data base will be disabled. (DL/I will stop issuing DLZ040A messages.)

- To stop DL/I and CICS/VS initialization, enter either:

CANCEL	To end DL/I and CICS/VS initialization.
DUMP	To end DL/I and CICS/VS initialization and DUMP the contents of storage at this point.

To correct this problem, remove this DBD from your DL/I online nucleus definition by updating the DLZACT statements used to build this nucleus. This is done by removing this DBD name from all DLZACT TYPE=BUFFER statements.

Rebuild the DL/I nucleus (ACT) using the ACTGEN procedure and rerun the CICS/VS and DL/I initialization job.

DLZ116I DBDNAME NOT SPECIFIED FOR {HDBFR|HSBFR} PARAMETER

Explanation: The DBDNAME is missing in an HDBFR or HSBFR parameter of DL/I parameter statement. The subparameter that caused the error, plus all parameter information following it that was already read, is listed before this message.

User Response: Correct the DL/I parameter statement and reexecute the job.

DLZ117I DUPLICATE DBD *dbdname* SPECIFIED IN HDBFR PARAMETER

Explanation: The DBDNAME encountered in an HDBFR statement is the same as a DBDNAME found in a previous HDBFR statement. The subparameter that caused the error, plus all parameter information following it that was already read, is listed before this message.

User Response: Correct the DL/I parameter statement and reexecute job.

DLZ118I DL/I DOS/VS *version* INITIALIZATION IN PROGRESS

Explanation: This is an information message issued during online initialization. It specifies the version and level of DL/I currently running.

User Response: None required.

DLZ119I DL/I DATABASE UPDATES WILL {NOT BE LOGGED| BE LOGGED TO THE CICS/VS JOURNAL| BE LOGGED TO THE DL/I LOG}

Explanation: This is an information message issued during online initialization. It specifies the status of CICS/VS journaling and DL/I logging.

User Response: None required.

DLZ120I DL/I TRACE DISABLED, ERROR NUMBER = *n*

Explanation: The tracing facility has discovered an error while trying to write the INCORE table to its CICS/VS extra-partition data set. Tracing is disabled and all trace entries from this point on are lost. The task continues to run, but without tracing.

n	Explanation
1	The ID (DLIT) used for the output extra-partition data set cannot be located by CICS/VS.
2	An I/O error occurred on the trace output device used by CICS/VS.
3	There is no more space available on the trace output file.
4	The trace output file is not open.

User Response: Correct the error condition on the CICS/VS extra-partition data set device and, if necessary, rerun the job.

If tracing is to be continued, it will be necessary to stop trace and restart it.

DLZ121I RESTART MPS BATCH JOB(S) USING THE FOLLOWING CHECKPOINT ID(S):

Explanation: This message is issued during MPS initialization as an introduction to one or more DLZ122I messages which will follow it.

User Response: Use the following DLZ122I messages(s) to obtain information required to restart MPS batch jobs.

DLZ122I JOB=*jobname* PARTITION=*id* DATE=*mm/dd/yy* TIME=*hh:mm:ss* CHKPT=*n*

Explanation: This message is issued for each MPS batch job using the MPS Restart facility which had taken a combined checkpoint during a previous MPS session and had not terminated normally. Typically, one or more of these messages will appear when starting up MPS after a system failure which occurred while restartable MPS batch jobs were running. The checkpoint ID to be used for restarting the job is given by *n*. The DATE and TIME information tells when the job originally started and may help to identify a particular job.

User Response: Identify each MPS batch job to be restarted and obtain the checkpoint ID to be used on the VSE RSTRT job control statement.

DLZ123I [*id*] *rouname* TEMPORARY STORAGE ERROR, TYPE={GETQ|PUTQ|PURGE} RESPONSE CODE=*rc*

Explanation: An unexpected response code was returned while using CICS/VS Temporary Storage Control to access the Temporary Storage Queue (TSQ) used by the MPS Restart facility. The name of the routine in which the error occurred is given by *rouname*, which will be one of the following:

DLZMPC00	Master Partition Controller (MPC)
----------	-----------------------------------

DLZBPC00	Batch Partition Controller (BPC)
DLZMPUR0	Purge Temporary Storage Transaction

- If *rouname* is DLZBPC00 then the partition identifier *id* of the batch partition that the BPC is serving appears in the message text.
- If *rouname* is DLZMPC00 and:
 - A partition ID appears in the message text, then the error occurred in the BPC abnormal termination routine located in DLZMPC00. This routine is executed under control of the BPC task serving the partition indicated.
 - No partition ID appears, the error occurred during start or stop partition processing in MPC.
- If *rouname* is DLZMPUR0, the error occurred when transaction CSDP was attempting to purge the Temporary Storage Queue (TSQ).

TYPE=*xxxx* corresponds to the TYPE=*operand* of the DFHTS macro and indicates whether the TSQ was being read from (TYPE=GETQ), written to (TYPE=PUTQ), or purged (TYPE=PURGE).

The meaning of response code *rc* can be obtained from *CICS/VS Application Programmer's Reference Manual (Macro Level)*, Chapter "Temporary Storage Control (DFHTS Macro)."

User Response: Determine the cause of the error and correct it. Restart the MPS batch job if it was canceled due to the error. If data in the TSQ was altered or destroyed because of the error, it may be necessary to purge the TSQ and use the checkpoint ID verification override option (see message DLZ123I) in order to restart jobs.

DLZ124I MPS RESTART IS NOT AVAILABLE, {CICS JOURNAL IS NOT ACTIVE|TEMPORARY STORAGE IS NOT AVAILABLE}

Explanation: An MPS batch job requesting the MPS Restart facility ('DLR' function code) was submitted for execution, but the necessary environment did not exist. Either the CICS/VS journal was not active, or the Temporary Storage facility was not available.

User Response: Correct the environment and resubmit the job.

DLZ125I WRONG CHECKPOINT ID WAS USED ON RESTART, USE CHECKPOINT ID *n*

Explanation: A VSE restart was attempted for an MPS batch job using the MPS restart facility, but the checkpoint ID specified on the VSE RSTRT job control statement was incorrect. The DL/I MPS Restart facility maintains the checkpoint ID of the last successful combined checkpoint taken before an MPS batch job fails. This message is issued and the job is canceled when the checkpoint ID specified on restart does not match the checkpoint ID maintained by DL/I.

User Response: Change the checkpoint ID specified on the VSE RSTRT job control

statement so that it matches the checkpoint ID *nnnn* provided in the message and resubmit the job.

DLZ126I CHECKPOINT ID NOT VERIFIED, TEMPORARY STORAGE PURGED OR MODIFIED

Explanation: During MPS batch reinitialization following a VSE restart, the DL/I MPS Restart facility could not verify that the checkpoint ID used on the VSE RSTRT job control statement corresponded to the last successful combined checkpoint for the job being restarted. This condition is caused when the Temporary Storage Queue (TSQ) in which DL/I maintains checkpoint IDs is not found or has been modified so that the checkpoint ID for the job being restarted is no longer available. This situation may result from purging the TSQ using transaction CSDP. However, the TSQ should not be purged when there are MPS batch jobs to be restarted, since checkpoint ID verification is then no longer possible for those jobs.

User Response: This message is followed by message DLZ059A which allows the user to cancel the job or continue execution. The user may want to cancel the job if there is any question about the correctness of the checkpoint ID, since restarting from the wrong checkpoint ID could invalidate data in the data base. The job may be allowed to continue if the user is sure that the checkpoint ID is correct. If the job is continued, the checkpoint ID used for restart will become the one maintained by DL/I to be used in case another failure occurs before the next combined checkpoint is taken. If the TSQ was not explicitly purged or modified, steps should be taken to determine what happened to the TSQ and why.

DLZ127I RESTART JOB *jobname* IN PARTITION *id* {USING CHECKPOINT ID *nnnn*}FROM BEGINNING OF JOB STEP}

Explanation: An MPS batch job using the MPS Restart facility abnormally ended. If the job issued one or more combined checkpoints before abnormally ending, then the correct checkpoint ID to use for restarting the job is indicated in the message. If the job did not issue a combined checkpoint before abnormally ending, then the message will state that the job should be restarted (started over in this case) from the beginning of the job step.

User Response: If the job issued a combined checkpoint before abnormally ending, use the checkpoint ID *nnnn* provided in the message on the VSE RSTRT job control statement to restart the job in the indicated partition *id*. Otherwise, resubmit the job for execution from the beginning of the job step in which the failure occurred (use the VSE EXEC job control statement).

DLZ128I TEMPORARY STORAGE QUEUE FOR MPS RESTART HAS BEEN PURGED

Explanation: This message indicates that the Temporary Storage Queue (TSQ) used by the MPS Restart facility has been successfully purged. The purge is requested by executing transaction CSDP.

User Response: None.

DLZ129I MPS BATCH PARTITION ENDED ABNORMALLY, PL/I RETURN CODE=*nnnn*

Explanation: A PL/I program running in an MPS batch partition ended with a return code greater than 999.

For information on the PL/I execution-time return code *nnnn* refer to the *DOS PL/I Optimizing Compiler: Programmer's Guide*.

User Response: Correct the error in the PL/I program and rerun the job.

DLZ130I TSQ WAS NOT PURGED, RESTARTABLE MPS BATCH JOB IS ACTIVE

Explanation: Transaction CSDP was executed, but the Temporary Storage Queue was not purged because an MPS batch job using MPS Restart was running when the purge was requested.

User Response: Wait until the MPS batch jobs using MPS Restart have normally terminated and reenter the purge request.

DLZ131I I/O ERROR DURING CHECKPOINT, MPS BATCH PARTITION CANCELLED

Explanation: A buffer write-out error occurred during a DL/I checkpoint issued by an MPS batch program using the MPS Restart facility. DL/I will cancel the job instead of returning an 'XD' status code to the application program. This is because the Temporary Storage Queue (TSQ) used by MPS Restart to maintain checkpoint IDs was updated before the DL/I checkpoint was attempted and cannot be committed (and subsequently made available for use by other tasks) unless a successful checkpoint is taken.

User Response: Correct the cause of the I/O error and restart the job using the checkpoint ID provided in message DLZ127I which was issued when the job was canceled.

DLZ132I PCB NOT SPECIFIED OR INVALID, MPS BATCH PARTITION CANCELLED

Explanation: An MPS batch program issued a DL/I call which did not contain a PCB parameter or contained one which was invalid.

User Response: Correct the error and resubmit the job.

DLZ133I FUNCTION (GSCD|PCB|TERM) IS INVALID, MPS BATCH PARTITION CANCELLED

Explanation: An MPS batch program issued a DL/I call which was not allowed in the environment in which it was operating. MPS jobs using MPS Restart are not allowed to issue PCB or TERM calls. MPS jobs operating in a partition in an address space other than that in which DL/I is operating are not allowed to use GSCD calls.

User Response: For PCB or TERM calls, either remove the calls or remove the use of MPS Restart and resubmit the job.

For GSCD the application logic may be converted to use GSTA, if the call is used to collect information concerning buffer statistics.

Otherwise, either the GSCD call logic will have to be removed or the application will have to be run in a partition in the same address space as DL/I.

DLZ134I MPS [BATCH PARTITION] ENDED ABNORMALLY, SUBSID NOTIFY ERROR, RC=rc

Explanation: The return code *rc* was received when the MPS batch module DLZMPI00 or online module DLZMPC00 issued a SUBSID macro with function NOTIFY.

MPS uses XPCC support for cross-partition communication between the batch and online partitions. Before an IBM subsystem like DL/I does an IDENTIFY, it has to establish its identity via the SUBSID supervisor service.

For an explanation of the SUBSID return code *rc* refer to *VSE/SP Messages and Codes*, Section "VSE/Advanced Functions Return Codes."

User Response: Resubmit the batch job after the cause of the error has been found and corrected.

DLZ135I INSUFFICIENT STORAGE FOR GETMAIN IN id DLZBPC00

Explanation: Insufficient CICS/VS GETMAIN storage in the online partition *id* was available to honor a request from the MPS online batch partition controller (DLZBPC00) for storage to process either a PSB schedule call or an access request for a DL/I data base.

User Response: Resubmit or restart the job when the activity in the system is at a lower level or after restarting the CICS/VS system in a larger partition.

DLZ136I INVALID PSB FOR RESTART, MPS BATCH PARTITION CANCELLED

Explanation: While using MPS restart, a wrong or modified PSB was used. The PSB is different from the PSB used the first time this job was run.

User Response: Use the correct PSB for restart.

DLZ137I TSQ WAS NOT PURGED, TSQ DOES NOT EXIST

Explanation: Transaction CSDP was executed, but the Temporary Storage Queue (TSQ) was not purged because it does not exist.

User Response: None required.

DLZ138I TSQ WAS NOT PURGED, TEMPORARY STORAGE IS NOT AVAILABLE

Explanation: Transaction CSDP was executed, but the Temporary Storage Queue (TSQ) was not purged because the Temporary Storage Program is not present in the current CICS/VS system.

User Response: None required.

DLZ139I TSQ WAS NOT PURGED BECAUSE DL/I IS NOT ACTIVE - IGNORED

Explanation: Transaction CSDP was executed, but the Temporary Storage Queue (TSQ) was not purged because DL/I was not active.

User Response: None required.

DLZ260I TOO MANY PARAMETERS IN A DL/I CALL, DL/I [TASK] TERMINATED

Explanation: The number of parameters (data items named in a COBOL USING list) in the application program CALL to DL/I exceeds the allowable limit. The maximum number of parameters allowed on a CALL to DL/I is 18. The count of parameters (if used) is not counted as one of the 18 parameters.

User Response: Verify that the parameter list contains a valid parameter count or the end of the parameter list is properly indicated.

DLZ261I INVALID VALUE PASSED IN A DL/I CALL, DL/I [TASK] TERMINATED

Explanation: One of the values passed in the application program CALL to DL/I is invalid. It either exceeds machine size, does not meet alignment requirements, or violates storage protection boundaries.

User Response: Verify that the count field value (if used) is valid and that the call function is valid. See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ262I UNABLE TO ASSIGN DMB TO DL/I BUFFER POOL, DL/I TERMINATED

Explanation: While attempting to allocate buffers to the DL/I buffer pool, one or more pointers were destroyed.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ263I INVALID DMB *dmbname* IN PSB SECONDARY LIST, DL/I TERMINATED

Explanation: While attempting to relocate the PSB secondary list, an invalid DMB reference was encountered.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ264I CICS/VS JOURNAL FILE WAS NOT OPENED, DL/I TERMINATED

Explanation: The CICS/VS journal was not opened. The DL/I system is terminated.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ265I ERROR WHILE WRITING TO CICS/VS JOURNAL FILE, DL/I TERMINATED

Explanation: An error (physical or VSAM logical) occurred while writing to the CICS/VS journal file. The DL/I system is terminated.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ266I INVALID TYPE CODE IN DMB, DL/I TERMINATED

Explanation: DL/I came across an invalid type code in the secondary list of the DMB. It may be a bad copy of the DMB, a user error, or a system error.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are

found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ267I BAD FUNCTION CODE TO DL/I QUEUEING FACILITY, DL/I TERMINATED

Explanation: Module DLZQUEF0 (queueing facility) was called with an unrecognizable function code in the PSTFNCTN field of the PST (program specification table).

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ268I INVALID LOGICAL CHILD SEGMENT ON INSERT, DL/I TERMINATED

Explanation: During insert processing, a logical child segment was encountered with both the logical and physical delete flags set on. This is an invalid condition.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ280I DL/I STATISTICS REPORT COMPLETE

Explanation: This message is issued when the transaction completes.

User Response: None required.

DLZ281I NO DL/I STATISTICS AVAILABLE

Explanation: DL/I is not active in this invocation of CICS/VS.

User Response: None required.

DLZ282I ERROR WRITING TO CICS DESTINATION CSSL - xxxxxx

Explanation: This message is issued when CICS/VS returns a non-zero return code from the DFHTD macro after attempting to write to the CSSL data set.

xxxxxxx Explanation

IDERROR The DFHTD TYPE=PUT macro took the IDERROR exit.

IOERROR The DFHTD TYPE=PUT macro took the IOERROR exit.

NOSPACE The DFHTD TYPE=PUT macro took the NOSPACE exit.

NOTOPEN The DFHTD TYPE=PUT macro took the NOTOPEN exit.

XXXXXXX The DFHTD TYPE=PUT macro did not take any of the above exits, and did not take the NORESP (normal) exit either.

System Action: A dump is taken using dump code D282. The transaction is terminated.

Operator Response: The terminal operator should contact the systems programmer. If this message occurs frequently, the system operator should notify the terminal operator not to run the transaction until the problem is corrected.

Programmer Response: If the reason code is:

xxxxxxx	Programmer Response
IDERROR	Check the definition of the CSSL in the CICS/VS destination control table (DFHDCT). The CSSL must be defined as either TYPE=EXTRA or TYPE=INDIRECT. If the CSSL is defined TYPE=INDIRECT, the indirect destination must be defined as TYPE=EXTRA.
IOERROR	Locate and correct the cause of the IOERROR.
NOSPACE	Change the definition of the TYPE=SDSCI data set from DASD to a printer, enlarge the DASD extent used for the data set, or enlarge the VSAM space used if the data set resides in VSAM managed space.
NOTOPEN	Check the definition of the CSSL in the CICS/VS destination control table (DFHDCT). If the CSSL is to be opened at initialization time, it must be resident and related to its own TYPE=SDSCI macro. If the open for the CSSL is to be deferred, then RESIDENT=NO can be declared and a TYPE=SDSCI definition shared with other destinations. However, an SDSCI can be opened for only one destination at a time. The command: CSMT OPEN/CLOSE,TRANSD,DSTID=xxxx can be used to open and/or close destinations dynamically.
XXXXXXX	Check the dump to determine the return code from the DFHTD macro. Refer to the <i>CICS/DOS/VS Problem Determination Guide</i> for an explanation of that return code.

DLZ3011 ERROR - UNABLE TO OPEN FILE - file name [VSAM EC=xx]

Explanation: The issuing utility program was unable to open a data set with the file name specified in the message. Additionally, the VSAM error code is printed for VSAM files (DL/I data bases). For an explanation of the VSAM error code xx refer to *VSE/SP Messages and Codes*, Section "VSE/VSAM Return and Error Codes."

User Response: Ensure that all ASSGN statements required for execution are supplied. Correct any missing job control statements and resubmit the job.

DLZ3021 ERROR - STATEMENT COLUMN 1 NOT {S|D|R|L}

Explanation: A utility control statement was encountered with an incorrect control code in column 1.

User Response: Correct the control statement, depending on the utility program being executed, and resubmit the job.

DLZ3031 ERROR - STATEMENT COLUMN 2 NOT A 1 OR 2

Explanation: A utility program encountered a control card with an incorrect copy request specification.

User Response: Correct control card, and resubmit the job.

DLZ3041 ERROR - NO DBD NAME SUPPLIED

Explanation: A utility program encountered a control card with the DBD name field blank.

User Response: Correct control card and resubmit the job.

DLZ3051 ERROR - DMB NAME NOT FOUND

Explanation: A utility program attempted to find a DMB corresponding to the DBD name specified in the utility control card in any of the libraries defined in the search chain. The DMB was not found.

User Response: Correct DBD name in the utility control statement and resubmit the job.

DLZ3061 ERROR - FILENAME file name NOT FOUND IN {DMB|DBD}

Explanation: A utility program attempted to locate the FILENAME specified in the control card in the DBD or DMB. DMB name was obtained from the DBD specified in the control card. The attempt was unsuccessful.

User Response: Correct the control card in error, and resubmit the job.

DLZ3071 ERROR - NO DATA SET FILENAME SUPPLIED

Explanation: A utility program found a control card which required a file name in card column 13 and the field was blank.

User Response: Correct the control card, and resubmit the job.

DLZ308I	ERROR - NO OUTPUT FILENAME SUPPLIED Explanation: A utility program found a control card which required an output file name in card column 22 and the field was blank. User Response: Correct the control card, and resubmit job.	DLZ314I	ERROR - ACCUMULATION FILE SUPPLIED BUT NO DUMP SUPPLIED Explanation: An accumulated change input data set was supplied to the data base data set recovery utility, but a dump input data set was not assigned. An accumulated change input may not be supplied without a dump input. User Response: Correct the job control statements and resubmit the job.
DLZ309I	ERROR - CONTROL STATEMENT ERRORS DETECTED Explanation: A utility program found one or more control card errors. (The particular errors have been described by messages preceding this one.) User Response: Examine the output listing, correct the errors, and resubmit the job.	DLZ315I	RESTART NUMBER EXCEEDS LAST CHECKPOINT NUMBER-- nnnn Explanation: The HD reorganization unload utility or the HD reorganization reload utility attempted a checkpoint restart, and the console operator entered a restart number which exceeded the last checkpoint record on the unloaded file. The four-digit decimal <i>nnnn</i> is the number of the last checkpoint record on the unloaded file. User Response: Resubmit the job and enter the correct restart number when responding to message DLZ318A.
DLZ310I	STATEMENT IN ERROR - <i>statement</i> Explanation: This message usually follows DLZ309I. The control card in error is printed as the second line to identify the particular card in error. User Response: Correct any errors, and resubmit the job.	DLZ316I	ERROR - DATA BASE ORGANIZATION UNRECOGNIZABLE Explanation: The DMB for the specified DBD did not contain a valid data base organization code. User Response: Ensure that the DMB for the specified DBD was created by the DL/I application control blocks creation and maintenance utility and link-edited into a library defined in the search chain. Correct any discrepancies and resubmit the job.
DLZ311I	<i>file name</i> FILE NOT ASSIGNED OR DEVICE INVALID Explanation: The specified data set was either not assigned or was assigned to an invalid device type. The file name may be incorrect. If <i>file name</i> =TRACEIN, the input file to the Trace Print Utility is involved. The real file name in this case is TAPEIN or DISKIN. User Response: Correct the control card and/or job control statements, and resubmit the job.	DLZ317I	ERROR - OUTPUT FILENAME SUPPLIED NOT FOUND ON DUMP HEADER Explanation: A dump input was supplied to the data base data set recovery utility that was for the correct DBD but not for the file name specified. User Response: Correct the input and resubmit the job.
DLZ312I	ERROR - DUMP INPUT FILE IS NOT A VALID IMAGE COPY OR HISAM UNLOAD FILE Explanation: An input data set defined by the DUMPIN file was supplied to the data base data set recovery utility with a header record of the incorrect format. User Response: Correct the input and resubmit the job.	DLZ318A	ENTER RESTART NUMBER - NNNN Explanation: The HD reorganization unload utility or the HD reorganization reload utility is processing a checkpoint restart and requires the restart number as a four-digit decimal number between 0001 and 9999. User Response: Enter the number of a valid checkpoint record on the unloaded file (usually the last checkpoint number). The number may be obtained either from message DLZ381I or DLZ315I.
DLZ313I	DUMP INPUT FILE IS NOT FOR A DATA SET IN THE DATA BASE TO BE RECOVERED Explanation: An input data set defined by the DUMPIN file was supplied to the data base data set recovery utility with a header record indicating that the data set was not defined in the DBD. User Response: Correct the input and resubmit the job.		

DLZ319I	ERROR - I/O ERROR ON FILE NAME - <i>file name</i>	<p>User Response: If any change records were created between the subject dates, and are required for data base integrity, the accumulated change input must be recreated and the recovery job submitted again.</p>
	<p>Explanation: An I/O error has occurred on the specified file name.</p> <p>User Response: Determine the cause of the error and take the appropriate action to rerun the job.</p>	
DLZ320I	2 COPIES SPECIFIED - WILL CONTINUE WITH ONE	<p>DLZ326I WARNING - ACCUMULATION FILE CREATION DATE EARLIER THAN DUMP CREATION DATE, ACCUMULATION INPUT FILE IGNORED</p>
	<p>Explanation: A utility program encountered an I/O error on an output volume. Two copies were specified, so the program continues with the alternate copy.</p> <p>User Response: None required.</p>	<p>Explanation: The data base data set recovery utility found that the creation date of the accumulated change input file to be earlier than the creation date of the image dump file. Since all records are already on the dump, the accumulated change input file is not being used.</p> <p>User Response: None required.</p>
DLZ321I	ERROR - OPERATION TERMINATED - WILL TRY NEXT CONTROL CARD	<p>DLZ327I WARNING - NO RECORDS ON ACCUMULATION FILE FOR RECOVERED DATA SET</p>
	<p>Explanation: A utility program encountered errors prohibiting the successful completion of the requested operation. The program attempts to obtain another control card.</p> <p>User Response: None required.</p>	<p>Explanation: The data base data set recovery utility scanned the accumulated change input file for records applicable to the file to be recovered and none were found.</p> <p>User Response: Ensure that the accumulated change input was correct. If not, correct the input and resubmit the job.</p>
DLZ322I	WARNING - NO DUMP INPUT SUPPLIED	<p>DLZ328I WARNING - NO RECORDS ON LOG FOR RECOVERED DATA SET</p>
	<p>Explanation: The data base data set recovery utility found the image copy input file was not assigned.</p> <p>User Response: If none was required, no action is necessary. Otherwise, correct the input and resubmit.</p>	<p>Explanation: The data base data set recovery utility scanned all input log file records for a record applicable to the file to be recovered. None were found.</p> <p>User Response: Ensure that all system log files were submitted. If any were left out, correct the job control statements, and resubmit the job.</p>
DLZ323I	WARNING - NO LOG INPUT SUPPLIED	<p>DLZ329I CHKPT PARAMETER INVALID - 5000 ASSUMED</p>
	<p>Explanation: The data base data set recovery utility found that the log input file was not assigned.</p> <p>User Response: If none was required, no action is necessary. Otherwise, correct the input and resubmit.</p>	<p>Explanation: The checkpoint option must be specified as <code>CHKPT=<i>n</i></code> or <code>CHKPT=NO</code>. <i>n</i> must be a one to six digit decimal number representing the number of segments desired between checkpoints.</p> <p>User Response: The job will run with a default of 5000. If another value is desired, cancel the job, correct the control statement, and resubmit the job.</p>
DLZ324I	WARNING - NO ACCUMULATION INPUT SUPPLIED	<p>DLZ330I ERROR - RELATIVE BLOCK NUMBERS OF ACCUMULATION FILE RECORD NOT IN ASCENDING ORDER</p>
	<p>Explanation: The data base data set recovery utility found that the accumulated change input file was not assigned.</p> <p>User Response: If none was required, no action is necessary. Otherwise, correct the input and resubmit.</p>	<p>Explanation: The data base data set recovery utility encountered an ESDS record in the accumulated change input file whose relative block number (HD) or RBA (HISAM) was lower than the last record update.</p> <p>User Response: Recreate the accumulated change input file and resubmit the job.</p>

DLZ331I	<p>ERROR - ACCUMULATION FILE RECORD KEYS NOT IN ASCENDING SEQUENCE</p> <p>Explanation: The data base data set recovery utility encountered a KSDS record in the accumulated change input file whose key was lower than the last record updated.</p> <p>User Response: Recreate the accumulated change input file and resubmit the job.</p>	DLZ337I	<p>ERROR - INSUFFICIENT SPACE IN BUFFER POOL</p> <p>Explanation: The data base data set recovery utility requested buffer pool space, but none was available. This is a DL/I logic error.</p> <p>User Response: Ensure that the DL/I system and all control statements are valid and resubmit the job. If the error persists, contact the IBM Support Center.</p>
DLZ332I	<p>ERROR - ESDS RECORD TO BE ADDED IS MORE THAN 1 GREATER THAN LAST RECORD</p> <p>Explanation: The data base data set recovery utility encountered an ESDS record in the accumulated change or log input file with a VSAM RBA value indicating that it was more than one greater than the last existing record.</p> <p>User Response: Ensure that all supplied input files are correct and resubmit the job.</p>	DLZ338I	<p>ERROR - INSUFFICIENT SPACE IN DATA SET</p> <p>Explanation: The data base data set recovery utility attempted to add a record to the recovered file. The file had used all available extent space allocated.</p> <p>User Response: If the data base is of HISAM organization, increase the space allocation by executing the VSAM AMS utility DEFINE command. If the data base is of HD organization, ensure that input data was correct, and resubmit the job.</p>
DLZ333I	<p>ERROR - DUPLICATE KEY ON LOG AND INSERT</p> <p>Explanation: The data base data set recovery utility encountered a system log input simple HISAM KSDS record with a key that exists in the recovered file and the log record code was not a physical delete (VSAM erase) or replace. This indicates that the input supplied was not correct.</p> <p>User Response: Ensure that all input files are correct, all log tapes are in the proper sequence, and resubmit the job.</p>	DLZ339I	<p>NO ERRORS DETECTED, <i>utility</i> SUCCESSFUL</p> <p>Explanation: The specified utility was successfully completed.</p> <p>User Response: None required.</p>
DLZ334I	<p>ERROR - LOG DATES INDICATE LOGS OUT OF SEQUENCE</p> <p>Explanation: The data base data set recovery utility program encountered a system log input record with a date lower than the highest date previously encountered.</p> <p>User Response: If multiple logs were submitted as input, resubmit the job, and ensure the log tapes are submitted in ascending sequence.</p>	DLZ340I	<p>ERROR - SPECIFIED FILE <i>file name</i> ESDS IS EMPTY</p> <p>Explanation: The VSAM ESDS specified to be dumped by Image Copy was empty.</p> <p>User Response: None required.</p>
DLZ335I	<p>ERROR - NO DUMP OR LOG INPUT SPECIFIED</p> <p>Explanation: The data base data set recovery utility found that neither the dump input (image copy or HISAM unload) file nor the log input file was assigned.</p> <p>User Response: Ensure that the correct input files are assigned and resubmit the job.</p>	DLZ341I	<p>CATASTROPHIC ERROR IN DL/I</p> <p>Explanation: The data base data set recovery utility requested DL/I to perform an operation on the file being recovered. This message indicates that the requested operation cannot be performed.</p> <p>User Response: Ensure that the DL/I system and all control statements are valid and resubmit the job. If the error persists, save dump and console log and contact the IBM Support Center.</p>
DLZ336I	<p>ERROR - PERMANENT READ ERROR ON RECOVERED DATA SET</p> <p>Explanation: The data base data set recovery utility encountered a permanent I/O error in the data set being recovered.</p> <p>User Response: Initialize the DASD volume, and resubmit the job.</p>	DLZ342I	<p>NUMBER OF LOG INPUTS INVALID</p> <p>Explanation: The number of log files in the utility control statement is invalid.</p> <p>User Response: Correct the utility control statement or provide the correct assignment and resubmit the job.</p>
DLZ336I	<p>ERROR - PERMANENT READ ERROR ON RECOVERED DATA SET</p> <p>Explanation: The data base data set recovery utility encountered a permanent I/O error in the data set being recovered.</p> <p>User Response: Initialize the DASD volume, and resubmit the job.</p>	DLZ343I	<p>ERROR - ACCUMULATION INPUT FILE SPECIFIED FOR SHISAM RECOVERY</p> <p>Explanation: The data base data set recovery utility found an accumulated change input file assigned, and the data base to be recovered was simple HISAM. Accumulated change utility files are not allowed as input when recovering simple HISAM data bases.</p>

User Response: Resubmit the job with dump input, log input or dump and log input.

DLZ344I ERROR - CARD COLUMN 3 NOT N, R, U, OR BLANK FOR REWIND

Explanation: A utility program encountered a control statement with an incorrect rewind option.

Column 3 must contain:

- U or a blank for rewind and unload,
- N for no rewind, or
- R for rewind only.

User Response: Correct the error in the control statement and rerun the job.

DLZ345I WARNING - file name NOT ASSIGNED, ONE COPY CREATED

Explanation: The utility program found the secondary output was not assigned. One output copy will be created.

User Response: If a second copy is required, correct job control statement and resubmit the job. Otherwise, no action is necessary.

DLZ346I ERROR - NO ALTERNATE UNIT AVAILABLE

Explanation: The HD reorganization utility encountered a permanent I/O error on an output volume and no alternate output unit was available. If two copies were requested, one copy had already terminated because of an I/O error.

User Response: Ensure output units and volumes are in good working order, and resubmit the job.

DLZ348I ERROR - UNEXPECTED STATUS CODE xx FROM DL/I CALL

Explanation: The HD reorganization utility encountered an unexpected return status after a DL/I call. The xx is replaced by the actual status received.

User Response: Refer to Chapter 6, "DL/I Status Codes" on page 111 in this manual for the meaning of the returned status. Correct the error and resubmit the job.

DLZ349I ERROR - PERMANENT READ ERROR ON FILENAME - file name

Explanation: A permanent read error has been encountered on the file name specified.

User Response: Resubmit the job. If the error persists, execute the data base data set recovery utility to recover the named file.

DLZ350I ERROR - A ZEROED VSAM CONTROL INTERVAL WAS FOUND

Explanation: The data set image dump utility encountered a zeroed control interval. A previous error has zeroed the control interval on the data base.

User Response: If the zeroed control interval does not make sense, reorganize the data base to remove the zeroed control interval.

DLZ351I ERROR - INVALID CONTROL STATEMENT

Explanation: The utility has encountered an invalid control statement. A valid control statement for the HD RELOAD utility must contain REW= or BLDINDEX=NO starting in column one.

User Response: Correct the error in the control statement and rerun the job.

DLZ353I ERROR - UNKNOWN RECORD TYPE ON INPUT FILE

Explanation: The HISAM reorganization reload utility found the input file record was not a valid record type.

User Response: Ensure the input was correct. If it was, the file must be recreated by the HISAM reorganization unload utility and the job resubmitted.

DLZ356I ERROR - NO HEADER RECORD FOUND ON FILENAME - file name

Explanation: The HISAM reorganization reload utility found the input file, described by the file name listed, was not a valid unloaded HISAM data base.

User Response: Ensure the input was correct. If it was the file must be recreated by the HISAM reorganization unload utility and the job resubmitted.

DLZ357I ERROR - COPY REQUEST INCOMPATIBLE

Explanation: A control card was encountered that requested two copies and only one output file name was assigned; or one copy and two output file names were assigned.

User Response: Make necessary corrections to the control card or job control statements and resubmit the job.

DLZ358I ERROR - INVALID SEGMENT CODE ENCOUNTERED

Explanation: In attempting to reorganize a HISAM data base, a segment was encountered with a physical segment code of zero, or a segment code higher than the highest segment code in the associated DBD.

User Response: Ensure that the supplied DBD accurately describes the data base. If the data integrity of the data base is questionable, execute

the data base data set recovery utility for the data base in question, and then resubmit this job.

DLZ359I ERROR - INVALID CONTROL STATEMENT

Explanation: The utility encountered an invalid control statement. A valid control statement for the utility must contain REW= or CHKPT= starting in column one.

User Response: Correct the error in the control statement and rerun the job.

DLZ360I ERROR - ID CARD ERROR AT OR NEAR COLUMN - nn

Explanation: The change accumulation utility encountered one of the following card errors:

- The control card was not a DB or ID card.
- Columns 31 to 33 specified a maximum key length of zero or greater than 236.
- Columns 50 and 51 specified the number of input log files as zero.
- The field was not numeric.

User Response: Correct the control card, and resubmit the job.

DLZ361I ERROR - DB CARD ERROR ENCOUNTERED

Explanation: The change accumulation utility encountered one of the following control card errors on a DB card:

- DB card expected but not found.
- The card was not DB0 or DB1.
- Column 4 is blank, but *ALL or *OTHER not specified.
- Column 11 is not blank.
- Card columns 12 to 20 not numeric.
- The DDD value exceeds 366.
- The HH value exceeds 23.
- The MM value exceeds 59.
- Purge date/time value exceeds current execution date/time.

User Response: Correct the control card, and resubmit the job.

DLZ363I ERROR - *ALL SPECIFIED WITH OTHER DB CARDS

Explanation: If *ALL is specified on a DB statement, no other DB statements of the same type (DB0 or DB1) may be used.

User Response: Correct control cards, and resubmit the job.

DLZ364I ERROR - A DUPLICATE DATA BASE NAME HAS BEEN ENCOUNTERED

Explanation: The same DBD name was specified on more than one DB0 or DB1 statement.

User Response: Correct the control card, and resubmit the job.

DLZ365I ERROR - TOO MANY DATA BASES

Explanation: The number of data bases to be processed as identified in columns 11 to 13 of the ID statement (default = 16) was less than the number encountered on subsequent DB0 and DB1 statements.

User Response: If an ID control statement was supplied, increase the maximum DBD name specification. If not, supply one, and resubmit the job.

DLZ366I ERROR - WRITE ERROR ON RECOVERED DATA SET

Explanation: The data base data set recovery utility came across permanent I/O error in the data set being recovered.

User Response: Initialize the DASD volume and resubmit the job.

DLZ367I ERROR - *OTHER SPECIFICATION INVALID

Explanation: The change accumulation utility has encountered more than one DB control statement with *OTHER specified, and:

- *ALL or *OTHER was specified on a previous DB card.
- No DB0 cards exist and *OTHER was specified on a DB1 card.
- No DB1 card exists and *OTHER was specified on a DB0 card.

User Response: Correct the control cards, and resubmit the job.

DLZ368I ERROR - NOT AN N, R, OR U IN COLUMN 5

Explanation: The utility encountered an invalid REW parameter in column 5 of the control statement. Column 5 must be an N for no rewind, an R for rewind only, or a U for rewind and unload.

User Response: Correct the error in the control statement and rerun the job.

DLZ369I ERROR - DMB NOT FOUND. SPECIFIED DBD NAME = dbdname

Explanation: DBD names were specified on DB control statements, but the DMB corresponding to the DBD name listed could not be located.

User Response: Ensure that the control statements are correct, and that all DMBs for the corresponding DBDs reside in a library defined in the search chain. Then resubmit the job.

DLZ370I RESTART NUMBER LESS THAN FIRST CHECKPOINT NUMBER - nnnn

Explanation: The HD reorganization reload utility attempted a checkpoint restart, and the console operator entered a checkpoint number smaller than the first checkpoint record on the unloaded file. The number nnnn is *first* checkpoint record on the unloaded file.

User Response: Resubmit the job, making certain the correct unloaded file volume is used and the correct restart number is entered.

DLZ371I ERROR - A RECORD HAS BEEN ENCOUNTERED WITH A KEY LONGER THAN SPECIFIED

Explanation: An input record has been encountered with a key longer than was specified on an ID control statement or, if no ID statement was supplied, 10 bytes.

User Response: Supply an ID control statement with a correct key length specification, and resubmit the job.

DLZ372I ERROR - INVALID LOG BUFFER SIZE

Explanation: The job terminated abnormally because the log buffer size parameter specified in the LI control statement was invalid for one of the following reasons:

- The value was longer than five digits.
- The value contained other than numeric characters.
- The column immediately following the value was not blank.
- The value did not begin in column 31.
- The value was greater than 32767.
- The value was less than 1100 for a CICS/VS journal or less than 1024 for a DL/I log.

User Response: Correct the log buffer size parameter in the LI control statement and resubmit the job.

DLZ373I INVALID ACCUMULATION INPUT FILE

Explanation: One of the following conditions was found on the input accumulated file:

- No records existed.
- First record not a valid header record.
- An unidentifiable record type was read.

User Response: Correct inputs, and resubmit the job.

DLZ374I WARNING - NO CHANGE RECORDS FOUND ON LOG(S)

Explanation: The log input supplied did not contain any records with a log record code of hexadecimal '50', or all records were purged.

User Response: Ensure that inputs are correct. If not, correct inputs, and resubmit the job.

DLZ375I ERROR - A HEADER RECORD MISSING CONDITION HAS BEEN DETECTED ON {CUMIN|LOGIN}

Explanation: A header record on either the input accumulation file or the input log file is missing, or a header record was lost in the sort phase.

User Response: Ensure that all inputs are valid, and resubmit the job.

DLZ376I INVALID DEVICE ASSIGNED FOR RESTART FILE - SYS010

Explanation: The HD reorganization reload utility attempted a checkpoint restart and found that SYS010 was assigned to a device that is invalid for the partially created work file from the unsuccessful run.

User Response: Resubmit the job with SYS010 assigned to a valid device for the partially created work file or with SYS010 unassigned.

DLZ377I ERROR - RESTART INPUT INVALID

Explanation: The HD reorganization unload utility was supplied a restart file that was not of proper format.

User Response: If a restart was not required, IGN should be specified for the restart file. If a restart was required, ensure that the restart assign card is correct and resubmit the job.

DLZ378I WARNING - INVALID CICS JOURNAL PROCESSING PARAMETER. 'C' ASSUMED

Explanation: The request for accumulating a second CICS/VS journal extent is invalid because column 4 of the ID-statement is neither blank nor 'C'.

DL/I assumes 'C' (for processing the second CICS/VS journal extent) and continues processing.

User Response: This is a warning message, only.

DLZ380I ERROR - UNABLE TO POSITION DATA BASE CHECKPOINT SEGMENT NOT FOUND

Explanation: The HD reorganization unload utility or the HD reorganization reload utility was attempting a checkpoint restart and to position the data base to the root segment described in the checkpoint record. The segment did not exist.

User Response: Ensure all job control statements are correct and the data base has not been changed between the original unload/reload and the restart attempt. If corrections can be made, resubmit the job. Otherwise, restart the job from the beginning.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ381I CHKPNTnnnn, {vol1|***}, {vol2|*****}, segmname**

Explanation: This message is written to SYSLOG as an information message every time a checkpoint is taken by the HD reorganization unload utility or the HD reorganization reload utility. The *nnnn* is a four-digit decimal value between 0001 and 9999 incremented by 1 for every checkpoint taken. The *vol1* parameter is the current primary output volume serial number (or asterisks if DASD). The *vol2* parameter is the

current secondary output volume serial number (or asterisks if the secondary output is not in use or if DASD). The *segmname* field is the segment name of the segment being retrieved when the checkpoint was taken.

User Response: No action required.

DLZ382I ERROR - DBD SUPPLIED IS NOT A VALID DBD

Explanation: The dbdname supplied on a control card was used to load a DBD. Upon examination of the DBD it was discovered that the DBD was not a valid DBD.

User Response: Either correct the control card or, if the control card is correct, replace the DBD with a valid DBD.

DLZ383I ERROR - INPUT RECORDS OUT OF SEQUENCE

Explanation: The HISAM reorganization unload utility has come across a root segment with a sequence field (key) equal to or lower than the previous root segment processed. The execution is terminated, and a storage dump is produced. Register 11 has the address of the current key and register 9 has the address of the previous key.

User Response: Correct error and resubmit the job.

DLZ384I ADDITIONAL DIAGNOSTICS ON SYSLST

Explanation: Additional diagnostic messages were written on SYSLST. If no additional messages were written, a failure occurred in SORT/MERGE.

User Response: Correct the errors and resubmit the job.

DLZ385I utility UNSUCCESSFUL

Explanation: An error was detected that kept the indicated utility from completing successfully.

User Response: Correct the errors and resubmit the job.

DLZ386I ERROR - DUPLICATE CONTROL CARD

Explanation: The utility has encountered a duplicate control statement for the checkpoint or rewind option.

User Response: Correct the control statement and resubmit the job.

DLZ387I ERROR - NO VALID RECORDS ON INPUT DATA SET

Explanation: The input file to the HD reorganization reload utility did not contain any valid unloaded data base records. The return code is 16.

User Response: Correct the invalid input, and resubmit the job.

DLZ388I NO RECORDS WRITTEN TO WORK FILE WORKFIL

Explanation: Reloaded data base had no active logical relationships.

User Response: None.

DLZ389I ERROR - STATISTICS RECORD NOT FOUND

Explanation: A statistics record was not found as the first record on input to HD reorganization reload utility.

User Response: Check the job control statements for the correct input file. Correct and resubmit the job.

DLZ390I ERROR - INVALID LOGICAL RECORD ENCOUNTERED

Explanation: During input processing of log tapes, a spanned record segment was encountered which did not contain a valid segment descriptor word (SDW).

User Response: Correct log tape input and resubmit the job.

DLZ391I ERROR OCCURRED EXECUTING {GETVIS|FREEVIS} MACRO, RETURN CODE=*rc*

Explanation: The utility program encountered an error executing either a GETVIS or FREEVIS macro request. The program terminated. A storage dump is produced if it was a FREEVIS error. The GETVIS or FREEVIS return code *rc* is given in decimal format.

For an explanation of the GETVIS or FREEVIS return code *rc* refer to *VSE/SP Messages and Codes*, Section "VSE/Advanced Functions Return Codes."

User Response: If GETVIS, correct the SIZE parameter in the EXEC statement and reexecute the program.

If FREEVIS, save log and dump and contact the IBM Support Center.

DLZ392I ERROR EXECUTING VSAM CONTROL BLOCK MACRO - PROGRAM TERMINATED

Explanation: The utility program encountered an error executing a VSAM control block generation macro request. The program terminated abnormally and produced a storage dump. The macro return code is indicated in register 15. Register 14 contains the address of the instruction following the BAL 14,15 instruction in the macro instruction.

User Response: Save log and dump and contact the IBM Support Center.

DLZ393I VSAM DEFINE INCOMPATIBLE FOR FILENAME
- *file name*

Explanation: The HISAM reorganization reload utility detected that one of the following fields in the input header record did not match the corresponding VSAM catalog entry for that data set:

- Record size for a KSDS or ESDS data set.
- Control interval size for a KSDS or ESDS data set.
- Key length for a KSDS data set.
- Relative key position for a KSDS data set.

User Response: Check DLBL and EXTENT information and execute the access method services utility DEFINE command for the data base and resubmit the job.

DLZ394I RBA FOR HISAM ESDS NOT SAME AS PRECALCULATED VALUE

Explanation: A HISAM ESDS record was written to an RBA which is not the same as the RBA calculated for this record by the HISAM reorganization unload utility.

User Response: This is a DL/I logic error. Call the IBM Support Center.

DLZ395I BACKOUT COMPLETE FOR PSB *psbname*

Explanation: The data base backout utility has successfully completed the requested data base backout.

User Response: None required.

DLZ396I SYSTEM ERROR DURING BACKOUT OF DATA BASE *dbname PSBNAME psbname*

Explanation: The data base backout utility was unable to locate a DMB directory entry for the data base specified.

User Response: This indicates that the PSB used for backout has changed since the original run. Check the PSB requested in the EXEC card to ensure the name and the PSB are correct. Correct any errors and rerun the job.

DLZ397I BACKOUT UNABLE TO OPEN DATA BASE *dbname PSBNAME psbname*

Explanation: The data base backout utility was unable to open the specified data base.

User Response: Check the job control statements to ensure they are present and identify the data sets properly. Correct any errors and rerun the job.

DLZ398I I/O ERROR DURING BACKOUT OF DATA BASE *dbname FOR PSB psbname*

Explanation: The data base backout utility encountered an I/O error while attempting backout the specified data base.

User Response: Run the data base recovery utility for the data base in error. Use all log tapes, excluding the output of this backout attempt, and then rerun the backout job.

DLZ399I BUFFER POOL TOO SMALL FOR BACKOUT OF DATA BASE *dbname PSBNAME psbname*

Explanation: The data base buffer pool was too small to accommodate the data base specified while attempting to run the data base backout utility. This error should not occur.

User Response: Ensure that the DL/I system and all control statements are valid and resubmit the job.

DLZ400I SEQUENCE ERROR IN DATA BASE

Explanation: While attempting to perform a reorganization unload, an out of sequence segment was encountered.

User Response: Correct the sequence within the data base and resubmit the reorganization unload job. A procedure that can be used to locate (in the dump) the segment that caused the sequence error is described in *DL/I DOS/VS Diagnostic Guide* under the heading "DLZURGU0 - HD Reorganization Unload Utility."

DLZ401I INPUT CONTROL STATEMENT ERROR NEAR COLUMN *nn*

Explanation: A DL/I utility encountered an invalid parameter at or near column *nn* of the input control statement.

User Response: Correct the invalid parameter on the utility control statement and resubmit the job.

DLZ402I ERROR - INVALID LOG ASSIGNMENT

Explanation: The DL/I log file was either not assigned or was assigned to an invalid device type.

User Response: Correct the utility control statement and/or job control statement, and resubmit the job.

DLZ403I WARNING - NUMBER OF UNLOADED RECORDS NOT EQUAL RELOADED

Explanation: A mismatch between the number of unloaded and the number of reloaded records is detected. This is correct and can be ignored if the reload utility is re-started. In this case, the number of segments reloaded includes only those that were reloaded since restart.

User Response: Ensure that the job and/or the input for the reload utility is correct and rerun the

job if the number of unloaded and reloaded segments should be equal.

DLZ404I	<p>ERROR - UNABLE TO OPEN FILENAME - <i>file name</i> VSAM CODE=xyyy</p> <p>Explanation: A DL/I utility was unable to open the VSAM data set with the file name specified in the message.</p> <p>For an explanation of the VSAM return code <i>xx</i> and the VSAM error code <i>yy</i> refer to <i>VSE/SP Messages and Codes</i>, Section "VSE/VSAM Return and Error Codes."</p> <p>User Response: Ensure that all utility control statements and job control statements are correct and resubmit the job.</p>
----------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

DLZ405I	<p>ERROR - UNABLE TO CLOSE FILENAME - <i>file name</i> VSAM CODE=xyyy</p> <p>Explanation: A DL/I utility was unable to close the VSAM data set with the file name specified in the message.</p> <p>For an explanation of the VSAM return code <i>xx</i> and the VSAM error code <i>yy</i> refer to <i>VSE/SP Messages and Codes</i>, Section "VSE/VSAM Return and Error Codes."</p> <p>User Response: Take the appropriate action for the VSAM code and resubmit the job.</p>
----------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

DLZ406I	<p>ERROR ACCESSING FILE <i>file name</i>, VSAM CODE=xyyy</p> <p>Explanation: A DL/I utility encountered an I/O error on the data set with the file name specified in the message.</p> <p>For an explanation of the VSAM return code <i>xx</i> and the VSAM error code <i>yy</i> refer to <i>VSE/SP Messages and Codes</i>, Section "VSE/VSAM Return and Error Codes."</p> <p>The following applies for DLZBACK0 (DB backout) and DLZLOGP0 (log print):</p> <ul style="list-style-type: none"> • If a log tape or disk is being processed, an incorrect length is indicated. • If a CICS/VS journal is being processed, the buffer size in the LI statement for the utility must be at least as big as the CICS/VS journal buffer size (1100 bytes minimum). <p>User Response: Take the appropriate action for the type of error indicated and resubmit the job.</p>
----------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

DLZ407I	<p>ERROR - INVALID INPUT CONTROL STATEMENT</p> <p>Explanation: The input control statement supplied to the utility did not begin with valid characters in column 1. TI and TO are the only valid statements if the Trace Print Utility (DLZPRT0) is being executed. Also, column 3 of the TO statement must be blank.</p> <p>User Response: Correct the utility control statement and resubmit the job.</p>
----------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

DLZ408I BACKOUT COMPLETE FOR PSB *psbname* **TO CHECKPOINT** *id*

Explanation: The data base backout utility has successfully completed the requested data base backout to the identified checkpoint *id*.

User Response: Record the checkpoint id and the volume serial number if restart processing must be done.

DLZ409I ERROR - INVALID PSBNAME

Explanation: The PSB name specified in the LO control statement is invalid for one of the following reasons:

- The name contained other than alphanumeric characters.
- The name was longer than 7 characters.
- The name did not begin in column 4.
- The columns immediately following the name were not blank.

User Response: Correct the PSBname and resubmit the job.

DLZ410I ERROR - INVALID START DATE

Explanation: The start date specified in the LO control statement is invalid for one of the following reasons:

- The date contained other than numeric characters.
- The date did not contain 9 numeric characters.
- The date did not begin in column 13.
- The date exceeded the maximum allowable date.

User Response: Correct the start date and resubmit the job.

DLZ411I ERROR - INVALID END DATE

Explanation: The end date specified in the LO control statement is invalid for one of the following reasons:

- The date contained other than numeric characters.
- The date did not contain 9 numeric characters.
- The date did not begin in column 23.
- The date exceeded the maximum allowable date.

User Response: Correct the end date and resubmit the job.

DLZ412I ERROR - START DATE EXCEEDS END DATE

Explanation: The start date specified in the LO control statement is a later date than the end date specified in the LO control statement.

User Response: Correct the start date and/or end date and resubmit the job.

DLZ413I ERROR - INVALID OUTPUT FORMAT SPECIFICATION

Explanation: The output format specified in the LO control statement is not KEYWORD or DUMP, or does not begin in column 33.

User Response: Correct the output format and resubmit the job.

DLZ414I ERROR - INVALID LOGICAL UNIT SPECIFICATION

Explanation: The logical unit specified in the LI or TI control statement is invalid for one of the following reasons:

- Columns 11 to 13 do not contain the characters SYS.
- Columns 14 to 16 do not contain valid numerics.
- Column 17 is not blank.
- The *nnn* of SYS*nnn* is greater than 255.

User Response: Correct the logical unit specified and resubmit the job.

DLZ415I ERROR - INVALID LOG FILE TYPE

Explanation: The log file type specified in the LI control statement is invalid for one of the following reasons:

- Column 21 was not blank or did not contain the characters L, U, V, S or C.
- Column 22 was not blank.

User Response: Correct the log file type and resubmit the job.

DLZ416I NO INPUT CONTROL STATEMENTS - DEFAULTS ASSUMED

Explanation: No input control statements were encountered by the log print utility. Utility defaults are assumed and processing continues.

User Response: No action required. If utility defaults are not desired, supply the appropriate input control statements and resubmit the job.

DLZ417I END OF FILE ON FILENAME - *file name*

Explanation: The log print utility detected an end-of-file condition on the file name indicated. Normal processing continues.

User Response: None.

DLZ418I TOTAL FILE RECORDS READ=*nnnnnnnn*, RECORDS PRINTED=*nnnnnnnn*

Explanation: The log print utility completed processing on the current file. The number of log records read from that file and the number of log records printed for that file are shown in the message text. Normal processing continues.

User Response: None.

DLZ419I ERROR - PHYSICAL I/O ERROR ON FILE *file name* VSAM CODE=*xxyy*

Explanation: A DL/I utility came across a VSAM physical I/O error on the data set with the file name specified in the message.

For an explanation of the VSAM return code *xx* and the VSAM error code *yy* refer to *VSE/SP Messages and Codes*, Section "VSE/VSAM Return and Error Codes."

User Response: Take the appropriate action for the VSAM error indicated and resubmit the job.

DLZ420I TOTAL JOB RECORDS READ=*nnnnnnnn*, RECORDS PRINTED=*nnnnnnnn*

Explanation: The log print utility completed processing on this job. The total number of log records read from all log files and the total number of log records printed from all log files are shown in the message text.

User Response: None.

DLZ421I END DATE EXCEEDED ON FILENAME - *file name*

Explanation: The log print utility encountered a data base log whose date is later than the end date specified in the LO input control statement. Processing on this file and any subsequent file is terminated.

User Response: None.

DLZ422I ERROR - LOGICAL I/O ERROR ON FILE *file name* VSAM CODE=*xxyy*

Explanation: A DL/I utility encountered a VSAM logical I/O error on the data set with the file name specified in the message.

For an explanation of the VSAM return code *xx* and the VSAM error code *yy* refer to *VSE/SP Messages and Codes*, Section "VSE/VSAM Return and Error Codes."

User Response: Take the appropriate action for the VSAM error indicated and resubmit the job.

DLZ423I GETVIS FAILED, TERMINATION RECORD VERIFICATION SUPPRESSED

Explanation: The log print utility was unable to obtain sufficient space for internal tables. As a result, a scheduling record without a termination record may go undetected. This condition is normally identified by messages DLZ424I and DLZ425I.

User Response: Resubmit the job so that it runs in a larger partition.

DLZ424I NO LOG TERMINATION RECORD FOUND FOR SCHEDULING RECORD

Explanation: The log print utility encountered a scheduling record without a matching termination record on the current log file, indicating a possible system failure occurred during creation of the log file. This message may also be written if the end date specified on the LO input control statement is exceeded before a matching termination record is found. This message is followed by message DLZ425I.

User Response: None.

DLZ425I PSBNAME=*psbname*, SEQUENCE NUMBER=*nnnnnnnn*

Explanation: The log print utility did not find a termination record to match the scheduling record whose PSB name is *psbname* and whose sequence number is *nnnnnnnn*. This message follows message DLZ424I.

User Response: None.

DLZ426I ERROR - INVALID LOG COPY PARAMETER SPECIFIED

Explanation: The log copy parameter specified in the LO control statement was not COPY or did not begin in column 41.

User Response: Correct the parameter and resubmit the job.

DLZ427I ERROR - LOG RECORD OUT OF SEQUENCE

Explanation: The log print utility detected a log record out of sequence for one of the following reasons:

- The sequence number in the record was not one larger than the previous record.
- The first record did not have a sequence number of one.
- A previous record was found to be in error for another reason.
- The record has an earlier date/time than the previous record.
- Multiple login files on the same tape are out of sequence.

User Response: None.

DLZ428I ERROR ON FILENAME - *file name*, LOG COPY HALTED

Explanation: The log print utility detected an invalid record on the file name indicated in the message text. The error condition is identified by a previous error message. The copy function that was requested by specifying COPY on the LO input control statement, is now stopped. Further processing on this file or any subsequent file is terminated.

User Response: None.

DLZ429I LOG COPY SUCCESSFUL

Explanation: The log print utility successfully copied the contents of the input log file to a new log tape file. The copy function was requested by specifying COPY on the LO input control statement. The new log tape file may be used as input to the backout utility.

User Response: None.

DLZ430I ERROR - INVALID CICS DUMP PARAMETER

Explanation: The CICS/VS dump parameter specified in the LO control statement was not CICS/D or did not begin in column 46.

User Response: Correct the parameter and resubmit the job.

DLZ432I ERROR - INVALID DBDNAME

Explanation: The DBD name specified in the LS control statement is invalid for one of the following reasons:

- The name contained other than alphanumeric characters.
- The name was longer than 7 characters.
- The name did not begin in column 4.
- The columns immediately following the name were not blank.

User Response: Correct the DBD name and resubmit the job.

DLZ433I ERROR - INVALID TASK ID

Explanation: The CICS/VS task id specified in the LS control statement is invalid for one of the following reasons:

- The id contained other than numeric characters.
- The id was longer than 5 characters.
- The id did not begin in column 2.
- The columns immediately following the ids were not blank.

User Response: Correct the error and resubmit the job.

DLZ434I INVALID RELATIVE BLOCK NUMBER IN LS CONTROL STATEMENT

Explanation: The relative block number (RBN) specified in the LS control statement is invalid for one of the following reasons:

- The RBN contained other than hexadecimal characters (0 to 9, A to F).
- The RBN was longer than 8 characters.
- The RBN did not begin in column 18.
- The columns immediately following the RBN were not blank.

User Response: Correct the RBN and resubmit the job.

DLZ435I INVALID REWIND OPTION SPECIFIED. DEFAULTED TO "NO REWIND"

Explanation: Only N (no rewind), U (rewind and unload), or R (rewind) may be specified as rewind options. The file will not be rewound at the end of processing.

User Response: Correct the parameter in error before running the job again.

DLZ436I REWIND OPTION SPECIFIED FOR DISK FILE IGNORED

Explanation: A rewind option was specified for a file along with a type parameter of V or S, identifying the file as a disk file.

User Response: Either remove the rewind specification or correct the file type specification before running the job again.

DLZ437I ONLY ONE {LO|DI|AI|AO} STATEMENT IS ALLOWED, STATEMENT IGNORED

Explanation: More than one LO, DI, AI, or AO statement was supplied. Only the first one will be used.

User Response: Remove the extra statements before running the job again.

DLZ438I BLKSIZE ON FILE *file name* IS *blksize* BYTES (*fact*)

Explanation: The Image Copy Utility shows the *blksize* selected for OUTFILE1/OUTFILE2 *file name* as specified in the 'D' control statement. The number of CIs that may fit into one output block is *fact*. This should be specified later in the control statement of the data base data set recovery utility if you want to recover the appropriate data base.

The *blksize* was calculated from the value specified (or the default) and the CI size (control interval size) specified at DBD generation for the data base to be dumped. If nothing was specified and the CI size of the data base does not exceed 4096 bytes, the default output block size of 4112 bytes was used. If the CI size is greater than 4096 bytes, the block size selected will contain one CI.

A VSE return code of 0 or 4 will be returned at job termination:

- retcode 0 will be returned if the specification leads to a valid calculated block size (or the default was used).
- retcode 4 will be returned if nothing was specified but the CI size exceeds the default; or if, in the specification, a valid keyword (nnCI) was found but the nn factor was invalid (not numeric or out of range); or the calculated block size was out of range and therefore should be corrected.

User Response: None. The Image Copy Utility will run successfully. If the selected block size is not what you expected, correct the parameter on

the control statement and rerun the job. Refer to the *DL/I DOS/VS Resource Definition and Utilities* for more information about control card specifications.

DLZ439I BLKSIZE OF IMAGE-COPY-INPUTFILE IS CALCULATED TO *blksize* BYTES (*fact* CI)

Explanation: The Data Base Data Set Recovery Utility shows the block size selected for the image copy input file specified in the 'S' control statement. The number of CIs that may fit into one input block is *fact*, which should be the same value used for creation of the image copy file. The *blksize* was calculated from the value specified (or the default) and the CI size (control interval size) specified at DBD generation for the data base to be recovered. If nothing was specified and the CI size of the data base does not exceed 4096 bytes, the default input block size of 4112 bytes was used. If the CI size is greater than 4096 bytes the block size selected will contain one CI.

For the values that can be specified in the control card refer to the *DL/I DOS/VS Resource Definition and Utilities*.

A VSE return code of 0 or 4 will be returned at job termination:

- retcode 0 will be returned if the specification leads to a valid calculated block size (or the default was used).
- retcode 4 will be returned if nothing was specified but the CI size exceeds the default; or if, in the specification, a valid keyword (nnCI) was found but the nn-factor was invalid (not numeric or out of range); or the calculated block size was out of range and therefore needs to be corrected.

User Response: None. The data base data set recovery utility will run successfully if the selected block size is not less than the block size of the image copy input file. Refer to the *DL/I DOS/VS Resource Definition and Utilities* for more information about control card specifications.

DLZ440I ERROR - INVALID BUFFER SIZE SPECIFICATION

Explanation: The buffer size specification on the TI statement was invalid for one of the following reasons:

- It contains non-numeric characters.
- It is greater than 32767.

User Response: Correct error and resubmit job.

DLZ441I BLKSIZE OF IMAGE-COPY-INPUTFILE (*blksize* BYTES, *fact* CI) IS GREATER THAN CALCULATED BLKSIZE

Explanation: The value specified in the 'S' control statement of the Data Base Data Set Recovery Utility leads to a calculated block size for the image copy input file which is less than the actual size of the file. (The previously issued message DLZ439I shows the values specified).

blksize is the real size of the image copy input file as it was calculated from the Image copy utility. The number of CIs that may fit into one input block is *fact*, which should be specified in the 'S' control statement. For the values that can be specified in the control statement (which applies only for image copy input files), refer to the *DL/I DOS/VS Resource Definition and Utilities*.

User Response: Correct the 'S' control statement to the value of *fact* and rerun the job.

DLZ442I ABNORMAL ENDING

Explanation: A serious error was detected by the trace print utility. The job is terminated with a core storage dump.

User Response: If possible, correct the error and rerun job. If cause of error cannot be determined, save dump for further problem determination.

DLZ443I I/O ERROR ON TRACE INPUT

Explanation: An I/O error occurred trying to read the trace input file.

User Response: Correct problem with device or assign a different device and rerun job.

DLZ445I END OF FILE REACHED ON TRACE INPUT

Explanation: An end-of-file condition was returned while reading the trace input.

User Response: None.

DLZ446I ONLY ONE TI CONTROL STATEMENT IS ALLOWED, JOB TERMINATED

Explanation: Only one TI statement is allowed per execution of DLZTPRT0. Additional statements are flagged with this message and the statement is not syntax checked. The job is terminated.

User Response: Ensure that the correct TI statement is in the job stream and remove all other TI statements. Rerun the job after correcting any error conditions.

DLZ447I {TYPECALL|CALLNUM|MODULE|TASKID} WAS PREVIOUSLY DEFINED

Explanation: Each of the parameters on the TO statement can be defined only once. If multiple entries of any parameter are found, all entries of the parameter after the first are flagged. The job is terminated.

User Response: Ensure that the correct parameter specification is in the job stream and remove all others. Rerun the job after correcting any error conditions.

DLZ448I {TYPECALL|CALLNUM|MODULE|TASKID} SYNTAX IS INVALID

Explanation: A syntax error was detected in the named TO statement parameter. An error flag is set and the job is terminated.

User Response: Correct the parameter syntax error and rerun the job. The correct syntax for the TO statement is given in the *DL/I DOS/VS Diagnostic Guide*.

DLZ449I {TYPECALL|CALLNUM|MODULE|TASKID} CONTAINS AN INVALID VALUE

Explanation: The named option contains an invalid value. Only the documented values are allowed for the option indicated. An error flag is set and the job is terminated.

User Response: Check the control statement for the proper specifications and spelling and rerun the job. The correct value specifications are given in the *DL/I DOS/VS Diagnostic Guide*.

DLZ450I AN INVALID OPTION WAS SPECIFIED

Explanation: An option not specified by the Trace Print Utility was found in the TO statement. The only supported options are TASKID=, MODULE=, TYPECALL=, and CALLNUM=.

User Response: Check the TO statement options for correct spelling and values. Rerun the job after correcting any error conditions. The correct TO statement options are given in the *DL/I DOS/VS Diagnostic Guide*.

DLZ451I {TYPECALL|CALLNUM|MODULE|TASKID} CONTAINS AN INVALID LIST

Explanation: The TYPECALL option contains an invalid list. A list must begin and end with parentheses. The TYPECALL list can contain a maximum of five call types. An error flag is set and the job is terminated.

User Response: Check the TYPECALL statement for proper specifications and number of calls. Rerun the job after correcting any error conditions. The correct TYPECALL parameters are given in the *DL/I DOS/VS Diagnostic Guide*.

DLZ452I A COMMA BLANK IS INVALID

Explanation: A comma followed by a blank was encountered. This is an invalid condition because a new parameter is expected after a comma. An error flag is set and the job is terminated.

User Response: Rerun the job after correcting any error conditions.

DLZ453I AN ERROR WAS DETECTED NEAR COLUMN *nn*. SEE FOLLOWING MESSAGE

Explanation: An error was detected at or near the specified column. This message precedes another message which explains the detected error.

User Response: Rerun the job after correcting any error conditions.

DLZ454I AN INVALID PARENTHESIS WAS DETECTED

Explanation: A parenthesis was detected at an invalid place. Not all options support the list format which is indicated by the parenthesis. An error flag is set and the job is terminated.

User Response: Rerun the job after correcting any error conditions. The correct syntax for commands are given in the *DL/I DOS/VS Diagnostic Guide*.

DLZ475I CALL PARAMETER LIST RETURN CODE FIELD MISSING, DLI: *task* TERMINATED

Explanation: A DL/I GSTA call was issued and the number of parameters was too small (PARMCNT less than 5) or the return code parameter entry was invalid.

User Response: Fix the bad call and resubmit the application.

DLZ476I INVALID PCB ADDRESS PASSED IN DL/I CALL, DL/I [TASK] TERMINATED

Explanation: An invalid or no PCB address in a DL/I call. Possible cause:

- The PSB language specified was not the same as the program language.
- For a remote PSB, the PCB address is not in the PCB address list.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ477I START OF JCB OVERLAID WITH KEYFEEDBACK DATA, DL/I [TASK] TERMINATED

Explanation: The beginning of the DL/I internal Job Control Block (JCB) is overlaid, because the keylength values of a DBD conflicted with the corresponding PSB coding.

Possible reasons for this are that, after the last change of the DBD:

- No ACBGEN was done.
- ACBGEN was done, but the PSB was not changed.
- ACBGEN failed and no action was taken on the corresponding messages.
- One of the libraries contains an old copy of the PSB.

User Response: Ensure that:

- The keylength values of the DBD match the corresponding PSB coding.
- The libraries are properly structured.

Redo ACBGEN and restart the job.

DLZ500I ROUTINE CREATED FOR PCB *pcbname*

Explanation: The ISQL routine was successfully created for the indicated PCB containing the DEFINE PCB, SEGMENT, and FIELD commands.

User Response: None.

DLZ501I ISQL EXTRACT DEFINES ROUTINE ALREADY EXISTS FOR PCB *pcbname*

Explanation: REPLACE has not been indicated on the control card and a routine already exists for PCB *pcbname* indicated. DEFINE commands were not put in the routine table for this control card.

User Response: If you desire this information, indicate REPLACE on the control card and rerun the job for this control card. Old information will be deleted.

DLZ502I REQUIRED PARAMETER {PSBNAME|PCBNAME|DLIPROC|USERID} MISSING ON CONTROL CARD

Explanation: A required parameter is missing from the EXTRACT DEFINES control card and the DEFINES were not created for that PCB.

User Response: Correct the control card and run the utility again for that PCB.

DLZ503I LOAD PROCESSING OPTION INVALID FOR EXTRACT

Explanation: The PCB specified on the EXTRACT DEFINES control card has a processing option of L or LS. The EXTRACT facility cannot EXTRACT PCBs with this processing option. DEFINES were not created for this PCB.

User Response: You should check to see if you specified the correct PCB number for the PSB and rerun the EXTRACT DEFINES Utility for that PCB.

DLZ504I DUPLICATE FIELD-NAME *fieldname* ENCOUNTERED PROCESSING SEGMENT *segmentname*.

Explanation: A duplicate field-name was encountered while processing the segment indicated. Probable cause may be that the segment being processed is a concatenated segment for a logical-child (LC) or a virtual logical-child (VLC), where the concatenated key, intersection data, logical parent or destination parent may have had fields defined with the same name.

User Response: One of the duplicate field-names must be changed in the ISQL routine generated, prior to execution of the routine.

DLZ505I CONTROL CARD SYNTAX ERROR - INVALID {OP CODE|PARAMETER|VALUE|DELIMITER|CONTINUATION}{word}

Explanation: A control card syntax error was detected. No processing will be attempted for the PCB with the error. The remaining cards will be scanned and processed if no errors are encountered. *word* is the word or delimiter which caused the error. OP CODE must be EXTRACT. PARAMETER is the keyword on the left of the equal sign. VALUE is the value on the right of the equal sign. If CONTINUATION is indicated, an invalid continuation was detected and no word is printed out in message.

User Response: Correct the control card and rerun the utility for that control card.

DLZ506I FIELD-NAME *fieldname* IN SEGMENT *segmentname* DOES NOT BEGIN WITH ALPHABETIC CHARACTER

Explanation: A field-name that starts with a non-alphabetic character was encountered while processing the segment specified. The DEFINE FIELD command was generated for this field which is defined in the DBD. The EXTRACT facility for SQL/DS does not allow field-names to start with a non-alphabetic character.

User Response: The field-name must be changed in the ISQL routine generated, prior to execution of the routine.

DLZ507I WARNING - NO FIELDS DEFINED FOR SEGMENT *segmentname*

Explanation: No fields were defined in the DBD for the segment specified. No data from this segment will be extracted to SQL/DS.

User Response: If data from the specified segment is desired to be extracted to SQL/DS, DEFINE FIELD commands which define the data in the segment should be inserted into the ISQL routine for the PCB.

DLZ570I INVALID SECONDARY LIST TYPE FOUND IN DBD *dbdname*

Explanation: This is an internal system error.

User Response: Remove all PSBs which reference the named DBD from the control card list, and reexecute the program. This is a DL/I logic error. Call the IBM Support Center.

DLZ571I *WARNING* DUPLICATE INPUT PSBNAME --*psbname*-- IGNORED

Explanation: A duplicate PSB name was specified in the PSB operand of the BUILD card.

User Response: The second occurrence of the specified PSB name is ignored.

DLZ572I *WARNING* DMB *dmbname* ALREADY EXISTS

Explanation: DMB=YES was coded on the BUILD statement indicating that DMBs were to be unconditionally generated. The DMB is already present in at least one of the libraries defined in the search chain. It will be overlaid if the output is linkage edited and cataloged and if the target library (where the linkage editor writes to) and the source library (from which the utility is reading) are the same.

In case of multiple libraries assigned via a LIBDEF JCL statement, the old DMB may still exist after linkage editing and cataloging.

DLZ573I *WARNING* DMB *dmbname* DOES NOT EXIST

Explanation: DMB=NO was coded on the BUILD statement, inhibiting generation of DMBs. The named DMB is not present in any of the libraries defined in the search chain and is required to successfully use at least one of the generated PSBs.

DLZ574I CDLOAD FAILED MODULE = {DLZDLBPP|DLZDLBDP} RETURN CODE=*rc*

Explanation: A CDLOAD request failed and was unable to load a module from the library. The CDLOAD return code *rc* is given in decimal format.

For an explanation of the CDLOAD return code *rc* refer to *VSE/SP Messages and Codes*, Section "VSE/Advanced Functions Return Codes."

User Response: Correct the error and rerun the ACBGEN.

DLZ575I ERROR PROCESSING TABLE *sql-table-name* SQLCODE=*nnnn*

Explanation: SQL/DS has returned an error in processing data in an SQL/DS table. Utility is terminated.

User Response: By using the SQLCODE value, refer to the *SQL/Data System Messages and Codes for VSE* manual to determine error. The user should also refer to message DLZ576I which is issued to SYSLST for further SQL/DS error information.

DLZ576I SQLCODE=*sqlcode* SQLERRP=*sqlerrp* SQLERRD=*sqlerrd* SQLERRM=*sqlerrm*

Explanation: SQL/DS has returned an error in its processing. On receiving a severe *sqlcode*, the utility is terminated at that point. For a non-severe *sqlcode*, a ROLLBACK WORK is performed and processing is continued for the next PSB or PCB. The SQL/DS variables have the following meanings:

Variable Explanation

sqlcode Summarizes the result of executing the statement. These return codes are documented in the *SQL/Data System Messages and Codes for VSE* manual. Severe *sqlcodes* are -805, -806, -807, -902, -931, -932, -933, -935, -938, -940, and -941. For EXTRACT DEFINES: If *sqlcode* is 100, probable cause of error is that an incorrect *psbname* or *pcbnumber* was specified on the control card.

sqlerrp Contains the name of the SQL/DS routine that discovered the error.

sqlerrd The message returns two numbers for this variable. The first is the Relational Data System (RDS) or Resource Manager return code. The second is the Data Base Storage System (DBSS) or Resource Manager return code.

sqlerrm SQL/DS message test variables used to generate the SQL/DS message for the *sqlcode* returned.

User Response: Use the information returned to determine the error. Correct the error and rerun the utility for those PSBs or PCBs for which processing was not successfully completed.

DLZ577I SQL/DS CONNECT FAILED, SQLCODE=*nn*

Explanation: An SQL/DS CONNECT failed for the *userid* and *password* specified on the control card. Utility is terminated.

User Response: By using the SQLCODE value, refer to the *SQL/Data System Messages and Codes for VSE* manual to determine error. Also refer to message DLZ576I which is issued to SYSLST for further SQL/DS error information.

DLZ578I ERROR OCCURRED EXECUTING {GETVIS|FREEVIS} MACRO, RETURN CODE=*rc*

Explanation: A DL/I module encountered an error executing a GETVIS or FREEVIS request. The utility terminated. The GETVIS or FREEVIS return code *rc* is given in decimal format.

For an explanation of the GETVIS or FREEVIS return code *rc* refer to *VSE/SP Messages and Codes*, Section "VSE/Advanced Functions Return Codes."

User Response: Correct the error and rerun the utility.

DLZ583I SYSPCH NOT ASSIGNED

Explanation: Output was designated to the SYSPCH device and SYSPCH was assigned IGN or UA.

User Response: Assign SYSPCH and reexecute the program.

DLZ584I SYSLNK NOT ASSIGNED

Explanation: Output was designated to the SYSLNK device and SYSLNK was assigned IGN or UA.

User Response: Assign SYSLNK and reexecute the program.

DLZ585I UTILITY TERMINATED DUE TO ERROR

Explanation: A control card error indicated by a previously printed message caused the program to terminate before all functions were complete. No control blocks were created.

User Response: Correct the errors and reexecute the program.

DLZ587I ERROR BUILDING PSB=*psbname* - IT WILL NOT BE BUILT

Explanation: The block builder returned a non-zero completion code after processing the indicated PSB. Processing continues for any other named PSBs.

User Response: Correct any errors indicated by the error messages, and reexecute the program with a BUILD card for this PSB.

DLZ588I CONTROL CARD SYNTAX ERROR - INVALID {CONTINUATION|BLK TYPE|FORMAT|LABEL|OP CODE|OPERAND}[*symbol*]

Explanation: A control card syntax error was detected. The remaining cards will be scanned, but no processing will be attempted. The *symbol* printed is the symbol or delimiter which caused the error. If FORMAT is indicated, the logical end of statement was reached before the statement was complete. If CONTINUATION is indicated, an invalid continuation was detected.

User Response: Correct the control card and reexecute the program.

DLZ589I PROCESSING COMPLETED FOR PSB----*psbname*

Explanation: The blocks for the indicated PSB have been successfully built and written to SYSLNK or SYSPCH.

User Response: None required.

DLZ600I SUCCESSFUL COMPLETION OF PARTIAL DATA BASE REORGANIZATION

Explanation: Partial data base reorganization has completed.

User Response: None required.

DLZ602I **INVALID KEYWORD IN PRECEDING STATEMENT COLUMN = *nn***

Explanation: The keyword beginning in column *nn* is invalid.

User Response: Correct the error in the control statement and rerun the job.

DLZ603I **ILLEGAL CONTINUATION**

Explanation: One of the following was detected:

- The ending character in the input string was a comma, and a blank was found in column 72.
- The ending character was not a comma, and a non-blank character was found in column 72.

User Response: Correct the error in the control statement and rerun the job.

DLZ604I **FROMAREA INVALID FOR HIDAM OR KEYRANGE INVALID FOR HDAM**

Explanation: FROMAREA can only be specified for HDAM. KEYRANGE can only be specified for HIDAM.

User Response: Correct the error in the control statement and rerun the job.

DLZ605I **REQUIRED KEYWORD MISSING, KEYWORD = *keyword***

Explanation: The indicated required keyword is missing.

User Response: Correct the error in the control statement and rerun the job.

DLZ606I **INVALID KEYWORD IN PRECEDING STATEMENT, KEYWORD = *keyword***

Explanation: The indicated keyword is invalid for the previous statement.

User Response: Correct the error in the control statement and rerun the job.

DLZ608I **KEYWORD PREVIOUSLY SPECIFIED, KEYWORD = *keyword***

Explanation: The indicated keyword was specified previously in another input statement. Only one keyword of this type is allowed.

User Response: Correct the error in the control statement and rerun the job.

DLZ609I **FROMAREA AND KEYRANGE ARE MUTUALLY EXCLUSIVE KEYWORDS**

Explanation: FROMAREA and KEYRANGE were both specified. FROMAREA may only be used with HDAM data bases. KEYRANGE may only be used with HIDAM data bases.

User Response: Correct the error in the control statement and rerun the job.

DLZ610I **INVALID OPERAND IN PRECEDING STATEMENT FOR KEYWORD = *keyword***

Explanation: An invalid operand has been found for the keyword indicated in the previous statement. Possible errors that might cause this message are:

- A missing quote on sort options.
- Required operand not present.
- Present operand not required/valid for PART2.
- Invalid operand length.
- Invalid key length.
- PSBname length greater than 8 or equal to zero.
- Invalid SCAN segment name specified.
- Non-numeric data specified where numeric data must be specified.
- Invalid range specified.
- Invalid sort operand.

User Response: Correct the error in the control statement and rerun the job.

DLZ611I **MORE THAN 10 RANGES SPECIFIED**

Explanation: No more than 10 KEYRANGES, FROMAREAs, or TOAREAs are allowed.

User Response: Correct the error in the control statement and rerun the job.

DLZ612I **DBD *dbdname* NOT FOUND**

Explanation: The DBD named was not found in any of the libraries defined in the search chain.

User Response: Correct the error in the control statement and rerun the job.

DLZ613I **DBD *dbdname* DOES NOT HAVE HD ACCESS METHOD**

Explanation: The named DBD is not an HDAM or HIDAM data base.

User Response: Correct the error in the control statement and rerun the job.

DLZ614I **DATASET SPECIFIED ON INPUT STATEMENT DOES NOT MATCH DBD *dbdname***

Explanation: The data set used in the TOAREA input statement is not the same as found in the DBD.

User Response: Correct the error in the control statement and rerun the job.

DLZ615I **INTERNAL ERROR IN *modname***

Explanation: Module *modname* has encountered an error that prevents further processing. Preceding messages further identify the error(s).

User Response: Save the output for problem determination.

DLZ616I **LCHILD ENTRY DOES NOT EXIST IN DBD *dbdname* FOR *segmname***

Explanation: No LCHILD entry was found in the related data base for the logical child in the primary data base, or an LCHILD entry referred to a related data base for which the identified segment does not exist.

User Response: Correct the error in the control statement and rerun the job.

DLZ617I **MORE THAN 49 RELATED DATA BASES**

Explanation: The limit of 49 related data bases has been exceeded.

User Response: Partial data base reorganization cannot be used for this reorganization: Full reorganization is required.

DLZ618I **MORE THAN 500 SEGMENTS PARTICIPATING IN REORGANIZATION**

Explanation: The limit of 500 segments participating in reorganization has been exceeded.

User Response: Partial data base reorganization cannot be used for this reorganization: Full reorganization is required.

DLZ623I **SCAN AND RELOAD ACTIONS REQUIRED FOR POINTER RESOLUTION EXCEEDED**

Explanation: The limit of 500 scan and related actions for logically related segment pointer resolution was exceeded.

User Response: Partial data base reorganization cannot be used for this reorganization: Full reorganization is required.

DLZ627I **NO PSB GENERATED FOR THIS EXECUTION OF PARTIAL REORGANIZATION**

Explanation: The PSB parameter was not included in the input. The default for this parameter is no PSB generation.

User Response: If a PSB is required, rerun PART1 with PSB=*psbname* specified.

DLZ633I **DBDS DO NOT MATCH, PART1 DBD = *dbdname*, PART2 DBD = *dbdname***

Explanation: The DBD specified in the PART1 input is not the same as the DBD specified in PART2.

User Response: Correct the error in the control statement and rerun the job.

DLZ634I **PREMATURE END OF FILE ON CONTROL TABLE DATASET**

Explanation: End of file was reached before all the records passed from PART1 to PART2 on the control table data set. Preceding messages further identify the error(s).

User Response: Rerun PART1 to recreate the control table data set.

DLZ635I **PART1 HAS TERMINATED AFTER PHASE *phasename***

Explanation: The phase shown in the previous message text encountered an error that prevents further processing. Another error message from the failing module should always precede this message.

User Response: Correct the error and rerun the job.

DLZ636I **UNSUCCESSFUL COMPLETION OF PARTIAL DATA BASE REORGANIZATION**

Explanation: An error occurred that prevents successful completion of Partial Data Base Reorganization. Preceding messages further identify the error(s).

User Response: Save output for problem determination.

DLZ639I **INSUFFICIENT PARTITION SIZE**

Explanation: There is not enough storage space available for PART1 or PART2 to build control blocks.

User Response: Increase the partition size or specify a SIZE parameter on the EXEC statement. Then rerun the affected part of the process.

DLZ641I **SEGMENT *segmname* IS NOT TARGET OF INDEX *indexname***

Explanation: The LCHILD statement in the index DBD definition identified *segmname* as the target segment for the index; however, there is no LCHILD for *segmname* that identifies the index.

User Response: Change index LCHILD statement if it identifies the wrong prime data base segment, or correct or add an LCHILD statement for the segment that identifies the index. Rerun Part 1 of the partial reorganization utility.

DLZ642I **RELOAD COUNT NOT EQUAL TO UNLOAD COUNT FOR ALL SEGMENTS**

Explanation: The number of segments loaded does not equal the number of segments unloaded. See range reload statistics for further information.

User Response: Execute the DL/I backout utility to undo changes made during partial data base reorganization.

DLZ643I **NO PRIME INDEX ENTRY IN DATA BASE TABLE FOR *file name***

Explanation: There is no primary index data set for the HIDAM data base.

User Response: Execute the DL/I backout utility to undo changes made during partial data base reorganization.

DLZ644I RBA IN PRIME INDEX NOT EQUAL TO RBA OF UNLOADED SEGMENT

Explanation: The RBA in the primary index is not equal to the RBA of the unloaded segment.

User Response: Execute the DL/I backout utility to undo changes made during partial data base reorganization.

DLZ645I FROMAREA *low-block-number* TO *high-block-number* NOT IN ROOT ADDRESSABLE AREA

Explanation: The specified FROMAREA is not wholly contained in the root addressable area.

User Response: Correct the error in the control statement and rerun the job.

DLZ646I NO RECORDS IN SPECIFIED RANGE

Explanation: No records are found in the indicated range.

User Response: Correct the error in the control statement and rerun the job.

DLZ647I SORT *xx* FAILED

Explanation: A bad return code was returned from the partial data base reorganization sort. *xx* refers to the sort step.

User Response: Execute the DL/I backout utility to undo changes made during partial data base reorganization.

DLZ648I END OF DATA BASE TABLE WITHOUT INDEX MAINTENANCE RECORD MATCH

Explanation: A specific data base was not found in the data base table as determined by the index maintenance record.

User Response: Execute the DL/I backout utility to undo changes made during partial data base reorganization.

DLZ649I UNEXPECTED EOF ON PRWRKF4

Explanation: An end of file condition occurred for PRWRKF4 while trying to locate a 'B' index work record that matches the DL/I index maintenance record currently being processed.

User Response: Execute the DL/I backout utility to undo changes made during partial data base reorganization.

DLZ650I INVALID RECORD TYPE *xx* IN DATA WORK RECORD

Explanation: A data work record was encountered that contained a record type not defined to the system.

User Response: Execute the DL/I backout utility to undo changes made during partial data base reorganization. See Chapter 6, "DL/I Status Codes" on page 111 for information on status codes.

DLZ651I NO DMB FOR DATA BASE *dbdname*

Explanation: The data management block (DMB) was not found in the directory (DDIR).

User Response: Save the output and contact the IBM Support Center.

DLZ652I PERMANENT READ ERROR FOR SEGMENT *segmname* {RBA|RBN}

Explanation: A permanent read error occurred for the segment identified. The RBA or RBN is identified if it is known by the module detecting the error.

User Response: Execute the DL/I backout utility to undo changes made during partial data base reorganization.

DLZ653I INVALID RETURN FROM DL/I CALL

{function, dbdname, procopt, segmname, level, status-code}
{DLZDBH00|DLZDBDS0} RETURN CODE =
xxxxxxx}

Explanation: One of the following has occurred:

- A status code other than bb, GA, or GK was returned from an external DL/I call. Refer to Chapter 6, "DL/I Status Codes" on page 111 for information on status code *status-code*.
- A non-zero eight-digit return code *xxxxxxx* was returned from an internal DL/I call. *xxxxxxx* is the value returned in field PSTRTCODE of the PST DSECT. If the return code is from:
 - DLZDBH00 (DL/I buffer handler), see message DLZ958I for an explanation of *xxxxxxx*. (Only the last two digits of *xxxxxxx* are significant.)
 - DLZDBDS0 (DL/I space management), return code *xxxxxxx* indicates the following:

xxxxxxx	Explanation
00000003	Bit map update required.
00000030	Block not used due to distributed free space parameter.

Note: DL/I makes internal calls while performing index maintenance.

User Response: Execute the DL/I backout utility to undo changes made during partial data base reorganization.

DLZ655I INVALID PHYSICAL TWIN CHAIN

Explanation: An invalid physical twin chain was encountered while unloading a HIDAM root. This is an indication of a bad data base.

User Response: Execute the DL/I backout utility to undo changes made during partial data base reorganization.

DLZ659I INVALID RECORD TYPE xx IN INDEX WORK RECORD

Explanation: An index work record was encountered that contained a record type code that is not defined to the system.

User Response: Execute the DL/I backout utility to undo changes made during partial data base reorganization.

DLZ772I ERROR IN DMB OF INDEX RELATIONSHIP, DL/I [TASK] TERMINATED

Explanation: There should be at least two secondary list entries in the secondary list of the SCD for the root segment of a data base being accessed via alternate processing sequence (secondary index). Only one was found.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ796I INVALID POINTER ON DELETE, DL/I [TASK] TERMINATED

Explanation: While deleting a segment, an invalid condition was detected which involved RBA pointers.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ797I INVALID POINTER ON INSERT, DL/I [TASK] TERMINATED

Explanation: While inserting a segment, an invalid condition was detected which involved RBA pointers.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ798I UNEXPECTED SEGMENT CODE ON RETRIEVE, DL/I [TASK] TERMINATED

Explanation: While retrieving a segment, the segment code from the segment just accessed does not match the expected segment code in the Segment Descriptor Block (SDB). There is either an invalid RBA pointer, or it is pointing to invalid data.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are

found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ799I SEGMENT LENGTH OVER MAXIMUM ON REPLACE, DL/I [TASK] TERMINATED

Explanation: During segment replacement the user compression routine returned a compressed segment whose length is greater than the maximum specified during the DBD generation for the segment.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ800I SEGMENT LENGTH OVER MAXIMUM ON RETRIEVE, DL/I [TASK] TERMINATED

Explanation: During segment retrieval the user compression routine returned a segment where length is greater than the maximum specified during the DBD generation for the segment.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ801I BAD RETURN CODE FROM DL/I BUFFER HANDLER, DL/I [TASK] TERMINATED

Explanation: DLZDLR00 received a bad return code from a call to the buffer handler. The following pertinent PST fields and contents may be found in the pseudoabend save area.

Name	Description
PSTFNCTN	Requested function
PSTRTCDE	Return code (see message DLZ958I for meaning of non-zero codes)

A possible cause for this message is the use of the processing option PROCOPT=GO(P) with limited data base sharing.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ802I NO MORE SPACE IN DATA BASE, DL/I [TASK] TERMINATED

Explanation: Space management (DLZDHDS0) indicated that no space is available on the data base.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and

debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ803I INVALID POINTER, DL/I [TASK] TERMINATED

Explanation: Invalid pointer conditions were found while deleting a logically related segment. The logical parent or child could not be found.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ804I INSUFFICIENT BUFFER SPACE, DL/I [TASK] TERMINATED

Explanation: There is no space for work area in the buffer pool. The number of the data base buffer subpools should be increased and the job rerun.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ805I CANNOT FREE WORK AREA, DL/I [TASK] TERMINATED

Explanation: Attempt to free buffer for work area unsuccessful.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ806I ERROR RETURN CODE FROM BUFFER HANDLER, DL/I [TASK] TERMINATED

Explanation: A non-zero return code from buffer handler was received for one of the following reasons:

- An error occurred when trying to mark a buffer altered.
- An error occurred on a read request.
- An error occurred on an erase request.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ807I PHYSICAL PATH TO ROOT SEGMENT UNAVAILABLE, DL/I [TASK] TERMINATED

Explanation: Delete/replace cannot obtain position for work area (physical path to root required).

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ808I INVALID LEVEL TABLE ENTRY, DL/I [TASK] TERMINATED

Explanation: The LEVTTR field does not contain a valid value for DLZDLD00.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ809I INVALID LOGICAL CHILD COUNTER, DL/I [TASK] TERMINATED

Explanation: Invalid pointer conditions were found while deleting a logically related segment. The counter in the prefix of the logical parent was negative.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ830I REQUESTED SPACE INCOMPATIBLE WITH CI SIZE, DL/I [TASK] TERMINATED

Explanation: Requested space larger than the maximum space in a VSAM control interval. Check DBD generation listing for required VSAM control interval size.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ831I DL/I CANNOT GET FBA DEVICE CHARACTERISTICS, DL/I TERMINATED

Explanation: Module DLZDCI00 or DLZDHDS0 has encountered one of the following:

- Cannot get space for SHOWCAT or GETVCE
- Error from SHOWCAT
- Error from GETVCE
- Cannot find data object returned from GETVCE
- Not an FBA device

User Response: Either:

- Ensure that the data base specified is mounted on a FBA device and rerun the job, or
- Regenerate and recatalog the DBDGEN specifying the proper device and then rerun the job.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ832I INVALID FREE SPACE ELEMENT FOUND, DL/I TERMINATED

Explanation: Space Management detected a free space element that pointed beyond the end of the control interval. This can be caused by a free space element pointer or length.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ841I BUFFER SPACE UNAVAILABLE, DL/I [TASK] TERMINATED

Explanation: No available space in data base buffer pool. The data base buffer pool size can be controlled through the parameter card for DL/I.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ844I FILE OUT OF SPACE, DL/I [TASK] TERMINATED

Explanation: No available space in file. Increase file size allocation.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ845I UNEXPECTED CONDITION IN DL/I BUFFER HANDLER, DL/I TERMINATED

Explanation: An internal error occurred in the DL/I buffer handler regarding either invalid data in a buffer or not being able to find an expected buffer. This message is issued by DL/I module DLZDBH00.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are

found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ847I UNEXPECTED RETURN CODE IN DL/I BUFFER HANDLER, DL/I TERMINATED

Explanation: Unexpected return from the enqueue/dequeue routine.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ848I BUFFER SPACE REQUEST TOO LARGE, DL/I [TASK] TERMINATED

Explanation: The request for buffer space specified a length that exceeded the size of any buffer in any subpool.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ850I INVALID FUNCTION CODE ON INSERT, DL/I [TASK] TERMINATED

Explanation: Unrecognized function in JCBPRESF.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ855I VSAM RBA NOT EQUAL TO RBA DL/I CALCULATED, DL/I [TASK] TERMINATED

Explanation: RBA returned by VSAM not the same RBA value calculated by DLZDDLE0.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ860I ERROR DETECTED BY DL/I BUFFER HANDLER, DL/I [TASK] TERMINATED

Explanation: Bad return from a call to the data base buffer handler. The following pertinent PST fields and contents may be found in the pseudoabend save area:

Name	Description
PSTFNCTN	Requested function
PSTRTCDE	Return code (see message DLZ958I for the meaning of the non-zero return code)

Within the save area trace, find the save area where:

Register 1: PST address.
 Register 4: SDB address.
 Register 6: JCB address.
 Register 8: Return within DLZDDLE0.
 Register 12: Entry point to DLZDDLE0.

Probable causes:

- Bad insert position in SDBs from retrieve (DLZDLR00).
- Data base had bad pointers.
- Error in DLZDDLE0.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ861I **INVALID SEGMENT CODE ON HISAM INSERT, DL/I [TASK] TERMINATED**

Explanation: During a dependent segment insert in HISAM, an invalid segment code was encountered by DLZDDLE0 (LOAD/INSERT) in determining length of segments to the right of the insert point.

Within save area trace, find the save area where:

Register 1: PST address.
 Register 4: SDB address.
 Register 6: JCB address.
 Register 9: Address of invalid segment code.
 Register 11: Insert point.
 Register 12: Entry point to DLZDDLE0.

Probable causes:

- DBD generation or VSAM access method services command DEFINE changed the length of segments and the data base was not reloaded.
- Bad insert position in SDBs from retrieve (DLZDLR00).

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ862I **LOGICAL PARENT KEY MISMATCH ON INSERT, DL/I [TASK] TERMINATED**

Explanation: When inserting a concatenated segment with a logical parent, using the insert rule virtual, the format of the user I/O area was correct, but, when attempting to replace the logical parent, the key in the logical parent did not match.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are

found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ863I **ERROR IN USER SEGMENT COMPRESSION ROUTINE, DL/I [TASK] TERMINATED**

Explanation: An error in your edit/compression routine resulted in one of the following:

- The segment length is greater than the defined maximum (if variable length).
- The segment length is smaller than specified in DMBDL for fixed length.
- The key was changed by this routine.

Online processing:

For online processing the registers in the DL/I abend save area are:

Register 1: PST address.
 Register 4: LSDB address.
 Register 5: PSDB address.

The formatted dump contains additional information in the registers in PSTSV3 in the PST save area:

Register 1: PST address.
 Register 2: Segment source address in the user's I/O area.
 Register 4: LSDB address.
 Register 5: PSDB address.

Batch processing:

For batch processing the registers in PSTSV3 in the DL/I abend save area are:

Register 1: PST address.
 Register 2: Segment source address in the user's I/O area.
 Register 4: PSDB address.
 Register 5: Segment compaction table address entry.
 Register 10: Segment destination address.
 Register 12: Entry point to DLZDDLE0.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ864I **WORK FILE ERROR ON LOAD, DL/I TERMINATED**

Explanation: During loading of a logical parent or logical child segment, the work file could not be opened, or an I/O error occurred on the work file. Message DLZ007I which was printed previously explains the error.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ868I BAD CONTROL BLOCKS FOR SECONDARY INDEX, DL/I [TASK] TERMINATED

Explanation: DLZDXMT0 found bad control blocks for a secondary index.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ869I OPEN ERROR ON WORK FILE, DL/I [TASK] TERMINATED

Explanation: An error occurred while DLZDXMT0 was trying to open the work file.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ870I FILE ACCESS ERROR ON WORK FILE, DL/I [TASK] TERMINATED

Explanation: An error occurred while DLZDXMT0 was trying to access the work file.

User Response: See "DL/I System/Task Abnormal Termination" on page 61.

Note: Additional aids for interpreting and debugging dumps after this message is issued are found in "Interpreting and Debugging Dumps" in the *DL/I DOS/VS Diagnostic Guide*.

DLZ888I NO DATA BASE RECORDS READ FOR PSB *psbname*

Explanation: The log tape supplied contained no data base log records for the PSB specified. This message may also occur if the task that used the named PSB terminated normally.

User Response: Information message; none required.

DLZ894I INVALID RECORD ENCOUNTERED ON INPUT LOG FILE

Explanation: A record was read from the input log file which could not be identified, or contained unexpected information.

User Response: Verify that the file read was a DL/I log file. If it was not, mount the correct log file and resubmit the job. If processing a CICS/VS journal, the buffer size in the LI statement for the utility must be at least as big as the CICS/VS journal buffer size (1100 bytes minimum).

DLZ900I SEGMENT *segmname1* IN DBD *dbdname1* INVALIDLY REFERENCES SEGMENT *segmname2* IN LOGICAL DATA BASE *dbdname2*

Explanation: *segmname2* in *dbdname2* was referenced either on an LCHILD statement or as the second (logical parent) operand for the PARENT parameter of the SEGM statement for a logical child segment. *dbdname2* is a logical data base.

User Response: Correct the DBD definition and rerun the ACB generation.

DLZ901I APPLICATION ORDER OF SUBFIELDS OF FIELD *fldname* NOT EQUAL PHYSICAL ORDER - SEGMENT *segmname*

Explanation: Field *fldname* in segment *segmname* is either the sequence field for a segment, or the key field for an alternate processing sequence, and the subfields in the application view are in a different order than in the physical definition.

User Response: Reorder the subfields to match the views. This will require either reloading the data base or rewriting the application.

DLZ902I WARNING.. POSSIBLE CONVERSION CONFLICT - SEGMENT *segmname*, FIELD *fldname*

Explanation: The identified field is indicated as having a different type in the application view than in the physical view. It has no subfields in the application view, but does have subfields in the physical definition. Conversion as a single field of a field made up of subfields will probably cause conversion errors.

User Response: Define the subfields in the application view.

DLZ903I PRIOR FIELD *fldname1* START LOCATION NOT VALID FOR ALTERNATE SEQUENCE KEY FIELD *fldname2* .. SEGMENT *segmname*

Explanation: The name of a field was supplied as the START location parameter for search field (or a subfield of a search field) for an alternate processing sequence, and the specified field was not another search field or a subfield of another search field.

User Response: Either redo the DBDGEN and define the field or correct the name of the field in the START parameter.

DLZ904I DMB *dmbname* ALREADY IN LIBRARY, DMB WILL NOT BE BUILT

Explanation: The DMB named in the above message already exists in a library. This is a warning message only.

User Response: If the DBD has changed and should be used to create a new DMB, delete the old DMB from the library and reexecute the application control blocks utility. Another

alternative is to simply reexecute the application control blocks utility with DMB=YES specified.

DLZ905I INSUFFICIENT STORAGE FOR PSB *psbname*

Explanation: A GETVIS request failed for one of the following reasons and the ACB utility was unable to build the control blocks for the named PSB:

1. The SIZE parameter is missing from the EXEC statement.
2. The utility program did not execute in a virtual partition.
3. There was not sufficient virtual storage available.

User Response: Increase the virtual partition size or specify a SIZE parameter on the EXEC statement. Reexecute the program.

DLZ906I INVALID PROCOPT FOR SENSEG *senseg-name* IN PSB *psbname*

Explanation: The named SENSEG statement in the named PSB has an invalid PROCOPT specified.

User Response: Correct the PSB and reexecute the program.

DLZ907I PSB *psbname* REFERENCES LOGICAL DBD *dbdname* FOR LOAD

Explanation: The PSB *psbname* contains a PCB which references the logical DBD *dbdname*. The PROCOPT in that PCB is LOAD. This is invalid.

User Response: Correct the PCB in the named PSB, execute PSBGEN again, then rerun this job.

DLZ908I ERROR FOUND IN DBD *dbdname* USED FOR BUILDING UTILITY PSB

Explanation: The utility attempted to build a utility PSB for the specified DBD and an error was found in the DBD or the GETVIS request failed.

User Response: Correct the DBD by running a new DBD generation and reexecute the program.

DLZ909I *WARNING* PROCOPT CHANGED FOR SENSEG *senseg-name* IN PSB *psbname*

Explanation: This is a warning message only. The blocks were built unless an error occurred. Segment *senseg-name* in PSB *psbname* is one of the following:

- An index target segment (or any of its physical parents) whose secondary index is designated as PROCSEQ in a PCB.
- An index pointer segment in a secondary index data base referenced by a PCB in the named PSB.
- A root level segment of a HIDAM data base is being accessed through an inverted path but this segment is not the root of the logical DB.

In either case, neither ISRT nor DLET sensitivity is allowed. The processing option(s) have been changed accordingly.

User Response: Check that the application program using this PSB does not intend to use ISRT or DLET calls for any of the named segments. If you wish the warning message not to appear, change the SENSEG statements accordingly.

DLZ910I DBD *dbdname* EXCEEDED MAXIMUM EXTERNAL REFERENCES FOR PSB *psbname*

Explanation: The named PSB was being processed by the block builder function. There are two possible reasons for this error:

1. The named DBD caused the number of external data base references for this PSB to exceed 50. This count is the sum of all unique data base names referenced by PCB statements in the PSB and all unique references to logically or index related data bases.
2. The sum of the number of unique DBDs referenced by successful PSB builds plus the number of DBDs referenced by each PSB that was unsuccessful exceeds 100.

User Response: Either:

1. Reduce the number of external data base references in the PSB and/or DBDs, and reexecute the program.
2. Reduce the number of PSBs being built in one execution of this program. More than one execution is required in this case.

DLZ911I ENQ LIST LENGTH ERROR FOR PSB *psbname*

Explanation: The processing option intent list length was calculated incorrectly for the named PSB.

User Response: This is an internal program logic error. Remove the offending PSB, and reexecute the program. Call the IBM Support Center.

DLZ912I PSB *psbname* REFERENCES SEGM *segname* IN DBD *dbdname*. SEGM NOT FOUND

Explanation: The named PSB referenced the named SEGM in the named DBD. The named SEGM does not exist in the named DBD.

User Response: Correct the PSB or DBD and reexecute the program.

DLZ913I DBD *dbdname* CONTAINS INVALID ACCESS METHOD

Explanation: The named DBD contains an invalid or unknown access method.

User Response: Correct the DBD and reexecute the program.

DLZ914I DSG SPACE ALLOCATION ERROR FOR PSB
psbname PCB pcbname

Explanation: The space required for the named PCB within the named PSB was calculated incorrectly.

User Response: This is an internal program logic error. Remove the offending PSB, and reexecute the program. Call the IBM Support Center.

DLZ915I INDEX DBD *dbdname* HAS INCORRECT SEQ FIELD LENGTH. LENGTH SHOULD BE -- *nnn* BYTES

Explanation: The named INDEX DBD has a sequence field length specified which is not equal to the indexed field length. The required length is specified.

User Response: Correct the INDEX DBD or the INDEXED DBD, and reexecute the program.

DLZ916I NON-UNIQUE OR NO ROOT SEGMENT SEQ FIELD SPECIFIED FOR DBD *dbdname*

Explanation: The named DBD requires that a sequence field be specified for the root segment. A sequence field was not specified, or the field was specified as non-unique.

User Response: Correct the DBD and reexecute the program.

DLZ917I DBD *dbdname* REFERENCES SEGM *segmname* IN DBD *dbdname*. SEGM DOES NOT EXIST

Explanation: The first DBD referenced the named SEGM in the second DBD. The SEGM does not exist in the second DBD.

User Response: Correct one or both DBDs and reexecute the program.

DLZ918I UNABLE TO LOCATE PHYSICAL PARENT. PSB *psbname* SENSEG *senseg-name*

Explanation: The named PSB contained a SENSEG statement for the named segment. The SENSEG did not have a valid physical parent, or the hierarchical structure of the PSB was different from that of the DBD.

User Response: Correct the PSB and/or the DBDs in error, and reexecute the program.

DLZ919I PSB *psbname* PCB *pcbname* -- KEYLEN SPECIFICATION TOO SMALL. KEYLEN SHOULD BE -- *nnn* BYTES

Explanation: The named PCB in the named PSB contains a KEYLEN parameter which is too small to hold the longest fully concatenated key. The correct key length is specified.

User Response: Correct the PSB and reexecute the program.

DLZ920I PSB *psbname* REFERENCES DBD *dbdname* WITH CONFLICTING PROCOPTS

Explanation: The named PSB contains at least one reference to the named DBD with a PROCOPT of L and at least one additional reference to the same DBD with a PROCOPT of something other than L. The reference may be direct, such as in a PCB statement, or the reference may be indirect, such as in a DBD which references another DBD via logical or index relationships.

User Response: Correct the PSB and/or one or more DBDs, and reexecute the program.

DLZ921I PSB *psbname* IS NOT A PSB

Explanation: The named PSB was loaded. Upon examination it was discovered the PSB was not a valid PSB.

User Response: Check job control and/or control cards. Correct the error and reexecute the program.

DLZ922I DBD *dbdname* IS NOT A DBD

Explanation: The named DBD was loaded. Upon examination it was discovered the DBD was not a valid DBD.

User Response: Check job control and/or control cards. Correct the error and reexecute the program.

DLZ923I LC (OR LC OF) SEGM --*segmname*-- IN DBD --*dbdname*-- DATA LENGTH TOO SMALL. LENGTH SHOULD BE --*nnnnn* BYTES

Explanation: The indicated logical child segment in the indicated DBD had a bytes specification shorter than its logical parent's fully concatenated key. The minimum acceptable length is indicated.

User Response: Correct the DBD and reexecute the program.

DLZ924I INDEX SEGMENT IN DBD *dbdname* TOO SHORT. LENGTH SHOULD BE --*nnnn* BYTES

Explanation: The root segment in the named INDEX DBD had a data length that was too small to hold the required index data. The minimum acceptable length is indicated.

User Response: Correct the DBD in error, and reexecute the program.

DLZ925I LC OF SEGM *segmname* IN DBD *dbdname* HAS SEQ FIELD *seqfld* IN VIRTUAL LPCK AREA

Explanation: A named logical child segment had a sequence field defined which fell within the logical parent's concatenated key area in the logical child and the key was specified as virtual.

User Response: Correct the DBD in error, and reexecute the program.

DLZ926I INTERNAL PROGRAM ERROR AT DISPLACEMENT *nnnnnn* IN DLZ*xxxxx*, PSB *=psbname*

Explanation: An internal program logic error has occurred in module DLZ*xxxxx* while processing the named PSB.

User Response: Remove the offending PSB from the processing list, and reexecute the program. Call the IBM Support Center.

DLZ927I INDEX DBD *dbdname* INDEXES NON-EXISTENT FIELD IN DBD *dbdname*

Explanation: The named INDEX DBD specified an indexed field in the INDEXED DBD which did not exist at the root level.

User Response: Correct the DBD in error, and reexecute the program.

DLZ928I INDEX DBD *dbdname* INDEXES A FIELD WHICH IS NOT A SEQ FIELD OR XDFLD

Explanation: The named INDEX DBD indexes a valid field, but the field is not a sequence field in the case of the primary HIDAM index or an XDFLD in the case of a secondary index.

User Response: Correct the DBD in error, and reexecute the program.

DLZ929I REQUIRED MODULE NOT FOUND
{psbname|dbdname}

Explanation: A DLZBLDL macro was issued for the named PSB or DBD which was not found in any of the libraries defined in the search chain.

User Response: Link-edit the PSB or DBD into the correct library or specify the library in which it already exists in the LIBDEF search chain. Reexecute the program.

DLZ930I NOT SENSITIVE TO ALL SEARCH FIELDS FOR ALTERNATE SEQUENCE.
SEGMENT *segmname*

Explanation: Sensitivity to the XDFLD only was specified and multiple search fields exist. SENFLD statements must be provided for the XDFLD and for each SRCH field if multiple search fields exist.

User Response: Respecify the PSBGEN, adding SENFLD statements for the additional search fields.

DLZ931I INVALID INDEX RELATIONSHIP BETWEEN INDEX DBD *dbdname* AND INDEXED DBD *dbdname* IN PSB *psbname*

Explanation: The named INDEXED DBD had an index relationship with the named INDEX DBD. Either the INDEX DBD did not have a similar relationship to the INDEXED DBD, or another DBD referenced in the named PSB also had an index relationship with the INDEX DBD.

User Response: Correct the appropriate PSB and/or DBDs, and reexecute the program.

DLZ932I NO SEQUENCE FIELD DEFINED FOR INDEX SEGMENT IN INDEX DBD *dbdname*

Explanation: The named INDEX DBD does not have a sequence field defined for the index segment.

User Response: Correct the INDEX DBD and reexecute the program.

DLZ933I LT SEQ FIELD EXTENDS PAST INTERSECTION DATA IN DBD -- *dbdname* -- OF PSB -- *psbname*

Explanation: The logical twin sequence field(s) defined in a virtual logical segment in the named DBD extended beyond the intersection data (if any) of the logical child. This is invalid.

User Response: Correct the named DBD and reexecute the program.

DLZ934I SENSITIVITY SPECIFIED FOR MORE THAN ONE SECONDARY INDEX.
SEGMENT *segmname*

Explanation: More than one SENSEG statement referencing a XDFLD field was supplied for a segment.

User Response: Respecify the PSBGEN and remove the SENFLD statement(s) for the additional XDFLD(s).

DLZ935I INVALID LOGICAL STRUCTURE FOR SEGM *segmname* IN DBD *dbdname*, PSB *psbname*

Explanation: The named PSB referenced the named DBD via a SENSEG statement. However, a logical structure or relationship within this segment definition is invalid. The structure of the PCB may also be invalid. Refer to *DL/I DOS/VS Data Base Administration* for the rules to follow in defining the blocks.

For HSAM or HISAM, the most common error is that the sensitive segments are not specified in the same order as in the physical DBD.

User Response: Correct the named PSB, DBD or the logical DBD referenced by the PSB, and reexecute the program.

DLZ936I SEGM *segmname* IN PSB *psbname* SPECIFIED LOAD FOR VIRTUAL SEGMENT

Explanation: The named SEGM was referenced in the named PSB with a PROCOPT of L or LS. The SEGM is a virtual segment and, as such, cannot be loaded.

User Response: Correct the PSB and reexecute the program.

DLZ9371 **INDEX DBD *dbdname1* NOT DEFINED IN INDEXED DBD *dbdname2***

Explanation: DBD *dbdname1* is defined as INDEX organization and identifies data base *dbdname2* as the target data base. However, it is not identified as an index data base in the definition for *dbdname2*.

User Response: Correct the db-name parameter on the LCHILD statement for the index data base or include a definition for the index data base in the DBD for *dbdname2*.

DLZ9381 **INCOMPLETE SENSITIVITY TO CONCATENATED KEY FOR SEGMENT *segmname***

Explanation: Insert sensitivity was specified for a concatenated segment with field sensitivity specified but the sensitivity to the entire concatenated key was not specified.

User Response: Do the PSBGEN over and either:

- Drop insert sensitivity, or
- Include sensitivity to the concatenated key.

DLZ9391 **INDEX DBD -- *dbdname* INDEXES A SYSTEM RELATED FIELD**

Explanation: The indicated INDEX DBD contained an INDEX operand which specified a field name with /SX or /CK as the first three characters.

User Response: Correct the DBD and reexecute the job.

DLZ9401 **INVALID ALTERNATE PROCSEQ FOR SENSEG *senseg-name* IN PCB *dbdname* IN PSB *psbname***

Explanation: The secondary index data base specified in the PROCSEQ operand in the indicated PCB is either incorrectly defined, or the index relationships and segment structure defined in the DBDs for the index and the data base do not match the segment structure defined in the SENSEG statements of the PCB.

User Response: Correct the DBD or PCB and reexecute the job.

DLZ9411 **PHYSICAL FIELD *fldname* NOT FOUND IN SEGMENT *segmname***

Explanation: A field name was supplied in the NAME=*fldname* parameter in the SENFLD macro statement at PSBGEN time which was not found in the corresponding field in the associated segment.

User Response: Either:

- Correct the name supplied in the NAME=*fldname* parameter in the SENFLD macro statement, or
- Add the field definition to the segment at DBDGEN time.

DLZ9421 **INCOMPLETE SENSITIVITY TO SEQ FIELD *fldname* IN SEGMENT *segmname***

Explanation: When inserting a segment, all sequence fields must be supplied including the fields defined for both the logical child and the virtual child in bi-directional relations as well as in the destination parent if logical or virtual insert rules are specified.

User Response:

- Specify sensitivity to the required field at PSBGEN time.
- Redesign and compile the application module to supply the identified field.

DLZ9431 ***t1* TO *t2* FIELD TYPE CONVERSION NOT SUPPORTED - SEGMENT *segmname*, FIELD *fldname***

Explanation: The type of field supplied in the TYPE=*t1* parameter in the SENFLD statement and the type of field supplied in the TYPE=*t2* parameter in the FIELD statement specify a conversion that is not supported by DL/I.

User Response: Either:

- Change one of the specifications for the TYPE=*t* parameter in the SENFLD or FIELD statement, or
- Supply a user field exit routine to do the conversion.

DLZ9441 **FIELD *fldname* IN SEGMENT *segmname* HAS SUBFIELDS IN USER VIEW AND NONE IN PHYSICAL VIEW**

Explanation: The identified field is defined as containing other fields in the view defined in the PSB. It has no subfields in the DBD view.

User Response: Either add the definition of the subfields to the physical view or delete the subfields from the user view.

DLZ9451 **PCB *dbdname* IN PSB *psbname* HAS NO SENSITIVE SEGMENTS**

Explanation: A PCB with no SENSEG statements was submitted.

User Response: Correct the PCB and perform a new PSB generation. Then reexecute the application control blocks utility.

DLZ9461 **FIELD *fldname* IN SEGMENT *segmname* INCOMPLETELY DEFINED BY SUBFIELDS**

Explanation: In the application view, a field must be completely defined by the subfields it contains (i.e., no 'holes') if one or more subfields exist or if the field overlaps another field (i.e., starts inside, ends outside).

User Response: Respecify the PSBGEN (and DBDGEN if necessary) to define sensitivity to the additional fields.

DLZ947I FIELD *fldname1*/*fldname2* INTERSECT IN SEGMENT *segmname* INCOMPLETELY DEFINED BY SUBFIELDS

Explanation: In the application view, the intersection between two fields must be completely defined (i.e., no 'holes') by a subfield or subfields of the intersecting fields.

User Response: Respecify the PSBGEN (and DBDGEN if necessary) to define sensitivity to the additional fields.

DLZ948I FIELD *fldname1* NOT A SUBFIELD OF FIELD *fldname2* IN BOTH VIEWS OF SEGMENT *segmname*

Explanation: *fldname1* and *fldname2* appear in both views of a segment and either *fldname1* is a subfield of *fldname2* in the physical view of the segment but not in the application view, or it is a subfield in the application view but not in the physical view.

User Response: Respecify either the DBDGEN or PCBGEN to ensure that *fldname1* is a subfield of *fldname2* in both views or that it is not a subfield in either view.

DLZ949I FIELD *fldname1* NOT A SUBFIELD OF *fldname2*/*fldname3* INTERSECT IN BOTH VIEWS OF SEGMENT *segmname*

Explanation: *fldname1*, *fldname2*, and *fldname3* appear in both views of the segment and *fldname1* is a subfield of both *fldname2* and *fldname3* in either the application or physical view but not both.

User Response: Respecify the PSBGEN or DBDGEN to ensure that *fldname1* is a subfield of both other fields in both views or that it is not in the intersection in either view.

DLZ952I ERROR RETURN FROM DLZDSEH0, CODE *rc*

Explanation: The issuing utility received an error return code from the work data set generator module DLZDSEH0. *rc* is replaced by one of the following codes:

rc	Explanation
04	WORKFIL file was assigned IGN (ignore).
08	Sort field size exceeded.
0C	GETVIS error occurred.
10	Invalid DL/I control blocks.
14	Length of a logical parent concatenated key is zero.
18	I/O error occurred on WORKFIL.
1C	CONTROL or WORKFIL file could not be opened.

User Response:

rc	User Response
04	The logical unit assignment for the WORKFIL file was IGN. Only during initial load of logical parent segments and when no work file is desired is this valid. Otherwise refer to return code 1C.

- 08 A dynamic limit check was performed as described under "Restrictions" in the introduction to "Part 7: Data Base Reorganization Utilities" in *DL/I DOS/VS Resource Definition and Utilities*. The limit check failed. Refer to the above referenced section for the appropriate action to be taken. The data base prereorganization utility (DLZURPR0) will have provided diagnostic messages during its execution that indicate the segments for which the above limit check failure occurred.
- 0C A GETVIS macro request failed. Ensure that a size parameter was specified on the execute card. If one was, determine that the value was the correct one for this execution. The partition size must be large enough to allow for the GETVIS space required.
- 10 DLZDSEH0 was attempting to locate DL/I control blocks for segments involved in a logical relationship. They could not be found. Verify that valid DMBs are available for the data base(s) being initially loaded, reloaded, or scanned.
- 14 Logical parent concatenated key length is defined as 0 bytes long. This could be caused by the field definition(s) specifying NAME=xx rather than NAME=(xx,SEQ,U).
- 18 The I/O error indicates that the job must be rerun using a different physical allocation for the WORKFIL file. The VSAM data bases should have been closed successfully.
- 1C The logical unit assignment for either the CONTROL or WORKFIL file was either missing or assigned to an unknown or unsupported device type. Refer to the *DL/I DOS/VS Resource Definition and Utilities* for the job control requirements of the DL/I utility that issued this message.

DLZ953I DATA BASE OPEN FAILURE, *dbdname*

Explanation: The issuing program received an error code from DLZDLOC0 while attempting to open a VSAM data base. The job is terminated.

User Response: Ensure that the correct DLBL and EXTENT cards were provided in the job control statements and that the data base was properly closed when it was loaded. An open error message was printed on SYSLOG which contains the VSAM return code.

DLZ954I INPUT CONTROL CARD FORMAT ERROR

Explanation: The issuing program found an input control card that was in error due to one of the following conditions:

- The control card identifier was missing or invalid.
- One or more options were invalid.
- The control card format was invalid.
- A file or segment named on a DBS control card was not found in the control file, or, if

found, was not listed as necessary to scan. The erroneous input control card is displayed preceding this message.

User Response: Correct the control card, and resubmit the job.

DLZ955I INVALID RECORD ON INPUT WORK FILE

One or more lines containing input record in hexadecimal and EBCDIC format.

Explanation: The issuing program found an input record from the input work file that was in error due to one of the following conditions:

- The record type code was invalid.
- One of the data base names or segment codes was invalid.
- An input record was read that, according to the utility control statement, should not occur, for example, the utility control statement specified L and an indexing record was found.
- A data base name or segment code could not be located in the control blocks available to DL/I.
- An input record was used to update a logical or index relationship for which no corresponding relationship was defined in the control blocks available to DL/I. Input record formats are described in the *DL/I DOS/VS Logic Manual, Volume 1*.

User Response: Ensure that the correct input file and/or control statement is submitted to the job. Reexecute the program which generated the work file that contained the record in error, then resubmit the job.

DLZ956I UNABLE TO LOAD DL/I CONTROL BLOCK - control-block-name

Explanation: The issuing program requested loading of the control block specified in the message text, but the request was not successfully completed. The job is terminated.

User Response: Ensure that a valid DMB exists for the named file and all referenced files. Resubmit the job.

DLZ957I MISSING OR INVALID CONTROL DATA SET

Explanation: The issuing program attempted to read the input control file generated by the prereorganization/load utility (DLZURPR0). One of the following conditions occurred:

- No control file was provided.
- The control file identifier was not present in the first block of the control file.
- The control file did not contain the number of blocks indicated in the first block of the control file.

User Response: Ensure that a valid control file is provided to the issuing program. Reexecute DLZURPR0, if necessary.

DLZ958I BUFFER HANDLER ERROR RETURN, rc

Explanation: The issuing program received a non-zero return code from the buffer handler. The *rc*, which can also be found at label PSTRTCDE of the PST DSECT, is an error code.

rc	Explanation
-----------	--------------------

04	RBA is beyond the end of the data set.
08	I/O error or permanent read error.
0C	No space in data set for additions.
10	An illegal call was made.
14	No record found (retrieve by key).
18	New block was created in buffer pool.
1C	Not enough space in buffer pool.
20	Size of requested buffer exceeded size of buffer in any subpool.
24	End of data set. No record returned.
28	Key or RBA higher than highest key or RBA in data set.
2C	End of data set is reached on a request issued by open.
34	Invalid request during data set loading.

User Response: From the code indicated, correct the error, and resubmit the job.

DLZ959I DL/I ERROR RETURN, code

Explanation: The issuing program received an unexpected return code after a DL/I call. See Chapter 6, "DL/I Status Codes" on page 111 for the meaning of the return code.

User Response: From the code indicated, correct the error, and resubmit the job.

DLZ960I DUPLICATE INDEX ENTRY

One or more lines containing the input record in hexadecimal and EBCDIC format.

Explanation: The issuing program found an input record from the input work file that attempted to create a duplicate index entry. Index entries must have unique sequence fields. Input record formats are defined in the *DL/I DOS/VS Logic Manual, Volume 1*.

User Response: Change the data base record that caused the duplicate entry creation to be attempted. Resubmit the job.

DLZ961I number input-control-card

Explanation: The issuing program is displaying an input control card provided for it. Numbers are assigned sequentially for reference by subsequent error messages that may be generated.

User Response: None.

DLZ962I DBS=dbdname,segmname

Explanation: This message may be issued more than once to provide a list of segments to be scanned by the data base scan utility (DLZURGS0). If an OPTIONS=(PUNCH) control card was provided to the program issuing the message, the portion of this message beginning with the characters DBS will be written to SYSPCH.

User Response: The data base scan utility (DLZURGS0) must be executed for each of the data bases indicated by the scan list provided by this message.

DLZ963I DUPLICATE NAME ON INPUT CONTROL CARD, *number*

Explanation: The issuing program found a file name that appeared on the preceding input control card indicated by *number*.

User Response: Correct the input control card preceding the message and resubmit job.

DLZ964I MISSING DBR AND/OR DBIL CONTROL CARDS

Explanation: The issuing program expected control cards with DBR or DBIL identifiers and found none.

User Response: Supply the required control cards and resubmit the job.

**DLZ965I UNABLE TO CONSTRUCT CONTROL LIST ENTRY,
[*file name1 segmname1 file name2 segmname2*]**

Explanation: The issuing program was attempting to construct a control list entry for the logical relationship that was defined to exist between the files and segments named above. The definition was not verified in the control blocks available to DL/I.

User Response: Ensure that valid DMBs are available to define the indicated logical relationship.

DLZ966I NORMAL PROGRAM TERMINATION [*number*]

Explanation: The issuing program terminated normally. *number*, if present, is the number of input work file records processed by the issuing program.

User Response: None.

DLZ967I UTILITY CHECKPOINT, *pgmname chkpt-number*

Explanation: The issuing program completed a checkpoint at one of the intervals you specified. The name of the program issuing the message and the checkpoint number are shown.

User Response: Record the checkpoint number in case a restart action must be requested.

DLZ968I BEGIN EXECUTION *pgmname date time*

Explanation: The program named is starting to execute. The date and time are also printed.

User Response: None.

DLZ969I SEGMENT NOT FOUND IN CONTROL BLOCKS, *segmname*

Explanation: The issuing program was unable to locate control blocks for the named segment. The segment was either listed on an input control card or on the input control file.

User Response: Ensure that valid DMBs are available for the supplied input control cards and/or control file. Resubmit the job.

DLZ970I SCAN PROCESSING STARTED, *dbdname*

Explanation: The issuing program started the scan of the named data base. This message is used by DLZURGS0.

User Response: None.

DLZ971I SCAN PROCESSING COMPLETED

Explanation: The issuing program completed the scan of the data base named by the preceding DLZ970I message.

User Response: None.

DLZ972I DATA BASE NOT SCANNED, *dbdname*

Explanation: The issuing program found one or more of the following errors when checking the scan list for the named data base:

- The named data base was also listed in the control file as being initially loaded or reorganized.
- One or more segments were not found in the control blocks available to DL/I, or, if found, did not participate in any logical relationships with segments in data bases listed as being initially loaded or reorganized.

User Response: Ensure that valid DMBs are available for the supplied input control cards and/or control file.

DLZ973I SEGMENT NOT SCANNED, *dbdname segmname*

Explanation: The issuing program found one or more of the following errors when checking the control blocks for the named segment:

- The named segment was not found in the control blocks for the named data base.
- The named segment does not participate in any logical relationships with any of the segments in the data bases listed in the control file as being initially loaded or reorganized.

User Response: Ensure that valid DMBs are available for the supplied control cards and/or control file.

DLZ974I UNABLE TO COMPLETE RESTART

Explanation: The issuing program was unable to complete a restart action for one of the following reasons:

- A restart file was not specified.
- A restart point could not be located in the data bases that were made available to the issuing program.
- The specified checkpoint record could not be found in the restart file.

User Response: Ensure that a restart file is available and that the data bases are the same as they were when the previous program operation was interrupted. Refer to *DL/I DOS/VS Resource Definition and Utilities* for instruction regarding the use of checkpoint/restart facilities.

DLZ975I RESTART COMPLETED, *pgmname* *chkpt-number*

Explanation: The issuing program completed a requested restart action using the checkpoint number indicated in the message. Refer to the section of the *DL/I DOS/VS Resource Definition and Utilities* pertaining to the program issuing this message.

User Response: None.

DLZ976I SPECIFICATION OF INDEX DBD IS NOT ALLOWED. DBDNAME = *dbdname*

Explanation: The name of a secondary index DBD was specified. This is invalid and unnecessary.

User Response: Remove this DBD name from the input and rerun the utility program.

DLZ977I DUPLICATE RECORD FOR LOGICAL PARENT
One or more lines containing input record in hexadecimal and EBCDIC format.

Explanation: The issuing program found a duplicate type-00 input work file record of a logical parent/logical child pair. Only one type-00 record can exist for an occurrence of a logical parent/logical child pair. Refer to the *DL/I DOS/VS Logic Manual, Volume 1* for a description of the record format.

Prefix resolution may be invalid if this message is issued. **The affected data base(s) must not be used.**

This condition could be caused by supplying (as input to the issuing program) work files that were generated by multiple runs of a data base initial load, reorganization, or scan.

User Response: Ensure that only work files from the data base(s) being initially loaded, reorganized, or scanned are supplied as input to the issuing program.

DLZ978I CAUTION -- NO LOGICAL CHILD RECORD FOR LOGICAL PARENT RECORD

One or more lines containing input records in hexadecimal and EBCDIC format.

Explanation: The issuing program expected to find one or more type-10 records for the type-00 input work file record displayed. A type-00 record is generated for a logical parent. A type-10 record is generated for each of its logical children. Since it is not necessary that logical children actually occur for each logical parent, **this message is issued as a caution only.** Prefix resolution may be invalid if this message is issued. It will also be invalid if you intended to load or reload the logical children belonging to the logical parent.

User Response: This condition could be caused by not supplying as input to the issuing program all work files that were generated by the data bases currently being initially loaded, reorganized, or scanned. Ensure that all work files are supplied as input.

Note: To suppress this message, specify the SUMM option of the OPTION control statement when you execute the data base prereorganization utility. This will cause the data base resolution utility to print counts of logical relationships rather than to print individual warning messages for each unmatched logical relationship. Refer to "Chapter 20: Data Base Prereorganization Utility" in *DL/I DOS/VS Resource Definition and Utilities* for information about the OPTIONS control statement.

DLZ979I NO LOGICAL PARENT RECORD FOUND FOR LOGICAL CHILD RECORD

One or more lines containing input record in hexadecimal and EBCDIC format.

Explanation: The issuing program expected to find a type-00 record for the type-10, -20, or -30 input work file record displayed. A type-00 record is generated for a logical parent. A type-10 record is generated for each of its logical children. Type-20 and type-30 records are generated for the logical twin forward and backward pointers, respectively, of each logical child. Refer to the *DL/I DOS/VS Logic Manual, Volume 1* for description of the record formats. Prefix resolution will be invalid if this message is issued. **The affected data base(s) must not be used.**

User Response: Same action as specified in message DLZ978I.

DLZ980I NO LOGICAL CHILD RECORD FOUND FOR LOGICAL TWIN RECORD

One or more lines containing input record in hexadecimal and EBCDIC format.

Explanation: The issuing program expected to find a type-10 record for the type-20 or -30 input work file record displayed. Type-10, -20, and -30 records are generated as indicated in the explanation of message DLZ979I. Refer to the *DL/I DOS/VS Logic Manual, Volume 1* for description of the record formats. Prefix resolution

will be invalid if this message is issued. **The affected data base(s) must not be used.**

User Response: Same action as specified in message DLZ978I.

DLZ981I **DUPLICATE RECORD FOR LOGICAL CHILD**
One or more lines containing input record in hexadecimal and EBCDIC format.

Explanation: The issuing program found an input work file record that was for the same occurrence of a logical child as a previous input work file record. Only one type-10 record can exist for an occurrence of a logical child. Refer to the *DL/I DOS/VS Logic Manual, Volume 1* for description of the record format. Prefix resolution may be invalid if this message was issued. **The affected data base(s) must not be used.**

User Response: Same action as specified for message DLZ977I.

DLZ982I **ABNORMAL PROGRAM TERMINATION, number [code]**

Explanation: The issuing program detected one or more error conditions during its operation:

- If the message was issued by the prefix update utility, *number* is the number of input work file records processed.
- If the message was issued by the prefix resolution utility, *code* is only included in the message text if the error was detected by a sort phase.

number indicates in which sort phase the last-detected error occurred.

code indicates the sort/merge return code.

Refer to the *DOS/VS-VM/System Product Sort/Merge Version 2 Application Programming: Guide* for an explanation of return code *code* of sort/merge phase *number*.

Prefix resolution may be invalid if this message is issued. **The affected data base(s) must not be used.**

User Response: Determine the cause of the error and take appropriate action.

DLZ983I **KSDS KEY NOT FOUND**
One or more lines containing input record in hexadecimal and EBCDIC format.

Explanation: The issuing program attempted to retrieve a record stored in a KSDS file and no matching key could be found. The type-40 input file record displayed in the message contains the KSDS key in error. Refer to the *DL/I DOS/VS Logic Manual, Volume 1* for a description of the record format.

User Response: Ensure that the data base indicated by the input record is correctly referenced by its DLBL and EXTENT cards and that it has been loaded properly.

DLZ984I **INVALID DEVICE ASSIGNMENT FOR DATA SET - *dtfname***

Explanation: The issuing program determined that the named file was incorrectly assigned.

User Response: Check that the correct SYSnnn was specified, and that the device to which the file was assigned is one that is supported by the program (for example, if a tape was assigned when only disks are supported). Correct and reexecute the job.

DLZ985I **CAUTION -- LIMIT CHECK FAILURE, *dbdname1 segmname1 dbdname2 segmname2***

Explanation: The issuing program detected a failure in the limit check for the logical parent/logical child indicated by the data bases and segments named above. The limit check performed is as described under "Restrictions" in the introduction to "Part 7: Data Base Reorganization Utilities" in *DL/I DOS/VS Resource Definition and Utilities*. This message is cautionary only, since the issuing program assumes worst case in computing the limit check.

User Response: Determine the cause of limit check failure. If none of the components of the limit check can be omitted, you should consider loading the logical parent and/or logical child with an update program, or redefining the affected segments so that one or more of the limit check components can be omitted.

DLZ989I **NO LOGICAL TWIN POINTERS FOR MULTIPLE LOGICAL CHILD RECORDS**

Explanation: The issuing program found a second logical child record for a logical parent record, but no logical twin pointers were specified for the logical child segment type. Refer to the *DL/I DOS/VS Logic Manual, Volume 1* for a description of the record format. Prefix resolution will be continued but the LP will point only to the last LC found. All other logical children will be lost through the logical parent path, although they are still accessible through the physical path and have proper pointers to the logical parent.

User Response: You should determine if this condition is to your specification. If it is, no further action is required. If not, make corrections and rerun the prefix resolution utility. A possible correction may be removal of input data which caused more than one LC to point to the same LP or a new DBDGEN to include LT pointers in the LC segment.

DLZ990I **ERROR ACCESSING FILE - *dtfname***

Explanation: An unrecoverable I/O error was detected for the named file through an error exit in the DTF.

User Response: If an output file, reallocate the file to a different extent or use another tape and/or tape unit (if applicable). Then reexecute the job. If an input file, the program that created it, must

be rerun before this job is reexecuted. Any VSAM data bases should have been closed.

DLZ991I INVALID COMBINATION OF DBIL, DBR OR DBS CONTROL CARDS

Explanation: The pre-reorganization utility DLZURPR0 found a combination of DBIL, DBR, or DBS control statements that implied a logical parent segment was to be initially loaded and its logical child reloaded. This is an invalid situation because a logical child segment cannot exist without its logical parent.

User Response: The utility runs to completion, but makes no entry in the control data set for the invalid logical relationship combinations. The

control data set is built correctly, however, for all other logical relationships found.

Check your control statements and resubmit the job after correcting any errors.

DLZ992I NO PREFIX UPDATE ALLOWED BEFORE END OF PARTIAL LOAD FOR *dbdname*

Explanation: Prefix update is allowed only if the load process of the data base is finished, by doing a load step with parameter card start PLC.

User Response: Do a last load step with parameter card PLC to finish your partial load process. If all segments are already inserted, insert a dummy segment, which you can delete later. Then redo prefix resolution and prefix update.

DL/I System/Task Abnormal Termination

The DL/I messages that refer to this section (that is, system or task termination messages) are issued when DL/I internal validity checking routines detect contradictory conditions.

- If DL/I was running in a batch environment when the condition was detected, the message text indicates that the DL/I system terminated abnormally.
- If DL/I was running in an online environment when the condition was detected, the message text indicates that the DL/I task terminated abnormally.

In either case, possible causes for the contradictory conditions are:

- Application program and application control blocks do not match.
- Erroneous control information in data bases.
- Destroyed DL/I control blocks or buffers.
- Undetected program errors in DL/I.

A storage dump is produced by these abnormal terminations.

Unless otherwise indicated, make sure that:

- The correct ACBs are being used with the application.
- The application was not working with a previously destroyed (not backed-out or not recovered) data base.
- The application program did not destroy DL/I code or control blocks.

In any case, you should execute the data base backout utility to guarantee that the data bases involved are restored to a valid status.

CICS/VS Transaction ABEND Codes

If a DL/I task abnormal termination occurs during online processing, control is not returned to the application program and the transaction is terminated with a CICS/VS message.

In that CICS/VS message, you will find a four-character ABEND code *code* following the word ABEND:

code	Explanation
<i>Dnnn</i>	Refer to DL/I error message DLZ <i>nnn</i> for an explanation of this ABEND code.
DACT	A program that was not defined in the ACT issued a TERM call.
DBPC	DLZBPC00 (MPS batch partition controller task) terminated abnormally.
DHxx	An application program using the HLPI terminated abnormally: <i>xx</i> is the corresponding DL/I status code (refer to "DL/I PCB Status Codes" on page 113). In the CICS/VS online environment, you can intercept status code ABENDs by coding: EXEC CICS HANDLE ABEND at the beginning of your program. Your ABEND exit routine can then attempt to correct the error and continue processing. Abends cannot be intercepted in the batch and MPS batch environments.
DLPV	A system scheduling call was issued with an invalid password.
DMPC	DLZMPC00 (MPS master partition controller task) terminated abnormally.
<i>Ennn</i>	Refer to DL/I error message DLZ <i>nnn</i> for an explanation of this ABEND code. In this case the termination cannot be noted on the transient data destination CSMT.

CICS/VS Dump Codes

The following codes are used to identify CICS/VS dumps taken by DL/I:

Code	Explanation
DLI, DLI2	Identify the CICS/VS dump as one taken by DLZFTDP0 (DL/I Formatted Task Dump Program). DLI identifies the first block of data dumped. DLI2 identifies the remaining blocks of data dumped.
DLII	Issued by program DLZODP when a task with no PST acquired ends abnormally and no active PPST can be found which contains its TCA address.

Chapter 2. Data Base Description Generation (DBDGEN) Error Messages

This chapter lists the DBD generation error conditions and the messages displayed for these conditions.

As DBD generation is composed of VSE Assembler language macro instructions, error statements from the assembler can also occur because of omissions, invalid sequence in DBD control cards, or invalid keyword parameters.

ACCnnn (ACCESS Macro)

ACC001 'REF' AND 'RMRTN' MAY NOT BOTH BE SPECIFIED ON ONE STATEMENT. 'REF' OPERAND IGNORED, 'RMRTN' OPERAND USED.

Explanation: The RMRTN parameter is used exclusively for definition of randomizing routine access and the REF parameter is used exclusively for the definition of indexed access. Both may not be specified on the same ACCESS statement. This is a warning message only.

User Response: Remove either the RMRTN or REF parameter.

ACC002 'REF' AND 'RMRTN' MAY NOT BOTH BE SPECIFIED ON ONE STATEMENT. 'RMRTN' OPERAND IGNORED, 'REF' OPERAND USED.

Explanation: The RMRTN parameter is used exclusively for definition of randomizing routine access and the REF parameter is used exclusively for the definition of indexed access. Both may not be specified on the same ACCESS statement. This is a warning message only.

User Response: Remove either the RMRTN or REF parameter.

ACC003 ACCESS NOT ALLOWED FOR *org* ORGANIZATION.

Explanation: ACCESS may be used to define external access to a data base for HD.

User Response: Either delete the ACCESS statement or change the organization to HD.

ACC004 MORE THAN 255 ACCESES SPECIFIED.

Explanation: A maximum of 255 external access points may be defined for a single data base.

User Response: Either reduce the number of defined external access points or split the data base into two or more separate data bases connected via logical relations.

ACC006 EITHER 'REF' OR 'RMRTN' MUST BE SPECIFIED.

Explanation: The ACCESS statement is used to define either randomized or indexed access to a data base. The RMRTN parameter is required for randomized access and the REF parameter is required for indexed. At least one must be specified.

User Response: Specify either a REF or a RMRTN parameter.

ACC007 INDEX TABLE LIMIT OF 4095 EXCEEDED.

Explanation: A total of 4095 index related values (sequence fields, suppression mechanisms, duplicate value specification) is allowed.

User Response: Either reduce the total access parameters or split the data base into two or more parts.

ACC101 INVALID CHARACTER 'c' IN POSITION 1 OF SEGM OPERAND.

Explanation: The initial character in a segment name must be alphanumeric.

User Response: Specify a proper segment name.

ACC102 SEGM OPERAND OMITTED.

Explanation: The segment being accessed must be defined via the SEGM operand.

User Response: Specify the name of the accessed segment.

ACC103 SEGM OPERAND LONGER THAN EIGHT CHARACTERS.

Explanation: A segment name must be from one to eight characters long.

User Response: Specify a proper segment name.

ACC201 INVALID CHARACTER 'c' IN POSITION *n* OF RMRTN OPERAND

Explanation: The name of the randomizing routine must be from one to eight characters long, the first character alphabetic, the remainder alphanumeric.

User Response: Correct the operand.

ACC202 FIRST CHARACTER OF RMRTN OPERAND NOT ALPHABETIC

Explanation: The name of the randomizing routine must be from one to eight characters long, the first character alphabetic, the remainder alphanumeric.

User Response: Correct the operand.

ACC203 RMRTN PARAMETER LONGER THAN EIGHT CHARACTERS

Explanation: The name of the randomizing routine must be from one to eight characters long, the first character alphabetic, the remainder alphanumeric.

User Response: Correct the parameter.

ACC204 ONLY ONE RANDOMIZED ACCESS ALLOWED

Explanation: Only one access to a physical data base may be by randomizing routine.

User Response: Delete either this or the other ACCESS statement specifying the RMRTN parameter or change this or the other one to indexed access by specifying REF instead of RMRTN.

ACC301 SEQFLD OPERAND LONGER THAN EIGHT CHARACTERS

Explanation: Field names may not be longer than eight characters.

User Response: Redefine the field name to be eight characters or less.

ACC302 SEQFLD OPERAND *n* LONGER THAN EIGHT CHARACTERS

Explanation: Field names may not be longer than eight characters.

User Response: Redefine the field name to be eight characters or less.

ACC303 SEQFLD OPERAND OMITTED.

Explanation: A field must be defined in the referenced segment for randomized and indexed access or in one of the physical dependents of the referenced segment for secondary access to be used to determine the order of sequential access.

User Response: Supply a SEQFLD operand.

ACC304 SEQFLD OPERAND *n* OMITTED

Explanation: Multiple suboperands were specified for the SEQFLD operand and two commas appeared in the list contiguous to each other.

User Response: Remove the extra comma.

ACC305 MULTIPLE SEQUENCE FIELDS NOT VALID FOR RANDOMIZED ACCESS.

Explanation: The concatenation of multiple fields to form a composite sequence field is only supported for secondary indexed access.

User Response: Either switch the access type to indexed or reduce the number of sequence fields specified to one.

ACC306 MORE THAN FIVE SEQUENCE FIELDS SPECIFIED.

Explanation: The maximum number of fields that may be concatenated for the sequence field of a secondary index is five.

User Response: Reduce the number of specified fields. Moving fields currently non-contiguous in the physical data base to

contiguous locations and defining a third field containing them would be a method of reduction. Field Level Sensitivity may be used to avoid any modification of applications dependent on the current field positions.

ACC307 {/CK/}SPECIAL FIELD TYPE MAY NOT BE SEQUENCE FIELD FOR ACCESS

Explanation: Field names starting with /CK or /SX are reserved in DL/I to reference the segment's concatenated key or RBA respectively. These values may not be used as the sequence value for access.

User Response: Redefine the field name to use a different name.

ACC401 SEQVAL OPERAND NOT 'UNIQUE', 'U', 'DUPLICATE' OR 'D'. 'DUPLICATE' ASSUMED

Explanation: The SEQVAL parameter is used to indicate whether the individual values of the sequence field are all unique or that duplicates may exist.

User Response: Specify a valid operand for the SEQVAL parameter or omit it.

ACC402 SEQVAL OPERAND NOT 'UNIQUE', 'U', 'DUPLICATE' OR 'D'. 'UNIQUE' ASSUMED

Explanation: The SEQVAL parameter is used to indicate whether the individual values of the sequence field are all unique or that duplicates may exist.

User Response: Specify a valid operand for the SEQVAL parameter or omit it.

ACC501 INVALID CHARACTER 'c' IN POSITION *n* OF REF OPERAND

Explanation: The reference name must be from one to seven characters in length, the first character alphabetic, the remainder alphanumeric.

User Response: Supply a valid reference name.

ACC502 FIRST CHARACTER OF REF OPERAND NOT ALPHABETIC

Explanation: The reference name must be from one to seven characters in length, the first character alphabetic, the remainder alphanumeric.

User Response: Supply a valid reference name.

ACC503 REF OPERAND LONGER THAN SEVEN CHARACTERS

Explanation: The reference name must be from one to seven characters in length, the first character alphabetic, the remainder alphanumeric.

User Response: Supply a valid reference name.

ACC504 REFERENCE NAME '*name*' INVALID. EQUAL TO PREVIOUSLY DEFINED REFERENCE NAME OR DATA BASE NAME.

Explanation: The reference name will be used as the data base name for the index data base and must be unique.

User Response: Define a unique name for this access.

ACC601 INVALID CHARACTER '*c*' IN POSITION 1 OF SEQSEG OPERAND

Explanation: The initial character in a segment name must be alphanumeric.

User Response: Specify a proper segment name.

ACC602 SEQSEG OPERAND LONGER THAN EIGHT CHARACTERS.

Explanation: A segment name must be from one to eight characters long.

User Response: Specify a proper segment name.

ACC603 SEQSEG SPECIFICATION NOT VALID FOR RANDOMIZED ACCESS.

Explanation: The field used for sequencing for randomized access must be in the segment accessed.

User Response: Either change to secondary indexed access or use a field in the accessed segment for the sequence field.

ACC703 SUPVAL OPERAND NOT A VALID SELF-DEFINING TERM

Explanation: Valid self-defining terms are one of four formats:

- *n*
- *C'c'*
- *X'hh'*
- *B'bbb...bb'*

where:

- *n* is a decimal number ranging from zero to 255.
- *c* is any character.
- *h* is an alphanumeric character from 0 to 9 or from A to F.
- *b* is 0 or 1.

User Response: Specify a valid self-defining term, 'ZERO', 'BLANK', or omit the parameter.

ACC704 SUPVAL OPERAND CONTAINS MORE THAN ONE BYTE OF INFORMATION

Explanation: The suppression value is limited to one byte of information: one character, two hex digits, eight binary digits, or a decimal number in the range zero to 255.

User Response: Either reduce the value to a single byte of information or omit it.

ACC705 SUPVAL SPECIFICATION NOT VALID FOR RANDOMIZED ACCESS.

Explanation: The SUPVAL parameter is used to suppress creation of indexes and has no meaning for randomized access.

User Response: Either change to secondary indexed access or omit the parameter.

ACC801 INVALID CHARACTER '*c*' IN POSITION *n* OF SUPRTN OPERAND

Explanation: The suppression routine name must be from one to eight characters in length, the first character alphabetic, the remainder alphanumeric.

User Response: Specify a valid suppression routine name.

ACC802 FIRST CHARACTER OF SUPRTN OPERAND NOT ALPHABETIC

Explanation: The suppression routine name must be from one to eight characters in length, the first character alphabetic, the remainder alphanumeric.

User Response: Specify a valid suppression routine name.

ACC803 SUPRTN OPERAND LONGER THAN EIGHT CHARACTERS.

Explanation: The suppression routine name must be from one to eight characters in length, the first character alphabetic, the remainder alphanumeric.

User Response: Specify a valid suppression routine name.

ACC804 SUPRTN SPECIFICATION NOT VALID FOR RANDOMIZED ACCESS.

Explanation: Suppression routines are used to suppress creation of indexes and have no meaning for randomized access.

User Response: Either change to secondary indexed access or omit the parameter.

DBDnnn (DBD Macro)

DBD001 DBD STATEMENT MUST OCCUR FIRST IN A DBDGEN

Explanation: The first statement in a data base definition must be a DBD statement.

Note: Message DBD001 can occur during assembly of any of the DBD generation macros.

User Response: Supply a DBD statement preceding the DBD generation macro statement that caused the message.

DBD100 MORE THAN ONE DBD STATEMENT SPECIFICATION

Explanation: Only one DBD may be defined during a single assembly.

User Response: Either remove the extra DBD statement or define the additional data bases in separate assemblies.

DBD110 NAME OPERAND IS OMITTED OR INVALID

Explanation: The NAME operand must be specified as one to seven alphanumeric characters in length, with the first character alphabetic.

Note: The at-sign (@) must not be used in the NAME operand.

User Response: Provide a proper NAME operand.

DBD120 ACCESS OPERAND IS OMITTED OR INVALID. DEFAULTED TO {HDAM|HIDAM}

Explanation: The ACCESS operand has been omitted or the entered value is not one of the supported access methods. Valid values are SHSAM, HSAM, SHISAM, HISAM, HD, HDAM, HIDAM, INDEX, or LOGICAL.

User Response: Correct the ACCESS operand and rerun DBDGEN.

DBD130 RMNAME ROUTINE NAME SUB-OPERAND 'name' IS INVALID

Explanation: The first suboperand of the RMNAME parameter, the randomizing routine *name*, must be one to eight characters in length with the first character alphabetic.

User Response: Supply a valid routine name.

DBD131 RMNAME ROUTINE NAME SUB-OPERAND IS OMITTED

Explanation: If the RMNAME parameter is specified, the first suboperand, the routine name, may not be omitted.

User Response: Supply a valid routine name.

DBD132 RMNAME PARAMETER ONLY VALID FOR HDAM ORGANIZATION. IGNORED

Explanation: The RMNAME parameter may only be specified if the organization is HDAM.

User Response: Redefine either with ACCESS=HDAM or without the RMNAME parameter.

DBD133 RMNAME ROOT ANCHOR POINTS SUB-OPERAND 'parm' IS INVALID

Explanation: The second RMNAME suboperand, the root anchor points per control interval parameter, must be numeric and in the range of one to 255.

User Response: Either omit the sub-parameter or specify a valid value.

DBD134 RMNAME PRIMARY CONTROL INTERVALS SUB-OPERAND 'parm' IS INVALID

Explanation: The third RMNAME suboperand, indicating the number of control intervals in the root addressable area, must be numeric and greater than zero.

User Response: Either omit the subparameter or specify a valid value.

DBD135 RNAME MAXIMUM INSERTABLE BYTES SUB-OPERAND 'parm' IS INVALID

Explanation: The fourth RMNAME suboperand, indicating the maximum number of bytes insertable in the root addressable area by an uninterrupted series of inserts for the same data base record, must be numeric and greater than zero.

User Response: Either omit the suboperand or specify a valid value.

DBD136 MAXIMUM OF FOUR SUB-OPERANDS ALLOWED FOR RMNAME PARAMETER. REMAINDER IGNORED

Explanation: More than four suboperands have been specified.

User Response: Redefine with the proper number of suboperands.

DBD137 RMNAME PARAMETER IS OMITTED.

Explanation: The RMNAME operand was specified for a DL/I access method other than HDAM.

User Response: Correct the RMNAME operand and rerun DBDGEN.

DBD140 'IMSCOMP' PARAMETER '*parm*' INVALID. 'YES' ASSUMED

Explanation: The only valid values for the IMSCOMP parameter are 'YES' or 'NO'.

User Response: Specify YES or NO, or omit the IMSCOMP parameter.

DBD501 CIANPT OPERAND NOT NUMERIC

Explanation: The CIANPT operand is used to specify the number of root anchor points per control interval and must be numeric.

User Response: Either supply a valid value for the CIANPT operand or omit it.

DBD502 CIANPT OPERAND GREATER THAN 255. 255 ASSUMED

Explanation: The CIANPT operand is used to specify the number of root anchor points per control interval and must be less than or equal to 255. This is a warning message only.

User Response: Either supply a valid value for the CIANPT operand or omit it.

DBD503 CIANPT OPERAND LESS THAN 1. 1 ASSUMED

Explanation: The CIANPT operand is used to specify the number of root anchor points per control interval and must be greater than zero. This is a warning message only.

User Response: Either supply a valid value for the CIANPT operand or omit it.

DBD504 CIANPT OPERAND NOT VALID FOR *org* ACCESS. IGNORED

Explanation: The number of root anchor points per control interval may only be specified for HD access organization.

User Response: Remove the CIANPT operand or change the organization.

DBD601 PRIMCI OPERAND MUST BE EITHER NUMERIC OR 'OVERLAP'

Explanation: The PRIMCI (number of VSAM control intervals allocated to the root addressable area) parameter must be either numeric or the characters 'OVERLAP', indicating that the prime and overflow areas are to coincide.

User Response: Either supply a valid value or omit the operand.

DBD602 PRIMCI OPERAND GREATER THAN 16777215. 16777215 ASSUMED

Explanation: The maximum number of control intervals reserved for the root addressable area is 16777215. This is a warning message only.

User Response: Either supply a number less than or equal to 16777215 or omit the operand.

DBD603 PRIMCI OPERAND LESS THAN 1

Explanation: The minimum number of control intervals reserved for the root addressable area is 1.

User Response: Either supply a number greater than or equal to 1 or omit the operand.

DBD604 PRIMCI OPERAND NOT VALID FOR *org* ACCESS. IGNORED

Explanation: The number of VSAM control intervals reserved for the root addressable area may only be specified for HD access organization.

User Response: Remove the PRIMCI operand or change the access organization.

DBD701 RILIM OPERAND MUST BE NUMERIC

Explanation: The RILIM operand is used to specify the maximum number of bytes to be stored in the primary area by a single contiguous series of inserts and must be a numeric value.

User Response: Either supply a valid value for the RILIM operand or omit it.

DBD702 RILIM OPERAND GREATER THAN 16777215. 16777215 ASSUMED

Explanation: The maximum value for the RILIM operand is 16777215. This is a warning message only.

User Response: Either supply a number less than or equal to 16777215 or omit the operand.

DBD703 RILIM OPERAND LESS THAN 1

Explanation: The minimum number of bytes to be stored in the primary area by a single contiguous series of inserts is 1.

User Response: Either supply a number greater than or equal to 1 or omit the operand.

DBD704 RILIM OPERAND NOT VALID FOR *org* ACCESS. IGNORED

Explanation: The maximum number of bytes to be stored in the primary area by a single contiguous series of inserts may only be specified for HD access organization.

User Response: Remove the RILIM operand or change the access organization.

DGENnnn (DBDGEN Macro)

DGEN010 DBD CONTAINS NO SEGMENTS.

Explanation: No SEGM statements were supplied.

User Response: Either supply one or more segment definitions or move the DBDGEN statement to its proper location.

DGEN020 NO PRIMARY ACCESS SPECIFIED FOR HD ORGANIZATION.

Explanation: Either no ACCESS statement was supplied for the root (first) segment or the first ACCESS for the segment had characteristics that precluded it from being a primary index.

Possible exempting characteristics are suppression routine or character specified, sequence segment other than target segment, multiple sequence fields, or duplicated sequence values.

User Response: Supply a valid ACCESS specification for the root segment, redefine the initial ACCESS and drop the restricted parameters, or define an additional ACCESS as primary.

DGEN021 NO PRIMARY INDEX SPECIFIED FOR HIDAM ORGANIZATION.

Explanation: No LCHILD statement with POINTER=INDX was supplied for the root (first) segment.

User Response: Supply a valid primary index definition for the root segment.

DGEN023 RANDOMIZED ACCESS MAY ONLY BE SPECIFIED FOR THE ROOT SEGMENT WHICH MUST BE THE FIRST SEGMENT DEFINED.

Explanation: An ACCESS statement was supplied specifying access via a randomizing routine and the target segment was not the root (first) segment.

User Response: Either make the referenced segment the data base root, switch to access via a secondary index, or split the data base into two or more smaller data bases connected via logical relations, with the referenced segment the root in one of them.

DGEN024 ROOT SEGMENT HAS NO SEQUENCE FIELD.

Explanation: No field was identified as the sequence field for the root (first) segment. The sequence field is required to define the sequencing for the primary access path.

User Response: Identify one of the fields for the root segment as the sequence field.

DGEN030 NO DEFINITION OF INDEXED DATA BASE PRESENT.

Explanation: The data base organization was specified as INDEX, but no LCHILD was specified to identify the index target segment and the indexed data base.

User Response: Either supply a LCHILD to identify the indexed data base and target segment or specify an organization other than INDEX.

DGEN090 DBDGEN TERMINATED. ERRORS IN DBD.

Explanation: Errors have been detected (see previous MNOTES) that preclude generation of a valid DBD.

User Response: Correct previous errors and rerun the DBD generation.

DGEN100 PRIMARY BLOCK SIZE (*n*) EXCEEDS MAXIMUM (*m*).

Explanation: The block size for the SHISAM or primary HISAM VSAM data set was greater than the maximum control interval size (4096).

User Response: Either reduce the record size or the blocking factor.

DGEN101 OVERFLOW BLOCK SIZE (*n*) EXCEEDS MAXIMUM (*m*).

Explanation: The block size for the HISAM overflow VSAM data set was greater than the maximum control interval size (4096).

User Response: Either reduce the record size or the blocking factor.

DGEN110 PRIMARY BLOCKING FACTOR RESET TO *n*.

Explanation: The specified blocking factor for the SHISAM or primary HISAM data set was either too large, causing MNOTE DGEN100, or too small and more logical records would fit in a control interval than were specified. The blocking factor was reset.

User Response: Omit the parameter and allow DL/I to calculate the blocking factor.

DGEN111 OVERFLOW BLOCKING FACTOR RESET TO *n*.

Explanation: The specified blocking factor for the HISAM overflow data set was either too large, causing MNOTE DGEN101, or too small and more logical records would fit in a control interval than were specified. The blocking factor was reset.

User Response: Omit the parameter and allow DL/I to calculate the blocking factor.

DGEN130 PRIMARY RECORD LENGTH CANNOT EXCEED 4088. RESET TO 4088.

Explanation: The specified record length for the SHISAM or primary HISAM data set exceeded the maximum size of a VSAM control interval (4096) less the length of the VSAM information stored in the control interval. The record length was reset.

User Response: Either specify a record length less than or equal to 4088 or omit the parameter and have DL/I calculate the required logical record length.

DGEN131 OVERFLOW RECORD LENGTH CANNOT EXCEED 4088. RESET TO 4088.

Explanation: The specified record length for the HISAM overflow data set exceeded the maximum size of a VSAM control interval (4096), less the length of the DL/I information stored in the control interval. The record length was reset.

User Response: Either specify a record length less than or equal to 4088, or omit the parameter and have DL/I calculate the required logical record length.

DGEN132 OVERFLOW RECORD LENGTH LESS THAN PRIMARY RECORD LENGTH. RESET TO PRIMARY RECORD LENGTH (n).

Explanation: The record length specified for the HISAM overflow data set was less than the one either specified or calculated for the primary data set.

User Response: Either specify a length less than or equal to the primary record length *n*, or omit the value and have DL/I calculate the proper length.

DGEN200 MAXIMUM OF 15 LEVELS EXCEEDED - SEGMENT 'segmname'.

Explanation: The named segment was defined with fifteen or more segments above it in the hierarchy.

User Response: Split the data base into two or more smaller data bases connected by logical relations, or redefine the current structure combining levels to insure that the structure is nowhere more than fifteen segment types deep.

DGEN210 SEGMENT 'segmname1' NOT INCLUDED IN HIERARCHY BEGINNING WITH ROOT SEGMENT 'segmname2'.

Explanation: No hierarchical path exists connecting *segmname1* to *segmname2*. All segments in a single data base other than the root must be dependents of the root.

User Response: Either connect the hierarchy containing *segmname1* to the hierarchy topped by

segmname2, or define the two hierarchies as two separate data bases.

DGEN220 SEGMENT 'segmname1' SPECIFIED AS PARENT OF SEGMENT 'segmname2' IS NOT DEFINED.

Explanation: The SEGM statement defining segment *segmname2* identified *segmname1* as its parent and no SEGM statement was included for *segmname1*.

User Response: Provide a SEGM statement defining *segmname1* or correct the name of the parent on the SEGM statement for *segmname2*.

DGEN230 LOGICAL CHILD SEGMENT 'segmname1' IS A PHYSICAL CHILD OF LOGICAL CHILD SEGMENT 'segmname2'.

Explanation: A logical child segment may not be the physical parent of a logical child segment.

User Response: Define a dummy segment in the hierarchy between the two logical children.

DGEN231 SEGMENT 'segmname1' IS A PHYSICAL CHILD OF VIRTUAL LOGICAL CHILD SEGMENT 'segmname2'.

Explanation: Virtual logical child segments cannot have physical children.

User Response: Define the segments as physical children of the logical child with which the virtual logical child is paired.

Another possibility, if the real logical child has no physical children, would be to switch the two, making the real virtual and the virtual real.

DGEN240 REFERENCED PAIRED SEGMENT 'segmname' DOES NOT EXIST.

Explanation: The segment *segmname* was identified as a paired segment on a LCHILD statement for a logical parent segment, but no SEGM defining *segmname* is present.

User Response: Either provide a SEGM statement defining *segmname* as a virtual logical child, or correct the name specified on the LCHILD statement.

DGEN241 REFERENCED PAIRED SEGMENT 'segmname' IS NOT A VIRTUAL LOGICAL CHILD.

Explanation: The segment was identified as a paired segment on a LCHILD statement for a logical parent segment, but did not specify a SOURCE parameter on its SEGM statement.

User Response: Either specify the SOURCE parameter on the SEGM statement for *segmname* or correct the specified name on the LCHILD statement.

DGEN242 VIRTUAL LOGICAL CHILD SEGMENT 'segmname' NOT IDENTIFIED AS PAIRED SEGMENT FOR LOGICAL PARENT.

Explanation: The segment was identified as being paired, but no LCHILD is present for its physical parent identifying the segment as paired.

User Response: Either supply a LCHILD statement for the physical parent of the virtual logical child identifying it as being paired, or remove the POINTER=PAIRED specification from the SEGM statement for *segmname*.

DGEN250 SEGMENT 'segmname' SIZE (n) EXCEEDS MAXIMUM CONTROL INTERVAL LENGTH (m) LESS VSAM CONTROL INTERVAL INFORMATION LENGTH (p).

Explanation: The combined length of the named segment and its prefix exceeds the maximum length of a VSAM control interval less space for VSAM related information.

User Response: Shorten the segment, or split it into one or more smaller segments with a parent/child relation.

DGEN251 SEGMENT 'segmname' SIZE (n) EXCEEDS MAXIMUM ALLOWED SEGMENT SIZE LESS ROOT ANCHOR POINT AND FSE SPACE (m-p).

Explanation: The combined length of the named segment and its prefix exceeds the maximum allowed segment size, which includes space for DL/I and VSAM related information.

User Response: Shorten the segment, or split it into one or more smaller segments with a parent/child relation.

DGEN252 SEGMENT 'segmname' LENGTH + PREFIX (n) EXCEEDS RECORD LENGTH (m).

Explanation: The combined length of the named segment and its prefix exceeds the specified record length.

User Response: Either increase the specified record size (or omit the parameter and allow DL/I to calculate the size to contain the largest segment), shorten the segment, or split it into one or more smaller segments with a parent/child relation.

DGEN253 SEGMENT 'segmname' LENGTH + PREFIX + LRECL EXPANSION (n) EXCEEDS RECORD LENGTH (m).

Explanation: The combined length of the named segment and its prefix and DL/I logical record related information exceeds the specified record length. (The root segment must fit within the primary data set record. Any dependents (HIDAM only) need only fit within the overflow data set record length.)

User Response: Either increase the specified record size (or omit the parameter and allow DL/I to calculate the size to contain the largest segment), shorten the segment, or split it into one or more smaller segments with a parent/child relation.

DGEN254 SEGMENT 'segmname' LENGTH (n) EXCEEDS INPUT RECORD LENGTH (m).

Explanation: The length of the named segment exceeds the specified record length.

User Response: Either increase the specified record size (or omit the parameter and allow DL/I to calculate the size to contain the largest segment), shorten the segment, or split it into one or more smaller segments with a parent/child relation.

DGEN255 SEGMENT 'segmname' LENGTH (n) EXCEEDS OUTPUT RECORD LENGTH (m).

Explanation: The length of the named segment exceeds the specified record length.

User Response: Either increase the specified record size (or omit the parameter and allow DL/I to calculate the size to contain the largest segment), shorten the segment, or split it into one or more smaller segments with a parent/child relation.

DGEN256 SEGMENT 'segmname' SIZE (n) EXCEEDS MAXIMUM RECORD SIZE FOR dddd DEVICE (m).

Explanation: The length of the named segment exceeds the maximum allowable record length for the indicated device.

User Response: Shorten the segment, or split it into one or more smaller segments with a parent/child relation, or change the device type.

DGEN257 SEGMENT 'segmname' SIZE (n) EXCEEDS MAXIMUM CONTROL INTERVAL LENGTH (m) LESS LRECL EXPANSION (l) AND VSAM CONTROL INTERVAL INFORMATION LENGTH (c).

Explanation: The combined length of the named segment and its prefix exceeds the maximum length of a VSAM control interval less space for DL/I and VSAM related information.

User Response: Shorten the segment, or split it into one or more smaller segments with a parent/child relation.

DGEN300 DUPLICATE FIELD NAME 'fldname' IN SEGMENT 'segmname'

Explanation: *fldname* is specified as the name on two different FIELD statements.

User Response: Remove one of the field statements, or rename one of the fields.

DGEN310 MULTIPLE SEQUENCE FIELDS DEFINED FOR SEGMENT 'segmname'.

Explanation: Segments (with the exception of virtual logical child segments) cannot have more than one field identified as the primary sequence field. Fields are identified as the primary sequence field either by specification or the SEQ parameter in the sublist of the NAME operand for the field statement, or, for root segments in HD organizations, by identifying the field as the sequence field for the primary (first) ACCESS.

User Response: Either remove specification of sequence field from all but one field, or if the segment was meant to be a virtual logical child, specify POINTER=PAIRED on its SEGM statement.

DGEN311 SEQUENCE FIELD 'fldname' IN SEGMENT 'segmname' WILL NOT BE WITHIN THE MINIMUM LENGTH (n) OF THE SEGMENT.

Explanation: Sequence fields for variable length segments must lie entirely within the minimum length of the segment.

User Response: Either increase the minimum length of the segment, or move the field to a point inside the current minimum.

DGEN312 ROOT SEQUENCE FIELD 'fldname' MUST HAVE UNIQUE VALUES.

Explanation: The sequence field for a SHISAM, HISAM, HIDAM, or INDEX data base or for a HD data base with indexed primary access must have unique values.

User Response: Either use another field as the sequence field, insure unique values by appending additional information to the field, or switch to HDAM or randomized HD organization.

DGEN313 LENGTH (n) OF SEQUENCE FIELD 'fldname' IN ROOT SEGMENT EXCEEDS 236.

Explanation: The maximum length of the sequence field for a HISAM or SHISAM is 236 bytes.

User Response: Either reduce the size of the sequence field or switch to HDAM or randomized HD organization.

DGEN400 MAXIMUM NUMBER OF ACCESSES TO A SEGMENT (32) EXCEEDED FOR SEGMENT 'segmname'.

Explanation: A segment may be externally accessed (by randomizing routine or by index) by a maximum of 32 different paths.

User Response: Either reduce the number of external accesses or split the segment into two or

more smaller segments and divide the access paths between them.

DGEN410 RANDOMIZING TARGET SEGMENT 'segmname' IS NOT DEFINED.

Explanation: *segmname* was defined as the target segment for an ACCESS via a randomizing routine. No segment by that name was defined for this data base.

User Response: Either correct the name supplied on the ACCESS statement or supply a definition for segment *segmname*.

DGEN411 INDEX TARGET SEGMENT 'segmname' IS NOT DEFINED. ACCESS REFERENCE 'ref'.

Explanation: *segmname* was identified as the target of an index access referenced by name *ref*. No segment by that name was defined for this data base.

User Response: Either correct the name supplied on the ACCESS statement or supply a definition for segment *segmname*.

DGEN412 INDEX TARGET SEGMENT 'segmname' IS A LOGICAL CHILD.

Explanation: An LCHILD was supplied for segment *segmname* identifying it as the target of an index. Logical children cannot be index targets.

User Response: Either delete the index definition or move the target to a segment above the logical child in the hierarchy.

DGEN413 INDEX TARGET SEGMENT 'segmname' IS A LOGICAL CHILD. ACCESS REFERENCE 'ref'.

Explanation: The ACCESS statement with reference name *ref* was supplied defining segment *segmname* as the target of an index. Logical children cannot be index targets.

User Response: Either delete the index definition or move the target to a segment above the logical child in the hierarchy.

DGEN414 INDEX TARGET SEGMENT 'segmname' IS DEPENDENT OF A LOGICAL CHILD SEGMENT.

Explanation: An LCHILD was supplied for segment *segmname* identifying it as the target of an index. Dependents of logical children cannot be index targets.

User Response: Either delete the index definition or move the target to a segment above the logical child in the hierarchy.

DGEN415 INDEX TARGET SEGMENT '*segmname*' IS DEPENDENT OF A LOGICAL CHILD SEGMENT. ACCESS REFERENCE '*refname*'.

Explanation: The ACCESS with the identified *refname* was supplied for segment *segmname* identifying it as the target of an index. Dependents of logical children cannot be index targets.

User Response: Either delete the index definition or move the target to a segment above the logical child in the hierarchy.

DGEN416 INDEX TARGET SEGMENT '*segmname1*' IS NOT ABOVE INDEX SOURCE SEGMENT '*segmname2*' IN HIERARCHICAL SEQUENCE.

Explanation: An LCHILD statement was provided for segment *segmname1* identifying it as the target of an index and an XDFLD statement was paired with it identifying segment *segmname2* as the segment containing the sequencing field(s). *segmname2* must be a dependent of *segmname1*.

User Response: Either correct the hierarchical structure or specify a different source or target segment.

DGEN417 INDEX TARGET SEGMENT '*segmname1*' IS NOT ABOVE INDEX SEQUENCE SEGMENT '*segmname2*' IN HIERARCHICAL SEQUENCE. ACCESS REFERENCE '*ref*'.

Explanation: The ACCESS statement with reference name *ref* was supplied indicating *segmname1* as the target of an indexed access and *segmname2* as the segment containing the sequencing field(s). *segmname2* was not a dependent of *segmname1*.

User Response: Either correct the hierarchical structure or specify a different target or sequencing segment.

DGEN420 INDEX SOURCE SEGMENT '*segmname*' IS NOT DEFINED.

Explanation: An XDFLD statement identified *segmname* as the source segment for an index. No SEGM statement exists defining segment *segmname*.

User Response: Either correct the specified name on the XDFLD statement or provide a definition for segment *segmname*.

DGEN421 INDEX SEQUENCE SEGMENT '*segmname*' IS NOT DEFINED. ACCESS REFERENCE '*ref*'.

Explanation: The ACCESS statement with reference name *ref* identified *segmname* as the sequence segment for an index. No SEGM statement exists defining segment *segmname*.

User Response: Either correct the specified name on the ACCESS statement or provide a definition for segment *segmname*.

DGEN422 INDEX SOURCE SEGMENT '*segmname*' IS A LOGICAL CHILD.

Explanation: An XDFLD was supplied identifying segment *segmname* as the source segment for an index. Logical children cannot be index source segments.

User Response: Either delete the index definition or make a segment above or below the logical child the index source segment.

DGEN423 INDEX SEQUENCE SEGMENT '*segmname*' IS A LOGICAL CHILD. ACCESS REFERENCE '*ref*'.

Explanation: The ACCESS statement with reference name *ref* was supplied defining segment *segmname* as the sequence segment for an index. Logical children cannot be index sequence segments.

User Response: Either delete the index definition or move the target to a segment above the logical child in the hierarchy.

DGEN424 INDEX SOURCE SEGMENT '*segmname*' IS VARIABLE LENGTH.

Explanation: The segment *segmname* in which the field(s) to be used to form the key for an index are located is defined as variable length. (The segment may be either implicitly defined as the same as the target segment, or may be explicitly identified as other than the target on the XDFLD statement.)

User Response: Either delete the index definition, make the segment fixed length, or move the field(s) into a fixed length dependent segment.

DGEN425 INDEX SOURCE SEGMENT '*segmname*' IS VARIABLE LENGTH. ACCESS REFERENCE '*ref*'.

Explanation: The ACCESS statement with reference name *ref* defines the segment *segmname* in which the field(s) to be used to form the key for an index are located. The segment is defined as variable length. (The segment may be either implicitly defined as the same as the target segment, or may be explicitly identified as other than the target using the SEQSEG parameter.)

User Response: Either delete the index definition, make the segment fixed length, or move the field(s) into a fixed length dependent segment.

DGEN430 DUPLICATE XDFLD NAME '*fldname*' IN SEGMENT '*segmname*'.

Explanation: XDFLD names must be unique within a segment.

User Response: Either rename one of the XDFLDs or delete one of the LCHILD/XDFLD pairs.

DGEN432 XDFLD NAME '*fldname*' IS ALSO A FIELD NAME IN SEGMENT '*segmname*'.

Explanation: XDFLD names may not be the same as any field names for the target segment.

User Response: Either rename the XDFLD or rename the field.

DGEN433 ACCESS REFERENCE '*ref*' IS ALSO A FIELD NAME IN SEGMENT '*segmname*'.

Explanation: The ACCESS statement with reference parameter *ref* identifies segment *segmname* as the target segment for the index. The reference name for an index access may not be the same as any field name for fields in the target segment.

User Response: Either change the reference name or change the field name.

DGEN440 INDEX KEY FIELD FOR XDFLD '*fldname*' EXCEEDS 236 BYTES.

Explanation: The XDFLD statement with name parameter *fldname* identifies a field or set of fields to be used as the sequencing fields. The sum of the lengths of the fields to be used as the search fields (SRCH=) and any to be used as subsequence fields (SUBSEQ=) must not exceed 236.

User Response: Reduce the number or sizes of the referenced fields to a combined length of less than 236.

DGEN441 INDEX KEY FIELD FOR ACCESS '*ref*' EXCEEDS 236 BYTES.

Explanation: The sum of the lengths (+4 if values are non-unique) of the fields to be used as the sequence field(s) for indexed access defined by the ACCESS statement with reference name *ref* exceeds 236.

User Response: Reduce the number or sizes of the referenced fields to a combined length of less than 236.

DGEN450 SPECIFIED SEQUENCE FIELD NAME '*fldname*' FOR ACCESS '*ref*' NOT DEFINED IN SEGMENT '*segmname*'.

Explanation: The ACCESS statement with the REF parameter *ref* identified the sequence field (SEQFLD=) *fldname*. No definition for the field is present in the definition of the sequencing segment *segmname*.

User Response: Either supply a definition for field *fldname*, correct the field name on the SEQFLD parameter on the ACCESS statement, or change the segment specified as the sequence segment (SEQSEG=).

DGEN451 SPECIFIED SRCH FIELD NAME '*fldname1*' FOR XDFLD '*fldname2*' NOT DEFINED IN SEGMENT '*segmname*'.

Explanation: The XDFLD statement specifying name *fldname2* identified search field (SRCH=) *fldname1*. It was not defined as a field in the source segment *segmname*.

User Response: Either supply a definition for field *fldname1*, correct the field name on the SRCH parameter on the XDFLD statement, or change the segment specified as the source segment.

DGEN452 SPECIFIED SUBSEQ FIELD NAME '*fldname1*' FOR XDFLD '*fldname2*' NOT DEFINED IN SEGMENT '*segmname*'.

Explanation: The XDFLD statement specifying name *fldname2* identified subsequence field (SUBSEQ=) *fldname1*. It was not defined as a field in the source segment *segmname*.

User Response: Either supply a definition for field *fldname1*, correct the field name on the SUBSEQ parameter on the XDFLD statement, or change the segment specified as the source segment.

DGEN453 SPECIFIED SOURCE FIELD NAME '*fldname1*' FOR XDFLD '*fldname2*' NOT DEFINED IN SEGMENT '*segmname*'.

Explanation: The XDFLD statement specifying name *fldname2* identified source field (DDATA=) *fldname1*. It was not defined as a field in the source segment *segmname*.

User Response: Either supply a definition for field *fldname1*, correct the field name on the DDATA parameter on the XDFLD statement, or change the segment specified as the source segment.

DGEN454 '*fldname*' FIELD IN SEGMENT '*segmname*' IS NOT LOCATED WITHIN THE SEGMENT CONCATENATED KEY (LENGTH *n*).

Explanation: DL/I has reserved the field names starting with the characters /CK to reference portions of the concatenated key of a segment for use in qualification of non-unique index keys. The specified field *fldname* was defined for segment *segmname* with a starting location and length such that it would end outside the concatenated key for the segment to which it belongs.

User Response: Either correct the start location or length of the referenced field, or, if the field was meant to be a standard field in the segment, change its name.

DMANnnn (DATASET Macro)

DMAN100 MORE THAN ONE DATASET STATEMENT PROVIDED. THIS ONE IGNORED.

Explanation: More than one DATASET statement has been specified.

User Response: Remove the extra DATASET statements.

DMAN110 DD1 OPERAND IS OMITTED OR INVALID.

Explanation: The DD1 operand (input file name for HSAM, SHSAM; primary data set name otherwise) must be specified and must be one to seven characters in length with the first character alphabetic.

User Response: Supply a valid DD1 name.

DMAN120 DD2 OPERAND IS OMITTED OR INVALID.

Explanation: The output file name operand must be supplied for HSAM or SHSAM organization and must be one to seven characters in length with the first character alphabetic.

User Response: Either supply a valid DD2 name operand or change the access organization to something other than SHSAM or HSAM.

DMAN140 OVFLW OPERAND IS OMITTED OR INVALID.

Explanation: A VSAM data set must be defined for overflow for HISAM. The name must be one to seven characters in length with the first character alphabetic.

User Response: Either supply a valid OVFLW data set name or change the access organization to something other than HISAM.

DMAN150 DEVICE OPERAND IS OMITTED OR INVALID.

Explanation: The device type must be specified. Valid types are 2314, 3330, 3340, 3350, 3375, 3380, FBA, or, for HSAM or SHSAM, TAPE.

User Response: Supply a valid DEVICE type.

DMAN160 'TAPE' DEVICE IS INVALID FOR ACCESS=*org*.

Explanation: Tape devices are only supported for HSAM or SHSAM.

User Response: Change either the device type or the access organization.

DMAN170 ENTERED DDNAME IS A DUPLICATE - *ddname*.

Explanation: The data set or file name entered as either the DD2 or OVFLW operand was identical to the DD1 name.

User Response: Change either the DD1 name or the OVFLW or DD2 name.

DMAN200 RECORD OPERAND IS INVALID.

Explanation: The RECORD operand values must be numeric.

User Response: Either omit the record specification or supply a valid one.

DMAN201 RECORD OPERAND IS INVALID FOR *org*. IGNORED.

Explanation: The RECORD operand is only meaningful for HSAM, SHSAM, HISAM, SHISAM, or INDEX data bases.

User Response: Either omit the RECORD operand or change the access organization.

DMAN202 A MAXIMUM OF ONE OPERAND IS ALLOWED ON THE RECORD PARAMETER FOR *org*. THE REST ARE IGNORED.

Explanation: The second operand on the RECORD parameter is only meaningful for HSAM, SHSAM, or HISAM. More than two operands are not allowed for any organization.

User Response: Either correct or omit the RECORD parameter or change the access organization.

DMAN203 A MAXIMUM OF TWO OPERANDS ARE ALLOWED ON THE RECORD PARAMETER FOR *org*. THE REST ARE IGNORED.

Explanation: No more than two operands are allowed for the RECORD parameter for HSAM, SHSAM, or HISAM.

User Response: Either correct or omit the RECORD parameter.

DMAN210 BLOCK OPERAND IS INVALID.

Explanation: The BLOCK parameter must be numeric.

User Response: Either omit the BLOCK parameter or supply a valid one.

DMAN211 BLOCK OPERAND IS INVALID FOR *org*. IGNORED.

Explanation: The BLOCK parameter is not valid for HSAM or SHSAM data bases.

User Response: Either omit the BLOCK parameter or change the access organization.

DMAN212 A MAXIMUM OF ONE OPERAND IS ALLOWED ON THE BLOCK PARAMETER FOR *org*. THE REST ARE IGNORED.

Explanation: Only one operand is allowed on the BLOCK parameter for HD, HDAM, or HIDAM.

User Response: Either omit the BLOCK parameter or supply a valid one.

DMAN213 A MAXIMUM OF TWO OPERANDS ARE ALLOWED ON THE BLOCK PARAMETER FOR *org*. THE REST ARE IGNORED.

Explanation: Two operands are allowed on the BLOCK parameter for HISAM, SHISAM, or INDEX.

User Response: Either omit the BLOCK parameter or supply a valid one.

DMAN215 BLOCK SIZE FOR *org* MUST BE A MULTIPLE OF *size* AND NOT GREATER THAN *max*.. BLOCK SIZE RESET TO 2048.

Explanation: Valid block sizes (control interval sizes) for *org* data base organization are:

512- 8192 bytes in steps of 512 bytes
8192-30720 bytes in steps of 2048 bytes

User Response: Either omit the BLOCK parameter or supply a valid value.

DMAN220 LOGICAL DATASET ASSUMED.

Explanation: The 'LOGICAL' parameter was not specified on the DATASET statement, but ACCESS=LOGICAL was coded on the DBD statement. This is a warning message only.

User Response: Specify 'LOGICAL' as the first positional operand of the DATASET statement.

DMAN230 KEYWORD SPECIFICATIONS ARE INVALID FOR A LOGICAL DATASET DEFINITION. IGNORED.

Explanation: If 'LOGICAL' is specified on the DATASET statement, or if 'ACCESS=LOGICAL' is coded for the DBD statement, no additional parameters may be specified on the DATASET statement.

User Response: Either change the organization to something other than LOGICAL or remove the additional parameters from the DATASET statement. This is a warning message only.

DMAN240 DD2 PARAMETER NOT ALLOWED FOR *org*. IGNORED.

Explanation: The output file parameter, DD2, may only be specified for HSAM or SHSAM data base organizations.

User Response: Either remove specification of DD2 or change the access organization to HSAM or SHSAM.

DMAN280 SCAN OPERAND IS INVALID.

Explanation: The SCAN operand must be numeric. For FBA devices, it must be a maximum of 32767. For all others, 255.

User Response: Either omit the SCAN operand or supply a valid value.

DMAN281 SCAN OPERAND IS INVALID FOR *org*. IGNORED.

Explanation: The SCAN operand is only valid for HD, HDAM, or HIDAM.

User Response: Either omit the SCAN operand or change the access organization.

DMAN300 FIRST RECORD LENGTH MUST BE EVEN, IT WAS SET TO *nn*

Explanation: Record lengths must be a multiple of 2. This is a warning message only.

User Response: Specify an even record length.

DMAN301 SECOND RECORD LENGTH MUST BE EVEN, IT WAS SET TO *nn*

Explanation: Record lengths must be a multiple of 2. This is a warning message only.

User Response: Specify an even record length.

DMAN310 OVFLW PARAMETER NOT ALLOWED FOR *org*. IGNORED.

Explanation: The OVFLW parameter is used to specify the overflow data set for HISAM organization.

User Response: Either delete the OVFLW parameter or change the organization to HISAM.

DMAN340 ACCESS METHOD LOGICAL MUST BE CODED FOR A LOGICAL DATASET. '*org*' RESET TO 'LOGICAL'

Explanation: The ACCESS parameter on the DBD statement must specify 'LOGICAL' if the LOGICAL parameter is coded on the DATASET macro statement.

User Response: Either change the access organization to LOGICAL or drop the LOGICAL specification from the DATASET statement.

DMAN341 'LOGICAL' PARAMETER INCONSISTENT WITH OTHER PARAMETERS. IGNORED.

Explanation: ACCESS=LOGICAL was not coded on the DBD statement and parameters not allowed for a logical data base have been specified on this statement.

User Response: Either specify ACCESS=LOGICAL on the DBD statement and drop the extra DATASET parameters or drop the specification of LOGICAL from the DATASET statement.

DMAN400 ESDS RECORD LENGTH LESS THAN KSDS RECORD LENGTH. RESET TO KSDS RECORD LENGTH (*n*).

Explanation: The HISAM overflow record length may not be less than the primary record length. This is a warning message only.

User Response: Redefine the second record length to be at least equal to the first record length.

DMAN420 RECORD LENGTH (*n*) EXCEEDS TRACK LENGTH (*m*) FOR *dddd* DEVICE.

Explanation: Either the input or output record length specified exceeds the track capacity for the device on which the record is to reside.

User Response: Either change device types or reduce the record length.

DMAN450 DEVADDR OPERAND OMITTED OR INVALID.

Explanation: For HSAM or SHSAM, the logical unit addresses associated with both the input and output files must be specified using the DEVADDR operand. The format of a logical unit address is 'SYS*nnn*'.

User Response: Specify valid logical unit addresses for both input and output files - DEVADDR=(SYS*nnn*,SYS*mmm*), or change the access organization to something other than HSAM or SHSAM.

DMAN451 DEVADDR OPERAND INVALID FOR *org*. IGNORED.

Explanation: DEVADDR only has meaning for HSAM or SHSAM tape data bases.

User Response: Either omit the DEVADDR operand or change the access organization to HSAM or SHSAM.

DMAN452 A MAXIMUM OF TWO OPERANDS ARE ALLOWED ON THE DEVADDR PARAMETER FOR *org*. THE REST ARE IGNORED.

Explanation: DEVADDR is used to identify the logical units for HSAM or SHSAM tape input and output files. More than two operands is meaningless.

User Response: Omit the additional operands.

DMAN453 'DEVADDR' MAY ONLY BE SPECIFIED FOR TAPE DEVICE. IGNORED.

Explanation: DEVADDR is used to identify the logical units for HSAM or SHSAM tape input and output files.

User Response: Either omit the DEVADDR operand or specify DEVICE=TAPE.

DMAN460 VALUE FOR NUMBER OF FREE BLOCKS IS INCORRECT IN FRSPC PARAMETER.

Explanation: The first FRSPC suboperand (free block value) must be numeric, either zero or in the range two to 100.

User Response: Either specify a valid value for the suboperand or omit it.

DMAN470 FREE SPACE PERCENTAGE VALUE IS INCORRECT IN FRSPC PARAMETER.

Explanation: The second FRSPC suboperand (free space percentage) must be numeric and in the range from zero to 99.

User Response: Either specify a valid value for the suboperand or omit it.

DMAN480 A MAXIMUM OF TWO OPERANDS ARE ALLOWED ON THE FRSPC PARAMETER FOR *org*. THE REST ARE IGNORED.

Explanation: The FRSPC parameter can contain only two subparameters: the free block value and the free space percentage.

User Response: Correct the FRSPC parameter and rerun the DBD generation.

DMAN490 FRSPC PARAMETER IS NOT VALID FOR *org*. IGNORED.

Explanation: The FRSPC parameter was included in the DATASET statement during DBD generation. This parameter is valid only when used to reserve space in an HD data base, but the data base specified was other than HD, HDAM, or HIDAM. This is a warning message only.

User Response: Either change the access organization to HD, HDAM, or HIDAM or remove the FRSPC parameter.

EXTnnn (LCHILD/DBDGEN Macro)

-
- EXT001** **EXTERNAL REFERENCE TABLE FULL**
- Explanation:** More than 255 external data base and/or edit/compression routines were specified in this DBDGEN.
- User Response:** Correct the error and rerun the DBDGEN.

FINnnn (FINISH Macro)

-
- FIN095** **DBDGEN STATEMENT MISSING. GENERATED**
- Explanation:** The FINISH statement was encountered and no DBDGEN statement had been processed. The DBDGEN macro was generated from within the FINISH macro.
- User Response:** Supply a DBDGEN statement preceding the FINISH statement.
-
- FIN100** **NO SUCCESSFUL DBD IN THIS RUN**
- Explanation:** Errors were encountered during DBDGEN and as a result, no DBD was generated.
- User Response:** Correct the errors and rerun the DBD generation.

FLDnnn (FIELD Macro)

-
- FLD100** **A SEGM STATEMENT MUST PRECEDE THE FIELD STATEMENT**
- Explanation:** A FIELD statement defines a field within the segment identified by the previous SEGM statement. There was no preceding SEGM.
- User Response:** Define a SEGM statement preceding this FIELD statement or move the FIELD or SEGM statement to achieve the proper relation.
-
- FLD109** **NAME OPERAND IS INVALID**
- Explanation:** Field names must only contain alphanumeric characters. The only exceptions are names of system-related fields which begin with a slash (/CK or /SX).
- User Response:** Supply a valid field name.
-
- FLD110** **NAME OPERAND IS OMITTED**
- Explanation:** Each field requires a name, identified via the NAME parameter. Either the NAME parameter was omitted entirely or the first suboperand of the NAME parameter (field name) was omitted.
- User Response:** Supply a valid field name.
-
- FLD111** **NAME OPERAND IS NOT 1-8 CHARACTERS**
- Explanation:** Field names must be from one to eight characters in length.
- User Response:** Specify a name eight characters or less in length.
-
- FLD112** **'SEQ' OPERAND INVALID. 'SEQ' ASSUMED**
- Explanation:** The second subparameter of the NAME parameter must be either SEQ or omitted.
- User Response:** Either omit the subparameter or specify SEQ.
-
- FLD114** **'M/U' OPERAND INVALID. 'M' ASSUMED**
- Explanation:** The third suboperand of the NAME parameter is used to indicate whether the values of different occurrences of the field will be unique (U) or duplicates (M). It must be U, M, or omitted.
- User Response:** Either specify U or M or omit the suboperand.

<p>FLD115 TOO MANY OPERANDS IN NAME FIELD. REMAINDER IGNORED</p> <p>Explanation: The NAME parameter is used to specify a maximum of three values and more than three were specified.</p> <p>User Response: Remove the extra values.</p>	<p>FLD153 BYTES PARAMETER NOT VALID FOR FLOATING POINT TYPE <i>t</i></p> <p>Explanation: Floating point field types E, D, and L have implied lengths of 4, 8, and 16 respectively, which may not be overridden.</p> <p>User Response: Either remove the length specification or change the field type.</p>
<p>FLD118 SEQ FIELD MUST PRECEDE XDFLD</p> <p>Explanation: The FIELD statement defining the sequence field of this statement must appear before the LCHILD and XDFLD statement.</p> <p>User Response: Define statements in proper order.</p>	<p>FLD160 START OPERAND IS LESS THAN 1</p> <p>Explanation: If the START parameter is a numeric value, it may not be zero or negative. The first location in the segment is location one.</p> <p>User Response: Specify a value greater than zero.</p>
<p>FLD130 TYPE OPERAND IS NOT X, H, F, P, C, Z, E, D, OR L</p> <p>Explanation: The only valid values for the TYPE operand are X, H, F, P, C, Z, E, D, or L.</p> <p>User Response: Either specify a valid TYPE or omit the parameter (default type = C).</p>	<p>FLD161 START OPERAND IS GREATER THAN 32767</p> <p>Explanation: The starting position of the field relative to the beginning of the segment within which this field is defined is greater than the maximum segment length.</p> <p>User Response: Specify a value less than or equal to 32767. If the segment must be this long, break it into two segments and retrieve it by a path call.</p>
<p>FLD132 'TYPE' PARAMETER INVALID FOR '/SX' FIELD TYPE. IGNORED</p> <p>Explanation: DL/I has reserved the field names starting with /SX to indicate references to the segment RBA for qualification of non-unique index keys. The type of this field is always X.</p> <p>User Response: Either rename the field or remove the type specification.</p>	<p>FLD162 START OPERAND IS GREATER THAN 3825</p> <p>Explanation: DL/I has reserved the field names starting with /CK to indicate references to the segment concatenated key for qualification of non-unique index keys. The maximum size of a concatenated key is 3825 bytes.</p> <p>User Response: Either rename the field or specify a start location less than or equal to 3825.</p>
<p>FLD150 BYTES OPERAND IS OMITTED OR NOT IN RANGE 1-256</p> <p>Explanation: The BYTES parameter must be specified and must be a numeric value in the range from one to 256.</p> <p>User Response: Specify a valid value. If the field is too long, break it into two or more smaller fields.</p>	<p>FLD163 'START' PARAMETER INVALID FOR '/SX' FIELD TYPE. IGNORED</p> <p>Explanation: DL/I has reserved the field names starting with /SX to indicate references to the segment RBA for qualification of non-unique index keys. The start position of the field is always 1.</p> <p>User Response: Either rename the field or remove the start specification.</p>
<p>FLD151 BYTES OPERAND IS OMITTED OR NOT IN RANGE 1-3825.</p> <p>Explanation: The BYTES parameter for concatenated key (/CK) fields must be specified and must be a numeric value in the range from one to 3825.</p> <p>User Response: Specify a valid value. If the field is too long, break it into two or more smaller fields.</p>	<p>FLD164 'START' FIELD NOT PREVIOUSLY DEFINED</p> <p>Explanation: A non-numeric START parameter indicates that this field is to start at the same location as the field with the specified name. No field with the specified name was previously defined for this segment.</p> <p>User Response: Either correct the specified name or move the definition for the specified field so it precedes this one.</p>
<p>FLD152 'BYTES' PARAMETER INVALID FOR '/SX' FIELD TYPE. IGNORED</p> <p>Explanation: DL/I has reserved the field names starting with /SX to indicate references to the segment RBA for qualification of non-unique index keys. The length of this field is always 4.</p> <p>User Response: Either rename the field or remove the length specification.</p>	

FLD170 FIELD EXTENDS BEYOND DEFINED SEGMENT LENGTH

Explanation: Adding the field length to the defined starting location results in a value that exceeds the length specified on the SEGM statement via the BYTES parameter.

User Response: Either lower the starting location or the field length, increase the segment size or omit specification of the segment size and let DL/I calculate it based on field locations and sizes.

FLD180 THE LIMIT OF 1020 FIELDS PER DBD HAS BEEN EXCEEDED

Explanation: The maximum number of FIELD statements may not exceed 1020.

User Response: Either reduce the number of fields specified or split the data base into two or more smaller data bases connected using logical relations.

FLD195 MORE THAN ONE /SX FIELD SPECIFIED WITHIN SEGMENT *segmname*

Explanation: DL/I has reserved the field names starting with /SX to indicate references to the segment RBA for qualification of non-unique index keys. Only one such field may be defined per segment.

User Response: Either rename the field(s) that were not meant to be RBA references for index qualification or change the SUBSEQ specification on the XDFLD statement to reference the other /SX field and delete this one.

FLD200 THE LIMIT OF 255 FIELDS PER SEGMENT HAS BEEN EXCEEDED

Explanation: The maximum number of FIELD statements per segment may not exceed 255.

User Response: Either reduce the number of fields specified or split the segment into two or more smaller segments and retrieve them using path calls.

FLD210 FIELD STATEMENT IS NOT ALLOWED IN LOGICAL DATA BASE DEFINITIONS. IGNORED

Explanation: Logical data base segments are all based on segments in physical data bases, for which fields have already been defined. Fields in logical data bases have the same names and characteristics as the fields in the base segments and may not be redefined.

User Response: Remove this FIELD statement.

FLD225 SEQUENCE FIELD SPECIFICATION IS INVALID FOR '/CK' OR '/SX' FIELD TYPES.

Explanation: DL/I has reserved the field names starting with /CK and /SX to indicate references to the segment concatenated key and RBA for qualification of non-unique index keys. These types of fields may not be used as sequence fields for segments.

User Response: Either rename the field if it was not meant to be a reference for index qualification or remove the SEQ specification from the NAME parameter.

LCHDnnn (LCHILD Macro)

LCHD100 A SEGM STATEMENT MUST PRECEDE THE LCHILD STATEMENT

Explanation: A LCHILD statement is used to specify information about indexes or logical relations for a segment. The segment involved is implicitly the one previously defined. None was previously defined.

User Response: Define a SEGM statement preceding this LCHILD statement or move the LCHILD or SEGM statement to achieve the proper relation.

LCHD110 THE LIMIT OF 255 LCHILD STATEMENTS HAS BEEN EXCEEDED

Explanation: The maximum number of logical relation specifications, either identification of the logical parent for a logical child or the identification of a logical child for a logical parent, may not exceed 255.

User Response: Either remove some logical relations or split the data base into two or more data bases.

Note: This message can be issued during assembly of the SEGM macro. In that case, the response is the same as above.

LCHD111 PREVIOUS LCHILD STATEMENT MUST BE FOLLOWED BY AN XDFLD STATEMENT

Explanation: A previous LCHILD statement was specified providing the first part of the definition of a secondary index and the XDFLD statement providing the remainder has not yet been encountered.

User Response: Either remove the LCHILD statement or supply (or move) the XDFLD statement.

Note: This message can be issued during assembly of the SEGM and DBDGEN macros. In both cases, the response is the same as above.

LCHD112 ONLY ONE LCHILD ALLOWED. THIS ONE IGNORED

Explanation: An INDEX DBD may have only one LCHILD statement.

User Response: Remove the extra LCHILD statement.

LCHD113 LOGICAL RELATIONS SUPPORTED FOR HD ORGANIZATIONS ONLY

Explanation: Logical relations are supported only for HD, HDAM, or HIDAM organizations.

User Response: Remove the logical relation specification or change the organization to HD, HDAM, or HIDAM.

LCHD114 ACCESS SPECIFICATION LIMIT OF 255 EXCEEDED

Explanation: A maximum of 255 external access points may be defined for a single data base.

User Response: Either reduce the number of defined external access points or split the data base into two or more separate data bases connected via logical relations.

LCHD120 NAME OPERAND IS OMITTED OR INVALID

Explanation: The NAME parameter is used to specify the referenced segment and the data base in which it resides. It may have no more than two values specified, and must be coded.

The first parameter specifies the segment name. It must be present and of one to eight characters in length. The second parameter identifies the data base in which the segment resides. It is optional, but if specified, must be from one to seven characters in length.

User Response: Specify a valid NAME parameter.

LCHD130 DATABASE NAME PARAMETER IS INVALID

Explanation: The second parameter of the NAME operand, the data base name for the referenced segment, did not start with an alphabetic character.

User Response: Specify a valid data base name.

LCHD131 INDEXED DATA BASE CANNOT HAVE SAME NAME AS INDEX DATA BASE

Explanation: Data base names must be unique and the data base containing the index segments cannot be the same as the data base being indexed.

User Response: Correct the data base name operand on the NAME parameter or rename either the index or indexed data base to insure unique names.

LCHD140 POINTER/PTR OPERAND IS INVALID

Explanation: The only acceptable values for the POINTER parameter for logical child definition are NONE, SNGL, DBLE, or omitted.

User Response: Specify one of the valid values or omit the parameter (default = NONE)

LCHD141 MUST SPECIFY POINTERS FOR BI-DIRECTIONAL RELATIONSHIP. 'POINTER=SNGL' ASSUMED

Explanation: In a bi-directional logical relationship, the logical parent must have at least one pointer to the logical child. This is a warning message only.

User Response: Either omit the PAIR parameter, allowing the relationship to become

unidirectional, or specify POINTER=SNGL or DBLE.

LCHD142 POINTER AND PTR MAY NOT BOTH BE SPECIFIED

Explanation: Either POINTER= or its abbreviation PTR= may be used for this parameter.

User Response: Remove either the POINTER= or PTR= parameter.

LCHD143 'POINTER' MUST BE 'SNGL' FOR INDEX. 'SNGL' ASSUMED

Explanation: One and only one pointer exits in the index segment pointing to the index target segment. This is a warning message only.

User Response: Specify POINTER=SNGL.

LCHD145 INDICES NOT SUPPORTED FOR *org* ORGANIZATION

Explanation: Access to data bases via DL/I index data bases is supported only for HD, HDAM, or HIDAM organizations. (VSAM indexing is used for access to SHISAM or HISAM using VSAM KSDS processing. No DL/I definition other than identification of the root segment sequence field is required.)

User Response: Either remove the LCHILD (and XDFLD if necessary) statement or change the organization type to HDAM or HIDAM. (For HD, use the ACCESS statement to define indexes.)

LCHD146 USE 'ACCESS' STATEMENT TO SPECIFY INDEXES IF ACCESS=HD

Explanation: Use of LCHILD/XDFLD pairs to define indexes is restricted to HDAM and HIDAM only. The ACCESS statement is used to define all external access points to an HD data base.

User Response: Either replace the LCHILD (and XDFLD if necessary) statements with an ACCESS statement or change the organization type to HDAM or HIDAM.

LCHD150 PAIR OPERAND IS INVALID

Explanation: The segment name specified by the PAIR operand must be from one to eight characters in length.

User Response: Specify a valid paired segment name.

LCHD151 'PAIR' PARAMETER ONLY MEANINGFUL FOR LOGICAL RELATION. IGNORED

Explanation: Specification of the PAIR parameter on a LCHILD statement with POINTER=INDX or in a DBD for index organization is meaningless.

User Response: Remove the PAIR parameter or change the POINTER parameter.

LCHD160 INDEX OPERAND IS INVALID

Explanation: The INDEX operand is used to identify the field in the index target segment to be used to define the key field for the index segment. It must be one to eight characters in length. INDEX may only be specified for index organization data bases.

User Response: Either specify a valid field name, or, if the data base organization is not INDEX, remove the parameter.

LCHD165 INDEX OPERAND IS MISSING

Explanation: The field in the index target segment that is to be used to define the index key must be identified using the INDEX parameter.

User Response: Specify a valid field name using the INDEX parameter.

LCHD170 LCHILD STMT IS INVALID FOR LOGICAL SEGMENTS

Explanation: Virtual logical child segments and segments in logical data bases may not specify logical children or index relationships using the LCHILD macro.

User Response: Either delete this LCHILD statement or redefine the segment or organization type.

LCHD190 RULES OPERAND IS INVALID. 'LAST' ASSUMED

Explanation: The insertion rule specified in the RULES operand of the LCHILD statement used to define a logical relationship is invalid. Valid insertion rules are FIRST (or F), LAST (or L), and HERE (or H). The default option is LAST.

User Response: Either specify a valid value or omit the parameter.

LCHD191 'RULES' PARAMETER ONLY MEANINGFUL FOR LOGICAL RELATION. IGNORED

Explanation: Specification of the RULES parameter on a LCHILD statement with POINTER=INDX or in a DBD for index organization is meaningless.

User Response: Remove the RULES parameter or change the POINTER parameter.

LCHD200 LOGICAL CHILD CANNOT BE ITS OWN LOGICAL PARENT

Explanation: The name specified for the logical child is the same as the name of the current segment and is in the same data base.

User Response: Either change the name in the NAME parameter, the name of the current segment, or add or change the name of the logical child's data base.

SEG Mnnn (SEGM Macro)

SEGM099 DBD GENERATION TERMINATED

Explanation: All macros following this message, except DBDGEN, are not processed. DGEN090 will also print.

User Response: Correct previous error.

SEGM100 A DATASET STATEMENT MUST PRECEDE THE SEGM STATEMENT

Explanation: The DATASET statement follows the DBD statement and precedes all other statements.

User Response: Either supply the missing DATASET statement or move it to precede this SEGM.

SEGM110 THE LIMIT OF 255 SEGM STATEMENTS HAS BEEN EXCEEDED

Explanation: A maximum of 255 segment types may be defined in a single data base.

User Response: Either combine some types to reduce the total number, remove some segment types, or split the data base into two or more separate data bases connected using logical relations.

SEGM111 ONE SEGMENT ONLY FOR *org* DATABASE

Explanation: Only one segment type may be defined for SHSAM, SHISAM, or INDEX organization data bases.

User Response: If organization is SHSAM or SHISAM, change it to HSAM or HISAM respectively. If organization is INDEX, remove the extra segment definitions.

SEGM120 NAME OPERAND IS OMITTED OR INVALID

Explanation: The segment name must be specified. It must be one to eight characters in length and the first character must be alphanumeric.

User Response: Specify a valid name for the segment.

SEGM140 SEGMENT NAME IS A DUPLICATE NAME

Explanation: Segment names within a data base must be unique.

User Response: Define a unique name for the segment.

SEGM150 PHYSICAL PARENT PARAMETER OMITTED

Explanation: The first segment defined for a data base is assumed to be the root segment and has no 'parent'. All other segments are dependents of the root or its dependents and must have 'parents'.

User Response: Either supply a parent name, make this segment the root segment of this data base, or split the definition and make this segment the root in another data base.

SEGM152 ROOT SEGMENT CANNOT BE LOGICAL CHILD

Explanation: The first segment defined for a data base is assumed to be the root segment and cannot be a logical child segment. The second suboperand specified on the PARENT parameter identifies a logical parent.

User Response: Either add a dummy segment as the root segment and physical parent of this segment or remove the second suboperand.

SEGM153 'PARENT=0' ASSUMED. REMAINDER IGNORED

Explanation: More than one operand was specified in the first parent sublist parameter for the PARENT=*keyword* parameter for a root segment.

User Response: Either remove the additional operands or add a additional segment as the root segment and parent of this segment.

SEGM154 ROOT SEGMENT CANNOT HAVE A PARENT. IGNORED.

Explanation: The first SEGM statement occurring in any DBD generation defines a root segment; therefore, the PARENT=*keyword* parameter either should be omitted or the PARENT=0 parameter should be specified.

User Response: Either move the SEGM statement for the root segment to the first position, or remove the PARENT parameter from this SEGM.

SEGM155 '*pt*' POINTER SPECIFICATION IN PHYSICAL PARENT PARAMETER INVALID. 'DBLE' ASSUMED

Explanation: The second operand of the PARENT parameter must be 'SNGL', 'DBLE', or omitted. This is a warning message only.

User Response: Specify a valid value for the operand.

SEGM160 PARENT SUBPARAMETER LIST MUST BE ENCLOSED IN 2 PAIRS OF PARENTHESES. 'parm' ASSUMED.

Explanation: The proper form for the PARENT= parameter is:
 PARENT=((*ppseg*,{SNGL|DBLE}),(*lpsg*,V,*lpdbname*)).

The only required operand is the physical parent segment name *ppseg*.

User Response: Supply a PARENT parameter with the proper format.

SEGM161 TOO MANY SUBSTRING LEVELS FOR PARENT OPERAND

Explanation: No more than two levels of parenthesis are allowed in the operand of the PARENT parameter.

User Response: Remove or correct the PARENT parameter.

SEGM162 TOO MANY PARAMETERS FOR PARENT OPERAND

Explanation: No more than two operands are allowed in the first substring, no more than three in the second substring, and no more than two substrings are allowed.

User Response: Remove or correct the PARENT parameter.

SEGM180 VIRTUAL LOGICAL CHILD CANNOT ALSO BE LOGICAL CHILD

Explanation: The second sublist of the PARENT parameter may not be specified in conjunction with the SOURCE parameter.

User Response: Either drop the second sublist or omit the SOURCE parameter.

SEGM181 LOGICAL PARENT MAY NOT BE SPECIFIED FOR SEGMENTS IN A LOGICAL DATA BASE. IGNORED

Explanation: The second sublist of the PARENT parameter may not be specified for a segment in a LOGICAL data base.

User Response: Either drop the second sublist or change the DBD organization to HDAM, HIDAM, or HD.

SEGM182 'BYTES' PARAMETER INVALID FOR VIRTUAL LOGICAL CHILD. IGNORED

Explanation: The BYTES parameter may not be specified in conjunction with the SOURCE parameter.

User Response: Either drop the BYTES parameter or omit the SOURCE parameter.

SEGM183 'BYTES' MAY NOT BE SPECIFIED FOR SEGMENTS IN A LOGICAL DATA BASE. IGNORED

Explanation: The 'BYTES' parameter is not valid for segments in a LOGICAL data base. Their length is based on the length of their physical equivalents.

User Response: Either drop the BYTES parameter or change the DBD organization to something other than LOGICAL.

SEGM184 'COMPRTN' PARAMETER INVALID FOR VIRTUAL LOGICAL CHILD. IGNORED

Explanation: The COMPRTN parameter may not be specified in conjunction with the SOURCE parameter.

User Response: Either drop the COMPRTN parameter or omit the SOURCE parameter.

SEGM185 'COMPRTN' MAY NOT BE SPECIFIED FOR SEGMENTS IN A LOGICAL DATA BASE. IGNORED

Explanation: The 'COMPRTN' parameter is not valid for segments in a LOGICAL data base. Their compression routine is the one defined for their physical equivalent.

User Response: Either drop the COMPRTN parameter or change the DBD organization to HDAM, HIDAM, or HD.

SEGM186 'RULES' PARAMETER INVALID FOR VIRTUAL LOGICAL CHILD. IGNORED

Explanation: The RULES parameter may not be specified in conjunction with the SOURCE parameter. Any rules (insert, delete, replace) are specified for the related logical child on its SEGM statement or on the LCHILD statement defining it for the logical parent.

User Response: Either drop the COMPRTN parameter or omit the SOURCE parameter.

SEGM187 'RULES' MAY NOT BE SPECIFIED FOR SEGMENTS IN A LOGICAL DATA BASE. IGNORED

Explanation: The 'RULES' parameter is not valid for segments in a LOGICAL data base. Their rules are based on the rules of their physical equivalents.

User Response: Either drop the RULES parameter or change the DBD organization to HDAM, HIDAM, or HD.

SEGM190 LOGICAL PARENT PARAMETER IS INVALID

Explanation: The second sublist of the PARENT= parameter is used to identify the logical parent. The first operand (segment name) may not be omitted.

User Response: Specify the segment name for the logical parent or remove the second sublist entirely.

SEGM191 LOGICAL PARENT PARAMETER 'c' OPERAND INVALID. 'V' ASSUMED

Explanation: The second operand of the second (logical parent) sublist of the PARENT= parameter, if specified, must be 'V'.

User Response: Either specify 'V' or omit the operand.

SEGM192 LOGICAL CHILD CANNOT BE ITS OWN LOGICAL PARENT

Explanation: The names specified as the segment name and the logical parent segment name were identical and in the same data base.

User Response: Either change the name of the segment or the name specified for the logical parent segment.

SEGM193 LOGICAL PARENT MAY NOT BE SPECIFIED FOR SEGMENTS IN A *org* DATA BASE

Explanation: The second PARENT sublist is used to identify the logical parent segment and is valid only for HDAM, HIDAM, or HD access organizations.

User Response: Either remove the second sublist or change the access type to HDAM, HIDAM, or HD.

SEGM200 LOGICAL PARENT DBNAME IS INVALID - *dbdname*

Explanation: The entered logical parent data base name does not begin with an alphabetic character, contains more than seven characters, or contains a non-alphanumeric character or an '@'.

User Response: Specify a valid data base name.

SEGM210 BYTES OPERAND IS INVALID

Explanation: The value specified for the bytes parameter was not numeric.

User Response: Either specify a valid numeric value or omit the BYTES parameter.

SEGM211 VARIABLE LENGTH SEGMENTS NOT SUPPORTED FOR *org. nn* BYTES FIXED LENGTH ASSUMED

Explanation: Two values were supplied for the BYTES parameter, indicating a variable length segment. Variable length segments are only supported for HD, HDAM, or HIDAM organizations.

User Response: Either remove the extra value or specify HD, HDAM, or HIDAM organization on the DBD statement.

SEGM212 MAXIMUM OF TWO OPERANDS FOR 'BYTES' PARAMETER. REMAINDER IGNORED

Explanation: A maximum of two values may be supplied for the BYTES parameter and more than two were specified.

User Response: Remove the extra values.

SEGM213 MINIMUM SEGMENT LENGTH '*n*' LESS THAN 4. SET TO 4.

Explanation: The minimum length of the data portion of a variable length segment is 4 bytes.

User Response: Specify a valid minimum length.

SEGM214 VARIABLE LENGTH SEGMENTS NOT SUPPORTED FOR A LOGICAL CHILD SEGMENT. *nn* BYTES FIXED LENGTH ASSUMED

Explanation: Two values were supplied for the BYTES parameter, indicating a variable length segment. In addition, the segment is a real logical child. A variable length segment is not allowed in logical relationships.

User Response: Either remove the extra value for the BYTES parameter or remove the corresponding LCHILD statement.

SEGM250 POINTER AND PTR OPERANDS CANNOT BOTH BE SPECIFIED

Explanation: Either POINTER= or its abbreviation PTR= may be used for this parameter.

User Response: Remove either the POINTER= or PTR= parameter.

SEGM270 '*operand*' IS INVALID POINTER/PTR SPECIFICATION.

Explanation: Only TWIN(T), TWINBWD(TB), NOTWIN(NT), LTWIN(LT), LTWINBWD(LTB) or PAIRED are valid options for the POINTER parameter. If NOTWIN is specified first, TWIN and TWINBWD are invalid. If TWIN or TWINBWD are specified first, NOTWIN is invalid.

User Response: Correct or remove the parameter.

SEGM271 LTWIN POINTERS ONLY VALID FOR LOGICAL CHILDREN.

Explanation: Logical twin pointers, specified via the LTWIN(LT) or LTWINBWD(LTB) operands on the POINTER parameter, are valid only for logical child segments. Logical child segments are defined by specification of a second sublist identifying the logical parent on the PARENT parameter.

User Response: Either add the logical parent sublist to the PARENT parameter or remove the logical twin pointer specification.

SEGM272 POINTER OPERAND '*parm*' SPECIFIED MORE THAN ONCE IN PARAMETER LIST.

Explanation: The identified parameter was also specified in the POINTER parameter. This is a warning message only.

User Response: Remove the duplicate specification.

SEGM273 TWINBWD IMPLIES TWIN.

Explanation: Both TWIN and TWINBWD were specified in the POINTER parameter. If backward twin pointers are specified, forward pointers are assumed.

User Response: Remove the TWIN specification.

SEGM274 LTWINBWD IMPLIES LTWIN.

Explanation: Both LTWIN and LTWINBWD were specified in the POINTER parameter. If backward logical twin pointers are specified, forward logical twin pointers are assumed.

User Response: Remove the LTWIN specification.

SEGM280 RULES OPERAND IS INVALID.

Explanation: The RULES parameter may contain a maximum of two values. The first, if specified, must be three characters, each of which must be P, L, or V. The second, also optional, must be FIRST, F, LAST, L, HERE, or H.

User Response: Either delete the RULES parameter or specify valid values for it.

SEGM281 FIRST OPERAND *idr* OF RULES PARAMETER NOT MEANINGFUL FOR *org* DATABASE. IGNORED.

Explanation: The insert, delete, and replace RULES parameter applies to logical relations only. Logical relations are supported for HD,HDAM or HIDAM only.

User Response: Either delete the RULES parameter or change the data base organization to HD, HDAM, or HIDAM.

SEGM282 'V' IS ONLY VALID REPLACE RULE FOR LOGICAL CHILDREN. RULES=*idr* CHANGED TO RULES=*idv*.

Explanation: The only replace rule for logical children supported by DL/I is virtual. This is a warning message only.

User Response: Either delete the RULES parameter or change the replace rule to 'V'.

SEGM300 SOURCE SEGMENT IS INVALID FOR *org* DBDS. IGNORED

Explanation: The SOURCE parameter is used in definition of logical data bases and for virtual logical child definition in HD, HDAM, or HIDAM data organizations only.

User Response: Either remove the SOURCE parameter or change the data base organization specified on the ACCESS parameter on the DBD statement.

SEGM311 'SOURCE' MUST BE SPECIFIED FOR SEGMENTS IN A LOGICAL DATA BASE

Explanation: Each segment in a LOGICAL data base must be based on one or two segments in a physical data base. These base segments are identified via the SOURCE parameter.

User Response: Either specify the proper base segment or change the data base organization to something other than LOGICAL.

SEGM321 TOO MANY SUBSTRING LEVELS FOR SOURCE OPERAND

Explanation: No more than two levels of parenthesis are allowed in the operand of the SOURCE parameter.

User Response: Remove or correct the SOURCE parameter.

SEGM322 TOO MANY PARAMETERS FOR SOURCE OPERAND

Explanation: No more than two substrings are allowed for the SOURCE parameter in a LOGICAL data base. A maximum of three operands are allowed for each substring.

User Response: Remove or correct the SOURCE parameter.

SEGM323 THE 'SOURCE' PARAMETER MAY HAVE ONLY ONE SUB-OPERAND IN A *org* DATA BASE DEFINITION. REMAINDER IGNORED

Explanation: No more than one substring is allowed for the SOURCE parameter in a HD, HDAM, or HIDAM data base.

User Response: Remove or correct the SOURCE parameter.

SEGM430 PTR/POINTER OPERAND IS INVALID FOR *org*. IGNORED

Explanation: Pointers are only defined for HD, HDAM, or HIDAM data bases.

User Response: Either remove the POINTER parameter or change the data base organization to HD, HDAM, or HIDAM.

SEGM431 'POINTER' OPERAND INVALID FOR LOGICAL DATA BASE. IGNORED

Explanation: Pointers are only defined for HD, HDAM, or HIDAM data bases.

User Response: Either remove the POINTER parameter or change the data base organization to HD, HDAM, or HIDAM.

SEGM432 'POINTER' OPERAND INVALID. 'PAIRED' ASSUMED

Explanation: The POINTER parameter must be supplied for a virtual logical child and must be 'PAIRED'. This is a warning message only.

User Response: Specify 'POINTER=PAIRED'.

SEGM433 'POINTER' OPERAND OMITTED. 'PAIRED' ASSUMED

Explanation: The POINTER parameter must be supplied for a virtual logical child and must be 'PAIRED'. This is a warning message only.

User Response: Specify 'POINTER=PAIRED'.

SEGM440 COMPRTN OPERAND IS INVALID

Explanation: The first COMPRTN suboperand, the compression routine name, must be one to eight characters in length, with the first character alphabetic and the remainder alphanumeric.

User Response: Supply a valid compression routine name.

SEGM441 COMPRESSION NOT SUPPORTED FOR *org* DATA BASES

Explanation: Only variable length segments may be compressed. Variable length segments are supported only in HD, HDAM, or HIDAM.

User Response: Either remove the compression specification or switch the data base organization to HD, HDAM, or HIDAM.

SEGM442 COMPRESSION ONLY VALID FOR VARIABLE LENGTH SEGMENTS

Explanation: Only variable length segments may be compressed.

User Response: Either remove the compression specification or change the segment definition to variable length.

SEGM443 A MAXIMUM OF THREE OPERANDS ARE ALLOWED ON THE COMPRTN PARAMETER. THE REST ARE IGNORED

Explanation: If more than three operands are supplied for the COMPRTN parameter, the extras have no meaning and will be ignored.

User Response: Either remove the extra operands or omit the COMPRTN parameter.

SEGM462 COMPRTN 'DATA' OPERAND MISSING. 'DATA' ASSUMED

Explanation: The value specified for the second suboperand of the COMPRTN parameter was 'INIT', a valid value for the third suboperand. The second suboperand is assumed to be omitted.

User Response: Either insert a valid value (D) for the second suboperand or insert a comma in its location to indicate its omission.

SEGM463 COMPRTN '*parm*' OPERAND INVALID. 'DATA' ASSUMED

Explanation: The value specified for the second suboperand of the COMPRTN parameter was invalid. 'DATA' is assumed to be the correct value.

User Response: Either insert a valid value (D) for the second suboperand or insert a comma in its location to indicate its omission.

SEGM464 COMPRTN '*parm*' OPERAND INVALID. 'INIT' ASSUMED

Explanation: The value specified for the third suboperand of the COMPRTN parameter was invalid. 'INIT' is assumed to be the correct value.

User Response: Either insert a valid value (INIT) for the second suboperand or omit it.

SEGM470 SOURCE SEGMENT TABLE LIMIT EXCEEDED

Explanation: The maximum number of segments that may be specified via SOURCE parameters is 512.

User Response: This error can only occur as a result of specification of more than two segments in this or previous SOURCE parameters. Correct those errors.

SEGM480 SOURCE SEGMENT NAME IS OMITTED

Explanation: The SOURCE parameter consists of one or two sublists of the format (*segmname*,D,*dbdname*). The segment name was omitted in one or both of these lists.

User Response: Specify a valid segment name.

SEGM490 SOURCE SEGMENT DATA OPTION IS INVALID, OPTION=*opt.* 'DATA' ASSUMED

Explanation: The SOURCE parameter consists of one or two sublists of the format (*segmname,D,dbdname*). The second parameter in the list must be D, DATA, or omitted.

User Response: Either omit the second parameter or specify D or DATA.

SEGM500 SOURCE SEGMENT DATABASE IS INVALID - *dbdname*

Explanation: The third parameter of a SOURCE operand sublist, the data base name, must be from one to seven characters in length with the first character alphanumeric. For a LOGICAL data base, it cannot be the same as the logical DBD name.

User Response: Specify a valid name for the data base.

XDFLDnnn (XDFLD Macro)

XDFLD001 XDFLD IS VALID FOR HIDAM OR HDAM ONLY

Explanation: The XDFLD statement is used to specify information concerning secondary indexing. DL/I secondary indexing is supported for HD, HDAM, or HIDAM organizations only.

User Response: Either remove the XDFLD statement or change the organization of the data base to HD, HDAM, or HIDAM.

XDFLD002 LCHILD STATEMENT MUST PRECEDE XDFLD STATEMENT. FIRST LCHILD STATEMENT IN HIDAM DBDGEN BELONGS TO THE PRIMARY INDEX AND NO XDFLD IS ALLOWED FOR PRIMARY INDEX DEFINITION

Explanation: The XDFLD statement is used, in conjunction with a LCHILD statement, to specify information concerning secondary indexing. The related LCHILD is assumed to be the preceding one. Either there was no preceding LCHILD statement for this segment, the preceding LCHILD was for a logical relation, the preceding LCHILD has already been paired with a XDFLD statement, or, for a HIDAM root segment, the preceding LCHILD was the first for the segment and is assumed to be for the primary index. (Specification of a XDFLD statement for a primary index is not allowed.)

User Response: Either remove the XDFLD statement or add a LCHILD statement preceding it.

XDFLD003 A SEGM STATEMENT MUST PRECEDE THE XDFLD STATEMENT

Explanation: An XDFLD statement is used to specify information about secondary indexes for a segment. The segment involved is implicitly the one previously defined. None was previously defined.

User Response: Define a SEGM statement preceding the LCHILD statement or move the LCHILD/XDFLD or SEGM statement to achieve the proper relation.

XDFLD004 NAME OPERAND IS OMITTED OR INVALID

Explanation: Either the NAME operand was not specified, or more than one parameter was specified, or the specified operand contained more than eight characters.

User Response: Specify a valid name.

XDFLD005 SRCH OPERAND IS OMITTED OR INVALID

Explanation: The SRCH parameter is used to specify the field or fields to be used to create the index key. It must be specified. From one to five field names may be specified. Field names must be from one to eight characters in length.

User Response: Specify a valid SRCH parameter (see explanation).

If too many fields were specified, reduce the number of specified fields. Moving fields currently non-contiguous in the physical data base to locations and defining a third field containing them would be a method of reduction. Field level sensitivity may be used to avoid any modification of applications dependent on the current field positions.

XDFLD006 SUBSEQ OPERAND IS INVALID

Explanation: Either more than five values were specified or one or more of the field names were longer than eight characters in length.

User Response: Correct the invalid field names and/or reduce the total specified to five.

If too many fields were specified, reduce the number of specified fields. Moving fields currently non-contiguous in the physical data base to locations and defining a third field containing them would be a method of reduction. Field level sensitivity may be used to avoid any modification of applications dependent on the current field positions.

XDFLD007 DDATA OPERAND IS INVALID

Explanation: Either more than five values were specified or one or more of the field names were longer than eight characters in length.

User Response: Correct the invalid field names and/or reduce the total specified to five.

If too many fields were specified, reduce the number of specified fields. Moving fields currently non-contiguous in the physical data base to contiguous locations and defining a third field containing them would be a method of reduction. Field level sensitivity may be used to avoid any modification of applications dependent on the current field positions.

XDFLD008 EXTRTN OPERAND IS INVALID

Explanation: Either more than one parameter was specified, or the operand contains more than eight characters, or the first character is not alphabetic.

User Response: Either supply a valid routine name or omit the parameter.

XDFLD011 SEGMENT OPERAND IS INVALID

Explanation: Either more than one parameter was specified or the specified parameter contains more than eight characters.

User Response: Either specify a valid segment name or omit the parameter (default: source segment = target segment).

XDFLD012 INDEX TABLE LIMIT EXCEEDED

Explanation: A total of 4095 index related values (search fields, subsequence fields, source fields, and suppression mechanisms) are allowed. That limit has been exceeded.

User Response: Either reduce the total index parameters or split the data base into two or more parts.

XDFLD014 SPECIFIED NAME MAY NOT BE OBJECT OF PREVIOUS XDFLD SRCH, DDATA, OR SUBSEQ

Explanation: The name specified in the NAME parameter was identified in a previous XDFLD statement as a search, subsequence, or source field. An XDFLD field may not be used as any of these.

User Response: If the XDFLD was meant to be used as the search, subsequence, or source field for another index, that index will have to be redefined to avoid this. Otherwise, rename the XDFLD to avoid the conflict with the actual field name.

XDFLD021 INVALID CHARACTER 'c' IN POSITION *n* OF NULLVAL OPERAND

Explanation: The suppression value must be specified as 'ZERO', 'BLANK', or a valid self-defining term, such as:

- 0
- C'
- X'00'
- B'0'

User Response: Either supply a valid value or omit the operand.

XDFLD023 NULLVAL OPERAND NOT A VALID SELF-DEFINING TERM

Explanation: Valid self-defining terms are one of four formats:

- *n*
- C'*c*
- X'*hh*'
- B'*bbb...bb*'

where:

- *n* is a decimal number ranging from zero to 255.
- *c* is any character.
- *h* is an alphanumeric character from 0 to 9 or from A to F.
- *b* is 0 or 1.

User Response: Specify a valid self-defining term, 'ZERO', 'BLANK', or omit the parameter.

XDFLD024 NULLVAL OPERAND CONTAINS MORE THAN ONE BYTE OF INFORMATION

Explanation: The suppression value is limited to one byte of information: one character, two hex digits, eight binary digits, or a decimal number in the range zero to 255.

User Response: Either reduce the value to a single byte of information or omit it.

Chapter 3. Program Specification Block Generation (PSBGEN) Error Messages

As PSB generation is composed of VSE/Assembler language macro-instructions, errors in omission or invalid sequence of control cards, or invalid parameters on control cards will result in additional errors specified by VSE during PSB generation.

The following are the PSB generation error conditions and the messages displayed for these conditions:

PCBnnn (PCB Macro)

PCB100 THE LIMIT OF 255 PCB STATEMENTS HAS BEEN EXCEEDED

Explanation: More than 255 PCB statements have been encountered.

PCB110 TYPE OPERAND IS OMITTED OR INVALID.. 'DB' ASSUMED

Explanation: For compatibility with IMS, a TYPE=DB is required.

PCB120 DBDNAME OPERAND IS INVALID

Explanation: One of the entered values exceeds 7 characters. A dbdname must begin with an alphabetic character; the remaining characters may be alphanumeric. An at-sign (@) must not be used in the dbdname.

PCB170 DBDNAME OPERAND IS OMITTED

Explanation: The DBDNAME=*name* parameter must be included because it specifies the name of the physical or logical DBD to be used as the primary source of data base segments for this logical data structure.

PCB180 PROCOPT OPERAND IS INVALID

Explanation: Valid parameters are:

- G Get function
- I Insert function
- R Replace function
- D Delete function
- A All, includes the above four functions
- P Required if command code D is used on get type functions or insert functions
- E Exclusive use of the data base
- O Inhibits locking by program isolation during retrievals of the same segment types in a data base
- L Load function for data base loading (except HIDAM)

- LS Load for HIDAM, and for HSAM and simple HSAM when a sequence field is defined in the root segment
- N Returns status code 'GG' instead of abnormal termination if invalid pointers are detected.
- T Same as 'N', except that an automatic retry of the call is performed if invalid pointers are detected. Status code 'GG' will be returned if the retry fails.

PCB190 KEYLEN OPERAND IS OMITTED OR INVALID

Explanation: The KEYLEN operand has been omitted, is not a numeric value, or is greater than 32767.

PCB210 POS OPERAND IS INVALID

Explanation: POS=, if included in the PCB statement, must be either:

POS=SINGLE(or S), or
POS=MULTIPLE(or M)

If POS= is omitted from the PCB statement, single positioning is assumed.

PCB230 INVALID SYMBOL IN DBDNAME OPERAND

Explanation: The entered data base name must be from one to seven characters in length, and must begin with an alphabetic character. The remaining characters can be alphanumeric.

PCB260 PROCSEQ OPERAND IS INVALID

Explanation: PROCSEQ is not valid for loading a data base; that is, when PROCOPT=L or PROCOPT=LS has been specified. PROCSEQ=*index-dbdname* is valid only if a secondary index data base exists for processing the data base identified in the DBDNAME= operand.

PCB270 PROCOPT SPECIFICATION IS INVALID

Explanation: PROCOPT=L or PROCOPT=LS was specified along with a PROCSEQ specification.

PGENnnn (PSBGEN Macro)

PGEN100 PSBNAME OPERAND IS OMITTED OR INVALID

Explanation: The PSBNAME=*psbname* operand must be included in the PSBGEN statement and *psbname* must be no longer than seven characters. At-signs (@) *must not be used* in the *psbname*.

PGEN110 LANG OPERAND IS OMITTED OR INVALID

Explanation: The LANG= operand must be included in the PSBGEN statement. Valid parameters are:

- ASSEM
- COBOL
- PL/I (or PL/1, PLI, PL1)
- RPG

with no trailing blanks. For example:

PSBGEN LANG=COBOL,PSBNAME=psbname

PGEN120 PSBGEN TERMINATED; ERRORS IN PSB

Explanation: Self-explanatory.

PGEN130 PCB *pcbname* HAS NO SENSITIVE SEGMENTS

Explanation: This is a warning message only. SENSEG statement(s) must immediately follow the PCB statement with which they are associated.

PGEN140 SYSTEM ERROR; PSBGEN TERMINATED

Explanation: An error occurred during source segment table generation (should not occur). Save documentation and contact the IBM Support Center.

SEGnnn (SENSEG Macro)

SEG100 THE LIMIT OF 500 SENSEG STATEMENTS PER PSB HAS BEEN EXCEEDED

Explanation: More than 500 SENSEG statements have been encountered in total for all PCBs in a PSBGEN.

SEG101 THE LIMIT OF 255 SENSEG STATEMENTS PER PCB HAS BEEN EXCEEDED

Explanation: More than 255 SENSEG statements have been encountered for a PCB.

SEG110 A PCB STATEMENT MUST PRECEDE THE SENSEG STATEMENT

Explanation: SENSEG statement(s) must immediately follow the PCB statement with which they are associated.

SEG150 DUPLICATE SENSEG NAME IN PCB

Explanation: Self-explanatory.

SEG160 SENSEG PARENT NOT DEFINED

Explanation: The named parent has not been previously defined as a sensitive segment.

SEG180 NAME OPERAND IS OMITTED OR INVALID

Explanation: The NAME= operand must contain a name that is from one to eight characters and identifies a segment defined in a SEGM statement during DBD generation.

SEG190 PARENT OPERAND IS OMITTED OR INVALID

Explanation: The PARENT= operand, if included in the SENSEG statement, must contain a name that is from one to eight alphanumeric characters, or 0 (zero). If the SENSEG statement defines a root segment, the PARENT= operand can be omitted, or PARENT=0 specified.

SEG230 PROCOPT OPERAND IS INVALID

Explanation: The PROCOPT= operand in the SENSEG statement has the same meaning as in the PCB statement. If PROCOPT=*p* is included in the SENSEG statement, it overrides the processing option specified in the PCB statement. Valid specifications are:

p	Explanation
G	Get function.
I	Insert function.
R	Replace function.
D	Delete function.
A	All, include the above four functions.
P	Required if command code D is used on get type functions or insert functions.
E	Exclusive use of the data base.

If PROCOPT=L or PROCOPT=LS is specified in a preceding PCB statement, the SENSEG PROCOPT operand *must be omitted*.

SFLDnnn (SENFLD Macro)

SFLD100 PCB AND SENSEG STATEMENTS MUST PRECEDE SENFLD STATEMENT

Explanation: One PCB statement and the applicable SENSEG statements must precede the SENFLD statement in a PSB generation.

SFLD110 SENSEG STATEMENT MUST PRECEDE SENFLD STATEMENT

Explanation: Following the PCB statement, the applicable SENSEG statements must precede the SENFLD statement in a PSB generation.

SFLD120 THE LIMIT OF 4095 FIELDS WITHIN A PSB HAS BEEN EXCEEDED

Explanation: The maximum number of fields that can be specified by the SENFLD, VIRFLD, and SENSEG statements is 4095.

SFLD130 THE LIMIT OF 255 FIELDS WITHIN A SEGMENT HAS BEEN EXCEEDED

Explanation: The maximum number of SENFLD and VIRFLD statements for a sensitive segment may not exceed 255.

SFLD140 NAME PARAMETER IS OMITTED OR INVALID

Explanation: The NAME=*fldname* parameter must be specified. *fldname* must begin with an alphabetic character and be 1 to 8 characters long. The remaining seven characters may be alphanumeric.

SFLD145 NAME WAS PREVIOUSLY DEFINED WITHIN SENSEG

Explanation: The name specified in the NAME= parameter was specified on a previous SENSEG statement within the same SENSEG. Duplicate SENFLD names within the same SENSEG are not permitted.

SFLD150 START PARAMETER IS NOT IN RANGE 1 TO 32767

Explanation: The START=*position* parameter specifies the starting position of this field in terms of bytes relative to the beginning of the user's view of the segment within which this field is defined. It must be numeric and in the range of 1 to 32767.

SFLD155 'START' FIELD NOT PREVIOUSLY DEFINED

Explanation: A non-numeric START=*name* parameter indicates that this field is to start at the same location as the field with the specified name. No field with the specified name was previously defined for this segment.

SFLD160 TYPE PARAMETER IS NOT X, H, F, C, Z, P, E, D OR L

Explanation: The TYPE=*t* parameter must have one of the following designated alphabetic characters substituted for *t*.

t	Explanation
X	Hexadecimal (binary)
H	Half word binary
F	Full word binary
P	Packed decimal
C	Character data
Z	Numeric character data
E	Floating point (short)
D	Floating point (long)
L	Floating point (extended)

SFLD180 BYTES PARAMETER NOT VALID FOR FLOATING POINT TYPE *t*

Explanation: Do not use the BYTES=*n* parameter with the floating point TYPE=E, TYPE=D, or TYPE=L parameter. Floating point has implicit lengths (E=4, D=8, L=16) that may not be overridden.

SFLD190 BYTES PARAMETER IS NOT IN RANGE 1 TO 256

Explanation: The length of the field must be within the range of 1 to 256 bytes. It must be numeric.

SFLD200 FIELD EXTENDS BEYOND MAXIMUM SEGMENT LENGTH (32767)

Explanation: The end of the field, calculated by adding the length to the starting position (indicated by the START=*position* parameter) is located beyond the maximum segment length (32767 bytes).

SFLD210 RTNAME PARAMETER IS INVALID

Explanation: In the RTNAME=*program* parameter, *program* is the phase name of a user-written field exit module in the library that is given control whenever this field is retrieved or stored. It must be a 1 to 8 character name; the first character must be alphabetic, and the remaining characters may be alphanumeric.

SFLD230 'REPLACE' AND 'REPL' NOT BOTH ALLOWED ON ONE SENFLD

Explanation: Do not specify both REPL=... and REPLACE=... in the same SENFLD statement.

SFLD240 REPLACE PARAMETER IS NOT 'YES' OR 'NO' - 'NO' ASSUMED

Explanation: In the SENFLD statement, a parameter other than REPLACE=YES, REPLACE=NO, or acceptable abbreviation was used. REPLACE=NO was assumed.

VFLDnnn (VIRFLD Macro)

VFLD100 PCB AND SENSEG STATEMENTS MUST PRECEDE VIRFLD STATEMENT

Explanation: Following the PCB statement, the applicable SENSEG statements must precede the VIRFLD statement in a PSB generation.

VFLD110 SENSEG STATEMENT MUST PRECEDE VIRFLD STATEMENT

Explanation: Following the PCB statement, the applicable SENSEG statements must precede the VIRFLD statement in a PSB generation.

VFLD120 THE LIMIT OF 4095 FIELDS WITHIN A PSB HAS BEEN EXCEEDED

Explanation: The maximum number of fields that can be specified by the SENSEG, VIRFLD, and SENSEG statements is 4095.

VFLD130 THE LIMIT OF 255 FIELDS WITHIN A SEGMENT HAS BEEN EXCEEDED

Explanation: The maximum number of SENFLD and VIRFLD statements for a sensitive segment may not exceed 255.

VFLD140 NAME PARAMETER IS OMITTED OR INVALID

Explanation: The NAME=*fldname* parameter must be specified, and it must begin with an alphabetic character and be 1 to 8 characters long. The remaining seven characters may be alphanumeric.

VFLD150 START PARAMETER IS NOT IN RANGE 1 TO 32767

Explanation: The START=*position* parameter specifies the starting position of this field in terms of bytes relative to the beginning of the user's view of the segment within which this field is defined. It must be in the range of 1 to 32767.

VFLD155 'START' FIELD NOT PREVIOUSLY DEFINED

Explanation: A non-numeric START=*name* parameter indicates that this field is to start at the same location as the field with the specified name. No field with the specified name was previously defined for this segment.

VFLD160 TYPE PARAMETER IS NOT X, H, F, C, Z, P, E, D OR L

Explanation: The TYPE=*t* parameter must have one of the following designated alphabetic characters substituted for *t*.

t	Explanation
X	Hexadecimal (binary)

H	Half word binary
F	Full word binary
P	Packed decimal
C	Character data
Z	Numeric character data
E	Floating point (short)
D	Floating point (long)
L	Floating point (extended)

VFLD170 'BYTES' PARAMETER REQUIRED FOR TYPES X, C, Z OR P

Explanation: The BYTES=*n* parameter must be used when the TYPE=X, TYPE=C, TYPE=Z or TYPE=P parameter is specified.

VFLD180 BYTES PARAMETER NOT VALID FOR FLOATING POINT TYPE 'f'

Explanation: Do not use the BYTES=*n* parameter with the floating point TYPE=E, TYPE=D, or TYPE=L parameter. Floating point has implicit lengths (E=4, D=8, L=16) that may not be overridden.

VFLD190 BYTES PARAMETER NOT IN RANGE 1 TO 256

Explanation: The length of the field must be within the range of 1 to 256 bytes. It must be numeric.

VFLD200 FIELD EXTENDS BEYOND MAXIMUM SEGMENT LENGTH (32767)

Explanation: The end of the field, calculated by adding the length to the starting position (indicated by the START=*position* parameter) is located beyond the maximum segment length (32767 bytes).

VFLD210 RTNAME PARAMETER IS INVALID

Explanation: In the RTNAME=*program* parameter, *program* is the phase name of a user-written field exit module in the library that is given control whenever this field is retrieved or stored. It must be a 1 to 8 character name; the first character must be alphabetic, and the remaining characters may be alphanumeric.

VFLD250 'TYPE' PARAMETER MUST BE SPECIFIED WITH 'VALUE' PARAMETER

Explanation: When the VALUE=*value* parameter is used to specify an initial value, the TYPE=*t* parameter must also be used to specify the type of field.

VFLD260 'VALUE' MUST BE NUMERIC FOR 'f' TYPE FIELDS

Explanation: The operand of a VALUE parameter must be numeric for type H, F, P, or Z fields.

VFLD270 'VALUE' WILL BE PADDED

Explanation: When the number of characters specified in the `VALUE=value` parameter is not sufficient to fill the length specified by the `BYTES=n` parameter, zeros are added in the high-order positions for `TYPE=P`, `TYPE=X` and `TYPE=Z` parameters and blanks are added to the low-order positions for the `TYPE=C` parameter.

VFLD280 'VALUE' LENGTH EXCEEDS 'BYTES' SPECIFICATION

Explanation: The number of characters specified in the `VALUE=value` parameter is too large to fit in the length specified by the `BYTES=n` parameter.

Chapter 4. Online Nucleus Generation Error Messages

This chapter lists the online nucleus generation error conditions and the messages displayed for these conditions.

Online nucleus generation is run as a standard VSE job. Therefore, in addition to the messages listed in this chapter, other messages are issued by VSE if error conditions are found during the processing of VSE job control statements.

DLZ1nnn (DLZACT Macro)

DLZ1000 AN ENTRY CARD WITH THE LABEL DLZNUC MUST BE SUPPLIED WHEN CATALOGING THIS MODULE

Explanation: An ENTRY DLZNUC card is required at LNKEDT time.

DLZ1001 ONLY ONE TYPE=INITIAL STATEMENT ALLOWED

Explanation: More than one DLZACT TYPE=INITIAL macro instruction was coded in the source deck used to assemble the DL/I ACT.

DLZ1002 SUFFIX OPERAND TOO LONG, BLANK SUFFIX ASSUMED

Explanation: A suffix was specified in the ACT TYPE=INITIAL macro instruction that exceeded two alphanumeric characters.

DLZ1003 TYPE=INITIAL STATEMENT MISSING, BLANK SUFFIX ASSUMED

Explanation: A DLZACT TYPE=INITIAL macro instruction was not coded as the first statement in the source deck used to assemble the DL/I ACT.

DLZ1004 ONLY ONE TYPE=CONFIG STATEMENT IS ALLOWED, STATEMENT IGNORED

Explanation: There may be only one DLZACT TYPE=CONFIG statement for each DL/I ACT generation.

DLZ1005 MAXTASK OPERAND MISSING, 10 DEFAULTED FOR MAXTASK VALUE

Explanation: A value was not specified in the MAXTASK operand of the DLZACT TYPE=CONFIG macro instruction; therefore, 10 is assumed.

DLZ1006 MAXTASK OPERAND INVALID, DEFAULTED TO 10

Explanation: The value specified in the MAXTASK operand of the DLZACT TYPE=CONFIG macro instruction was not a numeric value from 1 to 255, therefore, 10 is assumed.

DLZ1007 CMAXTSK NON-NUMERIC OR GREATER THAN MAXTASK, SET TO MAXTASK

Explanation: The value specified in the CMAXTSK operand of the DLZACT TYPE=CONFIG macro instruction was not a numeric value from 1 to 255, or else it exceeded the value specified in the MAXTASK operand; therefore the value specified for MAXTASK is assumed.

DLZ1008 BFRPOOL OPERAND INVALID, SET TO ZERO

Explanation: The value specified in the BFRPOOL operand of the DLZACT TYPE=CONFIG macro instruction was not a numeric value from 0 to 255; therefore, a 0 value is used.

DLZ1009 SLC NAME - *phname* - INVALID, HAS BEEN IGNORED

Explanation: The phase name of the storage layout control table specified in the SLC operand of the DLZACT TYPE=CONFIG macro instruction is invalid.

DLZ1010 *pgmname* PREVIOUSLY DEFINED, DUPLICATE IGNORED

Explanation: The application program name specified in PGMNAME of the DLZACT TYPE=PROGRAM macro instruction was previously defined; the duplicate is ignored.

DLZ1011 TYPE=CONFIG STATEMENT MUST PRECEDE THE FIRST TYPE=PROGRAM STATEMENT, STATEMENT IGNORED

Explanation: There must be one DLZACT TYPE=CONFIG statement for each DL/I ACT generation. It must follow the DLZACT TYPE=INITIAL macro instruction.

DLZ1012 PHASE NAME IN THE SLC OPERAND TRUNCATED TO 8 CHARACTERS

Explanation: The phase name of the storage layout control specified in the SLC operand of the DLZACT TYPE=CONFIG macro instruction exceeded 8 alphanumeric characters and was therefore truncated.

DLZ1013 PGMNAME PARAMETER MISSING, ENTRY IGNORED

Explanation: An application program name was not specified in the PGMNAME operand of the DLZACT TYPE=PROGRAM macro instruction. The instruction is ignored.

DLZ1014 ONLY ONE PGMNAME ALLOWED, ENTRY IGNORED

Explanation: More than one application program name was specified in the PGMNAME operand of the DLZACT TYPE=PROGRAM macro instruction. The instruction is ignored.

DLZ1015 PROGRAM NAME LONGER THAN 8 CHARACTERS IN TYPE=PROGRAM STATEMENT, STATEMENT IGNORED

Explanation: The application program name specified in the PGMNAME operand of the DLZACT TYPE=PROGRAM macro instruction exceeds 8 alphanumeric characters. The instruction is ignored.

DLZ1016	PGM COUNT EXCEEDS 4095, ALL FOLLOWING ENTRIES IGNORED Explanation: The maximum of 4095 DLZACT TYPE=PROGRAM statements was exceeded in a DL/I ACT generation.	DLZ1024	PSBCOUNT EXCEEDS 4095, ALL FOLLOWING PSB'S IGNORED Explanation: More than 4095 PSB names were defined in DLZACT TYPE=PROGRAM statements in a DL/I ACT generation.
DLZ1017	PSBNAME PARAMETER MISSING, ENTRY IGNORED Explanation: A PSB name was not specified in the PSBNAME operand of the DLZACT TYPE=PROGRAM macro instruction. The instruction is ignored.	DLZ1025	INVALID SPECIFICATION IN TYPE FIELD Explanation: The valid type fields for the DLZACT macro instruction are TYPE=INITIAL, TYPE=CONFIG, TYPE=PROGRAM, TYPE=RPSB, TYPE=BUFFER, and TYPE=FINAL. This entry is ignored.
DLZ1018	TYPE=PROGRAM STATEMENT OUT OF SEQUENCE, IGNORED Explanation: The DLZACT TYPE=PROGRAM statement was encountered after a TYPE=RPSB or TYPE=BUFFER statement. The out of sequence TYPE=PROGRAM statement is ignored.	DLZ1026	MORE THAN 255 HDBFR STATEMENTS WERE SPECIFIED, ENTRY IGNORED Explanation: More than 255 DL/I subpools were defined in DLZACT TYPE=BUFFER statements in a DL/I ACT generation. This entry is ignored.
DLZ1019	PSBNAME <i>name</i> EXCEEDS 7 CHARACTERS, STATEMENT IGNORED Explanation: The PSB name specified in the PSBNAME operand of the DLZACT TYPE=PROGRAM statement or the LNAME operand of the TYPE=RPSB statement exceeded 7 alphanumeric characters. The statement is ignored.	DLZ1027	NUMBER OF HDBFR STATEMENTS EXCEEDS BFRPOOL SIZE ON TYPE=CONFIG STATEMENT, BFRPOOL WILL BE ADJUSTED Explanation: The number of buffer subpools specified in the BFRPOOL operand of DLZACT TYPE=CONFIG macro instruction is smaller than the number specified in the HDBFR operand of the DLZACT TYPE=BUFFER macro instructions. At execution, the buffer pool will be built according to the specifications in the HDBFR operand.
DLZ1020	NUCLEUS GENERATION ERROR, GENERATION TERMINATED Explanation: A severe error condition occurred that caused nucleus generation processing to terminate. The error condition is given in the preceding message(s).	DLZ1028	INVALID BUFFER SPECIFICATION IN HDBFR OPERAND, IT WAS SET TO <i>nn</i> Explanation: The number of buffers specified in the HDBFR operand of the DLZACT TYPE=BUFFER macro instructions was not a numeric value from 2 to 32. It is set to the number indicated in the message text.
DLZ1021	NO VALID ACT ENTRY, GENERATION TERMINATED Explanation: There were no valid DLZACT TYPE=PROGRAM macro instructions in this online nucleus generation. There must be at least one.	DLZ1029	DBDNAME PREVIOUSLY DEFINED - <i>dbdname</i> - DUPLICATE IGNORED Explanation: The DBD name identified in the message text was previously specified in another HDBFR operand of a DLZACT TYPE=BUFFER macro instruction or appears twice in the same statement. The duplicate DBD name is ignored.
DLZ1022	ONE OR MORE NON-TERMINATION ERRORS DETECTED DURING GENERATION Explanation: Nucleus generation completed successfully, although one or more error conditions were detected. The error(s) that occurred are identified in the preceding message(s).	DLZ1030	INVALID DBDNAME IN HDBFR OPERAND, DBD <i>dbdname</i> IGNORED Explanation: The DBD name specified in the HDBFR operand of the DLZACT TYPE=BUFFER macro instruction and identified in this message is invalid. It must be 1 to 7 alphanumeric characters long.
DLZ1023	PASSWORD TRUNCATED TO 8 CHARACTERS Explanation: The password specified in the PASS operand of the DLZACT TYPE=CONFIG macro instruction exceeded 8 alphanumeric characters. It was truncated to 8 characters.		

DLZ1031 NO HDBFR OR HSBFR ENTRY IN TYPE=BUFFER STATEMENT, STATEMENT IGNORED

Explanation: The DLZACT TYPE=BUFFER macro instruction was ignored because no HDBFR or HSBFR operand was specified.

DLZ1032 INVALID FORMAT OF HSBFR - OPERAND IGNORED

Explanation: The HSBFR operand of the DLZACT TYPE=BUFFER macro instruction has either too many or too few subparameters. The operand is ignored.

DLZ1033 INDEX BUFFER SPECIFICATION INVALID - SET TO *number*

Explanation: The number of index buffers specified in the HSBFR operand of the DLZACT TYPE=BUFFER macro instruction was not a numeric value of one or more. It is set to the number indicated in the message text.

DLZ1034 DATA BUFFER SPECIFICATION INVALID - SET TO *number*

Explanation: The number of data buffers specified in the HSBFR operand of the DLZACT TYPE=BUFFER macro instruction was not a numeric value of two or more. It is set to the number indicated in the message text.

DLZ1035 ESDS BUFFER SPECIFICATION INVALID - SET TO *number*

Explanation: The number of ESDS (entry sequence data set) data buffers specified in the HSBFR operand of the DLZACT TYPE=BUFFER macro instruction was not a numeric value of two or more. It is set to the number indicated in the message text.

DLZ1036 DBDNAME MISSING, INVALID OR DUPLICATE - HSBFR IGNORED

Explanation: The HSBFR operand of the DLZACT TYPE=BUFFER macro instruction is ignored because the DBD name was one of the following:

- Not specified.
- Invalid (it must be 1 to 7 alphanumeric characters).
- Previously specified in another statement.
- Specified twice in the same statement.

DLZ1037 PI=YES ASSUMED IN THE TYPE=CONFIG STATEMENT

Explanation: The default value of PI=YES was assumed for the PI parameter in the DLZACT TYPE=CONFIG statement.

DLZ1038 TYPE=BUFFER STATEMENT MUST BE PRECEDED BY TYPE=PROGRAM STATEMENT, STATEMENT IGNORED

Explanation: A TYPE=BUFFER statement was encountered before any TYPE=PROGRAM statements. The TYPE=BUFFER statement is ignored.

DLZ1039 IN TYPE=CONFIG STATEMENT, VALUE *value* INVALID FOR REMOTE=, REMOTE=YES ASSUMED

Explanation: If specified, the REMOTE parameter of the DLZACT TYPE=CONFIG statement must be REMOTE=YES or REMOTE=NO. If specified incorrectly, REMOTE=YES is assumed.

DLZ1040 INVALID 'CONT' PARAMETER, NO CONTINUATION ASSUMED

Explanation: Invalid CONT parameter specified in DLZACT TYPE=PROGRAM statement.

DLZ1041 TYPE=RPSB STATEMENT OUT OF ORDER, STATEMENT IGNORED

Explanation: All TYPE=RPSB statements must immediately follow the last TYPE=PROGRAM statement. The TYPE=RPSB statement is ignored.

DLZ1042 PSB *psbname* IN TYPE=RPSB STATEMENT NOT PREVIOUSLY DEFINED IN TYPE=PROGRAM STATEMENT, STATEMENT IGNORED

Explanation: Only a PSB previously defined in a TYPE=PROGRAM statement can be defined as a remote PSB in a TYPE=RPSB statement. The TYPE=RPSB statement is ignored.

DLZ1043 REMOTE PSB *psbname* PREVIOUSLY DEFINED, DUPLICATE IGNORED

Explanation: The *psbname* specified in the PSB parameter of the DLZACT TYPE=RPSB statement was previously defined. The duplicate *psbname* is ignored.

DLZ1044 SYSID PARAMETER IN TYPE=RPSB STATEMENT LONGER THAN 4 CHARACTERS, STATEMENT IGNORED

Explanation: The SYSID parameter contained more than 4 characters. The TYPE=RPSB statement is ignored.

DLZ1045 TYPE=RPSB STATEMENT MUST HAVE SYSID PARAMETER, STATEMENT IGNORED

Explanation: The SYSID parameter must be specified on a TYPE=RPSB statement. The statement is ignored.

DLZ1046 RNAME *name* IN TYPE=RPSB STATEMENT IS LONGER THAN 8 CHARACTERS, STATEMENT IGNORED

Explanation: The RNAME parameter contained more than 8 characters. The TYPE=RPSB statement is ignored.

DLZ1047 TYPE=RPSB STATEMENT MUST HAVE PSB PARAMETER, STATEMENT IGNORED

Explanation: A PSB name was not specified in the PSB parameter of the DLZACT TYPE=RPSB statement. The statement is ignored.

DLZ1048 IN TYPE=RPSB STATEMENT, VALUE *value* INVALID FOR LANG=, PARAMETER IGNORED

Explanation: The LANG parameter on the TYPE=RPSB statement must have a value of PL/I, PL/1, PLI, or PL1. Other values are ignored.

DLZ1049 LOCAL PART OF AN RPSB IS NOT A LOCAL PSB, STATEMENT IGNORED

Explanation: The LNAME parameter must specify a local PSB. The PSB name in the LNAME parameter was omitted or was previously defined as an RPSB. The statement is ignored.

DLZ1050 REMOTE PSB *xxxxxx* PREVIOUSLY DEFINED AS LOCAL PSB IN TYPE=RPSB STATEMENT, STATEMENT IGNORED

Explanation: The PSB parameter specified was previously specified in an LNAME parameter of a TYPE=RPSB statement. The same PSB name cannot be used as the name of an RPSB. The statement is ignored.

DLZ1051 ONE OR MORE INVALID PARAMETERS SPECIFIED ON TYPE=xxxxxxx STATEMENT, PARAMETER(S) IGNORED

Explanation: One or more of the parameters specified on the TYPE=xxxxxxx statement should be specified on a different TYPE= statement. Refer to *DL/I DOS/VS Resource Definition and Utilities* for more details. The invalid parameters are ignored.

DLZ1052 WHEN LNAME IS USED ON TYPE=RPSB STATEMENT, PSBNAME MAY NOT BE THE SAME AS RNAME, STATEMENT IGNORED

Explanation: The second value specified for the LNAME parameter on the TYPE=RPSB statement must be either FIRST or SECOND. SECOND was assumed indicating that the local PCB addresses are to follow the remote PCB address when concatenated to form a single PCB address list.

DLZ1053 PI=NO SPECIFIED ON TYPE=CONFIG STATEMENT WHEN AN RPSB WITH A LOCAL COMPONENT HAS BEEN DEFINED, PI=YES ASSUMED

Explanation: PI=NO was specified on TYPE=CONFIG statement indicating program isolation was not to be used. PI=YES is required for Extended Remote PSB defined in this TYPE=RPSB statement; therefore, PI=YES is assumed.

DLZ1054 RNAME MUST BE SPECIFIED WHEN LOCAL COMPONENT USED ON TYPE=RPSB STATEMENT, STATEMENT IGNORED

Explanation: The name of an RPSB with a local PSB component is not allowed to be the same as the name of the remote PSB component. The PSB=*psbname* or the RNAME=*remotename* parameter of the TYPE=RPSB statement must be changed.

DLZ1055 REMOTE PSB WITH LOCAL COMPONENT NOT ALLOWED IN DFHMIR TYPE=PROGRAM STATEMENT, ENTRY IGNORED IN DFHMIR TYPE=PROGRAM STATEMENT

Explanation: Program DFHMIR, the CICS/VS intercommunication mirror program, is not allowed to schedule an RPSB with a local component. The name of this RPSB was specified in the TYPE=PROGRAM statement for DFHMIR. Therefore, the RPSB entry in the DFHMIR TYPE=PROGRAM statement will be ignored. Definition of this PSB as a remote PSB is allowed.

DLZ1056 ONLY ONE ENTRY PER PARAMETER ALLOWED IN TYPE=RPSB STATEMENT, STATEMENT IGNORED

Explanation: One or more parameters in a TYPE=RPSB statement contains a multiple number of entries. TYPE=RPSB statement parameters may consist of a single entry only. The TYPE=RPSB statement will be ignored.

DLZ1057 PSB COUNT EXCEEDS 4095, STATEMENT IGNORED

Explanation: The local component of the TYPE=RPSB statement cannot be defined since the number of PSBs in this nucleus generation exceeds the maximum allowed. The TYPE=RPSB statement will be ignored.

DLZ1058 PCB(S) FROM *xxxxxxx* PRECEDE PCB(S) FROM *xxxxxxx* IN RPSB *xxxxxxx*

Explanation: This MNOTE is for information only. The PCB(s) in the local component PSB will be concatenated and presented to the application program in the order specified in the MNOTE.

DLZ3nnn (CALLDLI Macro)

DLZ3001 NO PARAMETER CODED

Explanation: No parameters were coded in a CALLDLI macro instruction.

DLZ3002 ASMTDLI OR CBLTDLI MUST BE FIRST PARM

Explanation: The first parameter in a CALLDLI macro instruction must be ASMTDLI or CBLTDLI (for Assembler language programs).

DLZ3003 SUB-PARAMETER LIST EXCEEDS 19 ALL ADDITIONAL PARAMETERS IGNORED

Explanation: The subparameter list may not exceed 19 parameters.

DLZ3005 MF OPERAND REQUIRES 2 SUB-PARAMETERS

Explanation: The MF operand in a CALLDLI macro instruction requires two subparameters; E and the address of the parameter list or the register containing the address of the parameter list.

DLZ3006 FIRST MF PARAMETER MUST BE E

Explanation: The first parameter of a MF operand in a CALLDLI macro instruction must be E.

DLZ3007 REGISTER NOTATION REQUIRES 2 PARENTHESES

Explanation: The register notation used as the second parameter of a MF operand in a CALLDLI macro instruction must be enclosed in parentheses.

Chapter 5. LLC/CC Messages and Return Codes

Low-Level Codes/Continuity Check produces error messages during generation of the execution program from the macro instruction DLZNN, and during initialization of the control data base. In many cases, additional information is produced on the system console by DL/I, by VSE/VSAM, by VSE (refer to *VSE/SP Messages and Codes*), and other chapters in this book.

LLC/CC Messages

LLC0nnt (Execution Program Generation DLZNN Macro)

LLC001I	<p>PARAMETER <i>parm</i> MISSING, GENERATION ABORTED</p> <p>Explanation: A mandatory parameter <i>parm</i> was omitted.</p> <p>User Response: Complete macro parameter definition. Resubmit the assembly.</p>
<hr/>	
LLC002I	<p>PARAMETER KEYDIS ILLEGAL, GENERATION ABORTED</p> <p>Explanation: The displacement for the key points beyond the end of the root segment.</p> <p>User Response: Check parameters KEYLEN and ROOTLEN. Correct the erroneous parameter.</p>
<hr/>	
LLC003I	<p>PARAMETER KEYDIS AND/OR KEYLEN ILLEGAL, GENERATION ABORTED</p> <p>Explanation: The key overlaps the end of the root segment.</p> <p>User Response: Check parameters KEYDIS, KEYLEN, and ROOTLEN. Correct the erroneous parameter(s).</p>
<hr/>	
LLC004I	<p>PARAMETER LLCDIS ILLEGAL, GENERATION ABORTED</p> <p>Explanation: Displacement and length of the code field do not fit into the root segment.</p> <p>User Response: Check parameters LLCDIS and ROOTLEN. Correct erroneous parameter.</p>
<hr/>	
LLC005I	<p>PARAMETER COMPLEN ILLEGAL, GENERATION ABORTED</p> <p>Explanation: The length of the pointer segment has been specified less than the length of the key in the parameter KEYLEN.</p> <p>User Response: Check parameters COMPLEN and KEYLEN. Correct erroneous parameter.</p>

LLC1nnt (Control Data Base Initialization)

LLC100I	<p>CONTROL DATA-BASE HAS BEEN INITIALIZED</p> <p>Explanation: Successful completion.</p> <p>User Response: Continue.</p>
<hr/>	
LLC101I	<p>INITIALIZATION OF CONTROL DATA-BASE FAILED DL/I STATUS CODE = <i>xx</i></p> <p>Explanation: This message is issued if DL/I returns a non-blank status code when the program attempts to load a root segment.</p> <p>This message may occur under many conditions, for example:</p> <ul style="list-style-type: none"> • Conflicting definitions in DBD and VSAM catalog. • PSB does not indicate PROCOPT=LS. • VSAM catalog was not reset after an unsuccessful attempt to initialize the control data base. <p>User Response: The DL/I status code that is returned after loading has attempted is represented by <i>xx</i>.</p> <ol style="list-style-type: none"> 1. Check the DL/I status code (see Chapter 6, "DL/I Status Codes" on page 111). 2. Check the console output for additional information produced by DL/I (i.e. messages beginning with DLZ). 3. Correct errors. 4. Reset entries in the VSAM catalog by using the Access Method Services commands DELETE and DEFINE. 5. Resubmit the assembly.

LLC/CC Return Codes

After completion of operation, LLC/CC posts a return code into the 2-byte field identified by parm5 of the calling application program. The return code consists of two alphanumeric characters which indicate one of the following conditions:

- Successful completion: bb (two blank characters)
- Rejection of the request: return codes CC, CP, EC, EP, FC, FP, IC, NC, NP.

The request by the application program was not satisfied because of conflicting structures of the parts data base. Data in the parts data base is not changed or is restored to its previous state. You should correct the contents of the data base and/or the input data of the application program.

- Data Base Error: return codes DC, DP, EC, EP, FC, FP.

LLC/CC did not satisfy the request because an unexpected DL/I status code was encountered during execution. The leftmost 240 bytes of the related PCB are saved in a save area. The address of the save area is stored in a fullword which can be addressed by a pointer which is stored at DLZNNEC+12. DLZNNEC is the name of an entry point in LLC/CC. The fullword starting at DLZNNEC+8 points to the last parameter list submitted to DL/I. The actual status of the parts data base is not predictable. However, if the DL/I status code indicates an open error, the data bases are not likely to be affected.

- Return codes starting with letters E and F indicate both an invalid request and a data base error.

bb Successful completion, all affected low-level codes are properly assigned.

Action: Insert the new product-structure relationship between the parts identified by the parameters parm3 and parm4 (in updating mode only). Continue processing.

CC A part other than the part identified by parameter parm3 (that is, a component part of any level) is contained in itself either directly or indirectly. The key of the looping part is posted into the field identified by parameter parm6.

Action: The application program may continue. Correct input and/or parts data base. Resubmit the assembly.

CP The part identified by parameter parm3 (that is, the parent part) is contained in itself either directly or indirectly. The key is posted into the field identified by parameter parm6.

Action: The application program may continue. Correct input and/or parts data base. Resubmit the assembly.

DC Unexpected DL/I status code encountered when accessing the control data base.

Action: Display all available information, discontinue processing of the application program. Display contents of the parameter fields parm1 through parm6. Test whether input has already been processed. Obtain the output of the system console, check messages originating from DL/I and/or VSAM. Reset the VSAM catalog by means of a VERIFY run.

If the control data base is damaged, reinitialize. (Refer to section "Initialization of the Control Data Base" in *IBM System/370 Low-Level*

Code/Continuity Check in Data Language/I DOS/VS Program Reference and Manual, for further information.)

- DP** Unexpected DL/I status code encountered when accessing the parts data base.
- Action:** Display all available information, discontinue processing of the application program. Display contents of the parameter fields parm1 through parm6. Test whether input has already been processed. Obtain the output of the system console, check messages originating from DL/I and/or VSAM. Reset the VSAM catalog by means of a VERIFY run.
- If the parts data base is damaged, use standard reconstruction procedures of DL/I. Resubmit application program only after the error has been fixed.
- EC** Both error conditions CC and DP have occurred.
- Action:** See code DP. Thereafter, correct input and/or parts data base. Resubmit the assembly.
- EP** Both error conditions CP and DP have occurred.
- Action:** See code DP. Thereafter, correct input and/or parts data base. Resubmit the assembly.
- FC** Both error conditions CC and DC have occurred.
- Action:** See code DC. Thereafter, correct input and/or parts data base. Resubmit the assembly.
- FP** Both error conditions CP and DC have occurred.
- Action:** See code DC. Thereafter, correct input and/or parts data base. Resubmit the assembly.
- IP** Erroneous input parameters, for instance:
- Same part specified by parm3 and parm4 of the input parameter list (in updating mode only).
 - A parameter received from the application program is not likely to be a valid address.
- Action:** The application program may continue. Correct input. Resubmit the assembly.
- NC** The component part identified by parameter parm4 cannot be found (in updating mode only).
- Action:** The application program may continue. Correct input and/or parts data base. Resubmit the assembly.
- NP** The parent part identified by parameter parm3 cannot be found.
- Action:** The application program may continue. Correct input and/or parts data base. Resubmit the assembly.

Chapter 6. DL/I Status Codes

DL/I CICS/VS Return Codes

In a CICS/VS transaction, the status of the DL/I-CICS/VS interface should be checked after every DL/I call. This check should be made before checking the DL/I status codes returned in the PCB. There are four types of DL/I calls that are processed by the DL/I-CICS/VS interface:

- Scheduling
- Termination
- Data base
- System

The status of the DL/I interface is returned in the fields TCAFCTR (TCAFRCR in ANS COBOL) and TCADLTR of the CICS/VS TCA or fields UIBFCTR and UIBDLTR if the UIB is used, or UFCTR in RPG-II.

The following errors are returned in fields TCAFCTR and TCADLTR (UIBFCTR and UIBDLTR if used):

Type of Call	TCAFCTR TCADLTR	Explanation
Scheduling	08 01	PSB not in directory (PDIR)
	08 03	Task already scheduled
	08 05	PSB initialization error
	08 06	PSB not in program's ACT entry
	08 07	TERM call when not scheduled
	08 08	Data base call when not scheduled
	08 09	Illegal MPS scheduling call
	08 FF	DL/I not active
	0C 01	Data base not opened or stopped
	0C 02	Online scheduling conflict with MPS task
Termination	08 07	TERM call when not scheduled
	08 FF	DL/I not active
Data Base	08 08	Data base call when not scheduled
	08 FF	DL/I not active
System	01 00	Data base previously started (STRT)
		Data base previously stopped (STOP)
		Tracing already active (TSTR)
		Tracing was not active (TSTP)
	02 00	Load of trace module failed (TSTR)
		DMB failed to initialize (STRT/STOP)
	03 00	TESTCB failed (STRT/STOP)
	04 00	Trace initialization failed (TSTR)
	08 00	System interface in use ('SYSTEMDL' SCHED)
		Invalid CMAX value (CMXT)
08 FF	DMB not in directory (STRT/STOP)	
	DL/I not active	

DL/I CICS/VS Return Codes

A detailed description of the errors can be found in:

- *DL/I DOS/VS Data Base Administration*, “Appendix E” (for system calls).
- *DL/I DOS/VS Application Programming: CALL and RQDLI Interfaces*, Section “Checking the Response to a DL/I Call in a CICS/VS Environment” (for scheduling, termination, and data base calls).

The entries placed by DL/I in the CICS/VS trace table are described in *DL/I DOS/VS Application Programming: CALL and RQDLI Interfaces*, Section “CICS/VS Trace Table Entries for DL/I DOS/VS.”

DL/I PCB Status Codes

bb	Call or command completed. Explanation: Your call or command was correctly completed. User Response: Proceed.
AB	Error in call. Explanation: On a data base call, segment I/O area is required but was not specified in the call. User Response: Correct the program. Note: Should not occur when using DL/I High-Level Programming Interface.
AC	Error in call or command. Explanation: Segment selection contains an error in hierarchical sequence. Possible causes: <ul style="list-style-type: none"> • Segment name specified not found within scope of this PCB. • Level at which this segment appears is out of sequence with that specified by the PCB. • Two segments of the same level are specified in the same call or command. • In a path replace command, duplicate segment names are specified. User Response: Determine which segment was specified more than once, or incorrectly in a single call or command. Correct the call or command and rerun the program.
AD	Error in call or command. Explanation: An invalid function parameter was supplied. Possible causes: <ul style="list-style-type: none"> • A PCB or TERM call or SCHEDULE or TERMINATE command was executed in a batch or MPS environment. • A LOAD call or command was issued in an online or MPS batch environment. • A LOAD call or command was issued in a batch environment using a PCB not having sensitivity. User Response: Correct the program.
AH	Error in call or command. Explanation: No segment selection was specified in a data base function call or command other than GET NEXT or GET NEXT IN PARENT. Call or command type requires at least one segment selection; none was specified. User Response: Determine which segment name was required but not specified. Correct the call or command and rerun the program.
AI	I/O, system, or user error. Explanation: Open error. Possible causes: <ul style="list-style-type: none"> • Error in VSE job control statements. • Data base being opened for other than load mode and has not been loaded. User Response: Check job control statements; ensure that file name is the same as the name specified on the DATASET card of the DBD. DIBSEGM contains the name of the file name of the file that could not be opened, in batch and MPS batch applications.
AJ	Error in call or command. Explanation: Segment qualification or the command code was invalid. Possible causes: <ul style="list-style-type: none"> • Invalid relational operators. • Missing right parenthesis from SSA. • DLET call or DELETE command has qualified segment selection. • REPL call or REPLACE command has qualified segment selection. • ISRT call or INSERT command has the last segment selection qualified. • Field length incorrectly specified. • Invalid command codes. • Path insert with logical child. • Invalid Boolean connector.

- Value other than blank, '*', or '(' in column 9 of SSA.
- Call interface SSA is longer than 304 bytes in the ISC environment.

User Response: Correct the program.

AK Error in call or command.

Explanation: A field name or boolean operator is invalid;

- A field name does not correspond to any field name specified in the DBD.
- The specific field belongs to either the concatenated key or the logical parent of a logical child definition for a data base which is in partial load process.
- An invalid boolean operator was used.

User Response: Correct the program or run prefix resolution and prefix update utilities before accessing logical related fields in a partially loaded data base.

AM Error in call or command.

Explanation: Call or command function not compatible with processing option, segment sensitivity, or state of the data base (load processing is incomplete; for example, using a PCB with an alternate processing sequence for a secondary INDEX data base which isn't yet built). Possible causes:

- Processing option of L and call function not insert or command function not LOAD.
- DELETE, REPLACE, or INSERT command (DLET, REPL, or ISRT call) without corresponding segment sensitivity.
- DELETE or INSERT command (DLET or ISRT call) for the root of a secondary data structure or any of its physical parent segment types in the data base.
- A secondary index is processed as a data base itself and attempts are made to delete or insert a segment or to replace the system-maintained fields.
- INTO/FROM used for parent segments without path sensitivity.
- An attempt was made to unload a secondary index using the HD unload utility.
- Command code D used for a path retrieval call without path sensitivity.

Note: Path sensitivity must also be specified for segments retrieved by path calls, even if their lowest-level SSA has no D command code.

- Path insert call using XPCC support (cross address space) without path sensitivity.
- Path insert call using remote PSB without path sensitivity.
- Path insert call with field level sensitivity but without path sensitivity.

User Response: Correct the program, PSB, or system definition.

AO I/O error.

Explanation: There is a SAM or VSAM I/O error.

User Response:

- DELETE, REPLACE, or INSERT command (DLET, REPL, or ISRT call) without having finished the load process. Either the partial load steps were not completed using a parameter card with PLC, or the prefix update utility was not run to complete the load process if any work files were built during loading.
- The secondary index data base must be built first, before using a PCB with an alternate processing sequence referencing this index is possible.

DA Error in call or command.

Explanation: Segment key field has been changed on a REPL call or REPLACE command.

User Response: Correct the request.

DJ Error in call or command.

Explanation: No previous successful GET HOLD call or GET command.

User Response: Check and correct.

DX	<p>Error in call or command.</p> <p>Explanation: Violated delete rule. Review delete rule in SEGM statement of DBDGEN in <i>DL/I DOS/VS Resource Definition and Utilities</i>.</p> <p>User Response: Correct the program.</p>
-----------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

GA	<p>Call or command is completed.</p> <p>Explanation: Crossed hierarchical boundary into higher level. This status code is returned only on GET NEXT commands or calls with unqualified segments.</p> <p>User Response: This status code is for your information, only.</p>
-----------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

GB	<p>Call or command is not completed.</p> <p>Explanation: This is the end of the file; the position beyond the last segment is reached.</p> <p>User Response: None required.</p>
-----------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

GE	<p>Call or command is not completed.</p> <p>Explanation: Segment has not been found (GET call or command). For an ISRT call or INSERT command, the parent of the segment to be inserted was not found.</p> <p>User Response: None required.</p>
-----------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

GG	<p>Call or command is not completed.</p> <p>Explanation: This status code is returned only to application programs with processing options of 'GON[P]' or 'GOT[P]', after the program has issued one of the get calls. The segment the program was trying to retrieve contained an invalid pointer because of update activity in the data base going on concurrently with your program processing. The position is set to the beginning of the data base. The key feedback area (KFB) will contain the key of the last root segment successfully accessed.</p> <p>User Response: Terminate the program or continue processing, considering the position in the data base. The request for which the status code 'GG' was received might be successful if issued again.</p>
-----------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

GK	<p>Call or command is completed.</p> <p>Explanation: Different segment type at same level returned. This status code is returned only on unqualified GET NEXT calls or commands.</p> <p>User Response: This status code is for your information, only.</p>
-----------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

GP	<p>Error in call or command.</p> <p>Explanation: No parent established (or parent deleted) for this GET NEXT IN PARENT call or command, or the requested segment level is not lower than the parent level.</p> <p>User Response: None required.</p>
-----------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

II	<p>Call or command is not completed.</p> <p>Explanation: The segment that you tried to insert already exists in the data base. Possible causes:</p> <ul style="list-style-type: none"> • A segment with an equal physical twin sequence field already exists for the parent. • A segment with an equal logical twin sequence field already exists for the parent. • A logical parent has a logical child pointer, but the logical child does not have a logical twin pointer and the segment being inserted is the second logical child for the logical parent. • A physical parent has a physical child pointer but the physical child does not have a twin pointer (POINTER=NOTWIN was specified on the SEGM macro for the physical child segment) and another segment of the same type already exists under the parent. • For an HDAM data base, POINTER=NOTWIN was specified on the SEGM macro for the root segment. However, a root segment already exists in the data base under the root anchor point where you wanted to insert the segment. <p>User Response: Correct the error.</p>
-----------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

IX	<p>Call or command is not completed.</p> <p>Explanation: Violated the insert rule. Review the insert rule in SEGM statement of DBDGEN in <i>DL/I DOS/VS Resource Definition and Utilities</i>. Possible causes:</p> <ul style="list-style-type: none"> • Insert of logical child (insert rule of logical parent is physical) and the logical parent does not exist. • Insert of logical child and logical parent (insert rule is logical or virtual) and the logical parent does not exist and, in the user I/O area, the key of the logical parent does not match the corresponding key in the concatenated key in the logical child. • Insert of logical child (insert rule of logical parent is virtual, and logical parent exists) and, in the segment I/O area, the key in the logical parent does not match the corresponding key in the concatenated key in the logical child. <p>User Response: Correct the program.</p>
-----------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

KA	<p>Numeric truncation error.</p> <p>Explanation: During automatic conversion of a numeric field from one format to another format, an intermediate or final field was not large enough to contain the significant digits in the 'from' field.</p> <p>User Response: Correct the program.</p>
-----------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

KB	<p>Character truncation error.</p> <p>Explanation: During automatic length conversion of a character field the 'to' field was not large enough to contain all the non-blank characters moved from the 'from' field. The field is moved left justified.</p> <p>User Response: Correct the program.</p>
-----------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

KC	<p>Invalid packed decimal or zoned decimal format.</p> <p>Explanation: During automatic conversion, a 'from' field character was encountered that is not a valid packed decimal or zoned decimal character. botesp. Correct the program.</p>
-----------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

KD	<p>Type conflict for conversion.</p> <p>Explanation: A field-to-field conversion was requested that was not supported by DL/I.</p> <p>User Response: Do conversion in field exit routine.</p>
-----------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

KE	<p>NOREPL=NO violation.</p> <p>Explanation: The user attempted to modify a field that was not replace sensitive (REPLACE=NO was specified in the SENFLD statement for PSBGEN). The call is not completed.</p> <p>User Response: Correct the program or specify REPLACE=YES.</p>
-----------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

LB	<p>Call or command is not completed.</p> <p>Explanation: The segment that you tried to load already exists in the data base. Possible causes:</p> <ul style="list-style-type: none"> • A segment with an equal physical twin-sequence field already exists for the parent. • A physical parent has a physical child pointer but the physical child does not have a twin pointer (POINTER=NOTWIN was specified on the SEGM macro for the physical child segment) and another segment of the same type already exists under the parent. • A segment with an equal logical twin-sequence field already exists for the parent. • For unique segment support, another segment under the same parent at the same type level exists. • For an HDAM data base, POINTER=NOTWIN was specified on the SEGM macro for the root segment. However, a root segment already exists in the data base under the root anchor point where you wanted to insert the segment. <p>User Response: Correct the error.</p>
-----------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

LC	<p>Call or command is not completed.</p> <p>Explanation: Key field of segments is out of sequence.</p> <p>User Response: Check and correct.</p>
-----------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------

LD	<p>Call or command is not completed.</p> <p>Explanation: No parent has been loaded for this segment, or partial data base load starts without a root segment insert.</p> <p>User Response: Check and correct.</p>
-----------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

LE	<p>Call or command is not completed.</p> <p>Explanation: Sequence of child segment types for the parent segment type is not the same as the sequence in the DBD.</p> <p>User Response: Check and correct.</p>
-----------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

N	<p>Nx Status Codes</p> <p>Explanation: DL/I makes internal calls while performing index maintenance. If an error occurs in one of these calls, DL/I replaces the first character of the status code returned for the call with 'N'. This serves as notice to the application programmer that an error occurred, but not in one of the application calls.</p>
----------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

NA	<p>Call or command is not completed.</p> <p>Explanation: The user tried to replace data in an index source segment that is used in search or subsequence fields of the index pointer segment, while the secondary index is used as the processing sequence.</p> <p>User Response: Correct the program.</p>
-----------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

NE	<p>Error in some previous insert call or command, or system error. The INSERT, DELETE, or REPLACE command (ISRT, DLET, or REPL call) was completed, as if the NE status code were a warning.</p> <p>Explanation: The user tried to delete or replace an index source segment or to insert an index source segment that had not been physically removed because of logical relationship requirements, and the corresponding index pointer segment could not be found. Possible causes:</p> <ul style="list-style-type: none"> • During some previous insert call or command an index source segment was inserted with data in search and subsequence fields equal to an already existing index source segment. An NI status code had been returned with that call and backout had not been performed. • Some error had occurred during reorganization of the data base. <p>User Response:</p> <ul style="list-style-type: none"> • For DELETE command (DLET call), none. The index source segment which produced the duplicate key is now removed. <p>This status code is returned to notify the user that an unexpected condition was detected in the data base. Information describing the occurrence, including data bases, PSB, and segment being used should be provided to the person responsible for data base administration.</p> <ul style="list-style-type: none"> • For REPLACE command (REPL call), delete the segment and reinsert it with proper search and subsequence data. Information describing the occurrence of this status code should be provided to the person responsible for data base administration.
-----------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

NI	<p>VSAM open error or duplicate key for index data base.</p> <p>Explanation: Check the error message printed on the system log device to get detailed information on the error. Possible causes for being unable to open the index data base are:</p> <ul style="list-style-type: none"> • Error in VSE job control statements. • Control interval size, key length, or relative key position specified in the VSAM DEFINE macro do not match the values specified for DBD generation. • The processing option was L or LS but the data base was not empty, or the data base was empty and the processing option was not L or LS.
-----------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

A possible cause for having a duplicate key is that an index source segment was inserted with data in search and subsequence fields equal to an already existing index source segment. The index source segment is inserted, the index pointer segment is not inserted.

User Response: Delete the segment and insert it with a unique key. For a subsequent delete call for the index source segment two different situations have to be distinguished:

- If the index source segment just inserted and the one with the same search and subsequence fields point to different index target segments, the delete call returns an NE status code.
- If the two index source segments point to the same index target segment, the first delete call for one of the index source segment removes the index pointer segment created by the first inserted index source segment. The second delete call for the remaining index source results in an NE status code.

NO I/O error.

Explanation: Possible causes:

- An I/O error occurring during processing of an index, either in the index or indexed data base.
- Insertion of a duplicate secondary index pointer segment was attempted at load time.

User Response: Check and correct.

RX Error in call or command.

Explanation: Violated replace rule. Review replace rule in SEGM statement of DBDGEN in *DL/I DOS/VS Resource Definition and Utilities*.

User Response: Correct the program.

TA PSB not in PSB Directory.

Explanation: The PSB named in the SCHEDULE command does not appear in the PSB directory.

User Response: Verify that the proper PSB name is being used. Add PSB name to ACT program entry using DLZACT macro if required.

TC Task already scheduled.

Explanation: The calling program has already successfully issued a SCHEDULE command that has not been followed by a TERMINATE command.

User Response: Verify program logic to determine why a SCHEDULE command was issued when a PSB command was already scheduled. Add TERMINATE command if required.

TE PSB initialization error.

Explanation: The PSB could not be initialized by DL/I online initialization.

User Response: Determine why PSB failed to initialize by consulting console log produced during DL/I initialization. Correct initialization problem and rerun application program.

TF PSB not authorized for this program.

Explanation: The PSB named in the SCHEDULE command is not defined in the application control table, or is incorrectly defined.

User Response: Verify that the proper PSB name is being used. Add the PSB name to ACT program entry using DLZACT macro if required.

TG TERM attempted when not scheduled.

Explanation: A TERMINATE command was issued when a PSB had not previously been scheduled for the task.

User Response: Verify program logic to determine why a TERMINATE command was issued when a PSB was not scheduled. Delete TERMINATE command if required.

TH	Data access attempted when not scheduled. Explanation: A data base command was issued for a PSB which had not previously been scheduled for the task. User Response: Verify program logic to determine why a data base command was issued when a PSB was not scheduled. Add SCHEDULE command if required.
TI	Invalid path INSERT. Explanation: Data transfer was not requested for all parent segments between the first parent segment requesting data transfer and the object segment. User Response: Verify that FROM is coded for every segment after the first parent segment that is being inserted. After the command is corrected: translate, compile, and re-execute the application program.
TJ	DL/I not active. Explanation: The DL/I interface has been terminated or a DL/I initialization failed. User Response: Terminate and reinitialize online system and DL/I before attempting to rerun application program.
TK	Data base not active. Explanation: One or more DBD entries associated with the PSB are stopped. User Response: Start data base(s) using DL/I STRT call before attempting to rerun application program.
TL	Scheduling conflict with MPS task. Explanation: A scheduling conflict occurred with a currently active MPS batch partition. User Response: Rerun application program after completion of MPS task that uses same data base. Regenerate DL/I nucleus specifying PI=YES if required.
TN	Invalid system DIB address. Explanation: The address of the system DL/I Interface Block has been destroyed or made invalid. User Response: Verify that application program is not overlaying DL/I storage (a possible cause is execution of first DL/I command before execution of translator generated initialize call). If it is not, call your IBM Support Center for help.
TO	Path Replace Error. Explanation: A segment or segments being replaced in a path replace command were not retrieved by a previous GET command with no other intervening commands. Possible causes: <ul style="list-style-type: none">• Data was not transferred on previous GET command for a parent segment being replaced.• Variable length segment retrieved on previous GET command is specified as fixed length on this command.• Fixed length segment retrieved on previous GET command is specified as variable length on this command.• Fixed length segment retrieved on previous GET command does not have same length as same segment in this command. User Response: Verify that the segments being replaced are a subset of the segments retrieved on the previous GET command. Correct the appropriate command and translate, compile and execute the application program.

TP	Invalid PCB Index. Explanation: The PCB index being used is not valid for the PSB currently scheduled or the command was LOAD and the PCB PROCOPT was not L or LS. User Response: Verify that the PCB index being used is not valid for the PSB currently scheduled; or that PROCOPT is not L or LS. Correct the command and translate, compile, and execute the application program; or regenerate the PCB with correct PROCOPT specified and rerun program.
<hr/>	
V1	Program error. Explanation: Invalid length for variable length segment. This status code applies only to the virtual insert rule. The LL field of the variable length segment is either too large or too small. The length of the segment must be equal to or less than the maximum length specified in the DBD. The length must be long enough to include the entire sequence field. The LL field for the variable segment must also include the length of any related virtual logical child sequence field if the physical parent is variable and there is a bidirectional relationship. User Response: Correct the program.
<hr/>	
V2	SEGLength required and not specified or is zero or negative. Explanation: Specification of a SEGLength was required with this command, but was missing or incorrect. User Response: Determine proper SEGLength for segment referenced in failing command. Add SEGLength to command.
<hr/>	
V3	FIELDLENGTH required and not specified or is zero or negative. Explanation: Specification of FIELDLENGTH was required with this command, but was missing or incorrect. User Response: Determine proper FIELDLENGTH for segment referenced in failing command.
<hr/>	
V4	Invalid length for variable length segment. Explanation: The SEGLength specified for a variable length segment in this command was incorrect. Possible causes: <ul style="list-style-type: none">• Segment length in first two bytes of the segment greater than maximum segment length supported by DL/I.• Segment length in first two bytes of the segment greater than maximum segment length specified by SEGLength in the command.• Length of segment retrieved greater than segment length specified by SEGLength in the command, or the length of the I/O area in PL/I.• The length of the segment retrieved is greater than the segment length specified in the first two bytes of the segment. User Response: Determine proper SEGLength for variable length segment referenced in failing command. Increase SEGLength or size of I/O area if required.
<hr/>	
V5	OFFSET greater than SEGLength or is zero or negative. Explanation: Within a command relating to segments having a logical relationship; the OFFSET specified is zero, a negative value, or has a value greater than that specified by SEGLength. User Response: Determine the proper value for OFFSET for the segment referenced in the failing command and update, translate, compile, and re-execute the application program.
<hr/>	
V8	Key Feedback Length missing or invalid Explanation: The value specified for FEEDBACKLEN is not valid or has not been specified. User Response: Determine the proper value for FEEDBACKLEN and update, translate, compile, and re-execute the application program.

XD I/O error.

Explanation: An error occurred when the data base buffers were being written out to secondary storage during processing of a checkpoint (CHKP) call.

User Response: Check and correct.

XH Call or command is not completed.

Explanation: Data base logging was inactive during checkpoint (CHECKPOINT command or CHKP call) processing.

User Response: Ensure that data base logging is active during checkpoint processing.

XR Error during checkpoint processing for MPS restart

Explanation: For application programs using the MPS Restart facility, a DL/I checkpoint was not taken for one of the following reasons:

- A VSE CHKPT was not taken before the DL/I CHKP.
- The VSE CHKPT failed.
- A Temporary Storage error occurred during DL/I CHKP processing.

User Response: If the status code was returned because of failure or error, correct the situation described in the accompanying error message. Otherwise, make sure that a VSE CHKPT is coded before each DL/I CHKP in the program.

Communicating Your Comments to IBM

IBM Data Language/I Disk Operating System/
Virtual Storage (DL/I DOS/VS)
Messages and Codes
Version 1 Release 8
Publication No. SH12-5414-10

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of the book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

If you are mailing a readers' comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF form and either send it postage-paid in the United States, or directly to:

IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany

- If you prefer to send comments by FAX, use this number:
 - (Germany): 07031-16-3456
 - (Other countries): (+49)+7031-16-3456
- If you prefer to send comments electronically, use this network ID:
INTERNET: s390id@de.ibm.com

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies.

Readers' Comments — We'd Like to Hear from You

IBM Data Language/I Disk Operating System/
Virtual Storage (DL/I DOS/VS)
Messages and Codes
Version 1 Release 8

Publication No. SH12-5414-10

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>				

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>				
Complete	<input type="checkbox"/>				
Easy to find	<input type="checkbox"/>				
Easy to understand	<input type="checkbox"/>				
Well organized	<input type="checkbox"/>				
Applicable to your tasks	<input type="checkbox"/>				

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? Yes No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

Company or Organization

Phone No.



Fold and Tape

Please do not staple

Fold and Tape

PLACE
POSTAGE
STAMP
HERE

IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany

Fold and Tape

Please do not staple

Fold and Tape



File Number: S370/S390-40
Program Number: 5746-XX1



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SH12-5414-10

