SH12-5414-6

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

Program Product

Messages and Codes

Program Number 5746-XX1

IBM

Seventh Edition (June 1979)

This edition, SH12-5414-6, applies to Version 1, Release 5 (Version 1.5) of IBM System/370 Data Language/I Disk Operating System/Virtual Storage (DL/I DOS/VS), Program Number 5746-XX1, and to all subsequent versions and modifications until otherwise indicated in new editions or Technical Newsletters. It supersedes SH12-5414-5 with Technical Newsletter SN24-5612. Changes are continually made to the information herein; any such changes will be reported in subsequent revisions or Technical Newsletters.

Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Summary of Amendments
For a list of changes, see page iii.

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

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation Programming Publications, Dept. G60, PO Box 6, Endicott, New York, U.S.A. 13760. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Summary of Amendments for DL/I DOS/VS Messages and Codes

Summary of Amendments for SH12-5414-6 Version 1.5

This major revision supports Version 1, Release 5 (Version 1.5) of DL/I DOS/VS, Program Number 5746-XX1. It adds and updates pages with information related to:

- Field Level Sensitivity
- Extended Logical Relationships
- Unique Segment Support
- All-Partition MPS Support
- IPCS Dump Hook Support
- Selective Log Print

Miscellaneous additions, improvements, and corrections also appear throughout this manual.

Summary of Amendments for SH12-5414-5 Updated by SN24-5612 Version 1.4

This Technical Newsletter contains changes relating to the use of FBA (Fixed Block Architecture) devices with DL/I DOS/VS Version 1, Release 4.

Summary of Amendments for SH12-5414-5 Version 1.4

This major revision supports Version 1, Release 4 (Version 1.4) of DL/I DOS/VS, Program Number 5746-XX1. It has been extensively reformatted and adds/updates messages for Performance Improvements and Intersystem Communication.

Miscellaneous additions, improvements, and corrections also appear throughout this manual.

Preface

This manual provides information on all DL/I messages and status codes for persons operating the DL/I system and for application programmers and data base administrators who may need this information.

Because DL/I DOS/VS is a functional subset of the IBM Information Management System/Virtual Storage (IMS/VS), some specific IMS or OS terms are used in this manual. These terms are used to allow easy reference to the documentation of the related systems.

Related Publications

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

DL/I DOS/VS System/Application Design Guide, SH12-5413.

DL/I DOS/VS Application Programming Reference Manual, SH12-5411.

DL/I DOS/VS Utilities and Guide for the System Programmer, SH12-5412.

DL/I DOS/VS Logic Manual, LY12-5016.

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

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

For DOS/VS messages and return codes:

DOS/VSE Messages, GC33-5379.

VSE/VSAM Messages and Codes, SC24-5146.

DOS/VS Supervisor and I/O Macros, GC33-5373.

Users employing DL/I DOS/VS in an online environment should have access to the following CICS/DOS/VS publications:

CICS/VS System Programmer's Guide (DOS/VS), SC33-0070.

CICS/VS System Programmer's Reference Manual, SC33-0069.

CICS/VS Application Programmer's Reference Manual, SC33-0079.

CICS/VS System/Application Design Guide, SC33-0068.

DL/I Message Format

The following conventions are used in formatting the 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 contains the message formats in alphabetical order, identifies the function and/or macro that issues each message, and provides the chapter number where the messages are documented.

Message Format*	Function and/or Macro	Chapter
DBDnnn	DBD Generation - DBD macro	2
DGENnnn	DBD Generation - DBDGEN macro	2
DLZnnnt	DL/I System Messages	1
DLZ1nnn	Nucleus Generation - DLZACT macro	4
DLZ3nnn	Nucleus Generation - CALLDLI macro	4
DMANnnn	DBD Generation - DATASET macro	2
EXTnnn	DBD Generation	2
FINnnn	DBD Generation - FINISH macro	2
FLDnnn	DBD Generation - FIELD macro	2
LCHDnnn	DBD Generation - LCHILD macro	2
LLCnnnt	LLC/CC Error Messages	**
PCBnnn	PSB Generation - PCB macro	3
PGENnnn	PSB Generation - PSBGEN macro	3
SEGnnn	PSB Generation - SENSEG macro	3
SEGMnnn	DBD and PSB Generation - SEGM macro	2
SFLDnnn	DBD Generation - SENFLD macro	2
VFLDnnn	DBD Generation - VIRFLD macro	2
XDFLDnnn	DBD Generation - XDFLD macro	2

* In the message format, nnn indicates the message number and t (if shown) indicates the message type. In an actual message, t will be either I (for an information-type message) or A (for an action-type message).

** The error messages and return codes that may be encountered during generation and execution of LLC/CC facilities are described in *IBM System/370 Low-Level Code/Continuity Check in DL/I DOS/VS Program Reference and Operations Manual,* SH20-9046.

Contents

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:

```
DLZxxxI 'text' or DLZxxxA 'text'
```

DLZ	indicates a DL/I message
xxx	is the message number
I	indicates an information message
A	indicates an operator response is required
text	is a brief description of the problem.

Many of these messages produce a storage dump. 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. The numeric part of that code corresponds to the numeric portion of the applicable DL/I message number as listed below. The code normally begins with D but it begins with E if the termination cannot be noted on the transient data destination CSMT.

For any message that contains a VSAM error code, the appropriate VSAM action should be taken for that error code as stated in *DOS/VS Supervisor and I/O Macros.*

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.

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, it means that the error was not detected by DL/I and that the termination routine received control from the DOS/VS supervisor. The only way to determine the reason for the error, therefore, 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.

Action: None.

DLZ002I 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 2 =	Address of SCD
Register 3 =	Address of user save area if abnormal termination 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 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 through 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'10' - Buffer pool or data base damage. X'08' - Buffer pool unload and close complete.
90(5A)	1	STXIT AB reason code as returned by DOS/VS (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:

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.

Action: Take appropriate action.

DLZ003I filename STOPPED I/O ERROR

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

Action: Reexecute the application program.

DLZ004I filename I/O ERROR - VSAM CODE=xxyy

Explanation: An I/O error occurred during attempt to read or write to the file 'filename'. The VSAM return code is included as xx. The VSAM error code is included as yy. 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.

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

DLZ005I filename STOPPED OUT OF SPACE - VSAM CODE=xxyy

Explanation: An attempt was made to add a record to the file 'filename' and there was no more EXTENT allocation. The VSAM return code is included as xx. The VSAM error code is included as yy.

Action: 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.

Action: 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=return-code

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

Action: If the return code 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.

DLZ008I INVALID, DUPLICATE, OR NO DBDNAME SPECIFIED IN HDBFR OR HSBFR STMT

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

- The DBDNAME in an HDBFR or HSBFR statement is not specified in the PSB.
- The DBDNAME is missing in an HSBFR statement.
- The DBDNAME encountered in an HDBFR statement is the same as a DBDNAME found in a previous HDBFR statement.
- An HSAM, simple HSAM, HDAM, or HIDAM DBD is specified in an HSBFR statement.
- The DBDNAME in an HDBFR statement was for HSAM or simple HSAM, or for HISAM or simple HISAM when the processing option did not include load or insert.

The subparameter that caused the error, plus all parameter information following it that was already read, are listed before this message appears.

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

DLZ009I NO OF SUBPOOLS NOT EQUAL TO NO OF HDBFR STMTS. DEFAULTS SET

Explanation: The number of buffer subpools specified in the parameter statement was either greater or smaller than the number of HDBFR statements, or not all DBDs were assigned with HDBFR statements. DL/I adjusts the number of subpools accordingly. See Chapter 7 of the *DL/I DOS/VS Utilities and Guide for the System Programmer* for details.

Action: None.

DLZ010A ENTER DL/I PARAMETER INFORMATION

Explanation: One of the following occurred:

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

Action: Enter the correct DL/I parameter information via SYSLOG. Refer to the *DL/I DOS/VS Utilities and Guide for the System Programmer* for the correct format. In case 2, this message is preceded by message DLZ014A.

DLZ011I REQUIRED NUCLEUS MODULE MISSING NAME=modname

Explanation: While building the DL/I nucleus loadlist, the named module was not found in a DOS/VS core image library. The program is canceled.

Action: Catalog the module in a DOS/VS core image library and reexecute the application program.

DLZ012I APPLICATION PROG OR DL/I CONTROL BLOCK NOT FOUND. NAME=modname

Explanation: While building a loadlist entry, the named module was not found in a DOS/VS core image library.

Action: Catalog the named module in a DOS/VS core image library and reexecute the application program.

DLZ013I INVALID DMB REFERENCE

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

Action: Enter CANCEL, DUMP, or GO in response to message DLZ040A.

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.

Action: Reply 'CANCEL' on SYSLOG to terminate the DL/I application program, or depress the END button to continue. An EOB character will be transmitted and message DLZ010A will be written.

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.

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

Action: Correct the error and reexecute the job.

DLZ016I 'filename' MODCB ERROR - VSAM CODE=xx

Explanation: An error occurred during execution of the VSAM MODCB macro for 'filename'. The VSAM error code describes the type of error.

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

DLZ017I PSB VERSION INVALID. INIT TERMINATED

Explanation: While loading the requested PSB, initialization detected that the block was created prior to V1.1.

Action: Rebuild the PSB using the application control block utility DLZUACB0 provided with the current level of DL/I DOS/VS.

DLZ018I DBD VERSION INVALID DBDNAME=dbdname. INIT TERMINATED

Explanation: While loading the named DBD, initialization detected that the block was created prior to V1.1.

Action: Rebuild the block using the application control block utility DLZUACB0 provided with the current level of DL/I DOS/VS.

DLZ019I BUFFER NUMBER IN {HDBFR|HSBFR} STMT 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 Chapter 7 of the *DL/I DOS/VS Utilities and Guide for the System Programmer* for information on the requirements.

Action: None.

DLZ020I 'filename' OPEN ERROR - VSAM CODE=xx

Explanation: An error occurred during an attempt to open a DL/I file 'filename'. The VSAM error code is indicated as xx.

Action: Take the appropriate VSAM action for the error code. If 'filename' indicates the log file, the error condition should be removed. If necessary, data base recovery can be done and the job resubmitted. If 'filename' 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 'filename' CLOSE ERROR - VSAM CODE=xx

Explanation: An error occurred during an attempt to close the DL/I file 'filename'. The VSAM error code is indicated as xx.

Action: If 'filename' indicates a data base data set, take the appropriate VSAM action for the error code and reexecute the application program. If 'filename' 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 'filename' 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 'filename'.

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

DLZ023I 'filename' OPEN ERROR - INVALID PROCOPT

Explanation: One of the following occurred:

1. A PROCOPT≠L was specified for the PCB accessing the file and the file was empty.
2. A PROCOPT=L was specified for the PCB accessing the file and the file was not empty.

Action:

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

DLZ024I 'filename' SHOWCB ERROR - VSAM CODE=xx

Explanation: An error occurred during execution of the VSAM SHOWCB macro for 'filename'. The VSAM error code describes the type of error.

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

DLZ025I 'filename' 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 'filename'.

Action: The user should perform one or more of the following prior to reexecuting the application program:

1. Execute a new DBD generation.
2. Delete the DMB and execute the ACB creation and maintenance utility.
3. 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.

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

DLZ027I 'filename' 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 'filename'.

Action: The user should perform one or more of the following prior to reexecuting the application program:

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

DLZ028I 'filename' 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 'filename'.

Action: The user should perform one or more of the following prior to reexecuting the application program:

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

DLZ029I INVALID DBDNAME IN HDBFR OR HSBFR STMT. NAME=dbdname

Explanation: One of the following invalid conditions occured:

- The DBDNAME identified in the message text (dbdname) is not specified in any of the PSBs.
- An HDAM or HIDAM DBD is specified in a HSBFR statement.

This message is followed by DLZ040A.

Action: Enter CANCEL, DUMP, or GO in response to message DLZ040A.

DLZ030A INVALID SLC TABLE LOADED

Explanation: A module name specified in the SLC parameter in the ACT (application control table) is not in the storage layout control table. This message is followed by DLZ040A.

Action: Enter CANCEL, DUMP, or GO in response to message DLZ040A.

DLZ031I DYNAMIC TRANSACTION BACKOUT IS NOT ENABLED

Explanation: Program isolation requires CICS/VS dynamic transaction backout to be active which in turn requires CICS/VS jounaling. This message, therefore, is issued if either:

1. Dynamic transaction backout program (DFHDBP) cannot be found in the CICS/VS processing program table (PPT), or
2. CICS/VS journaling was suppressed for DL/I.

This message is followed by DLZ040A.

Action: Respond to message DLZ040A as follows:

1. DUMP or CANCEL - Regenerate a DL/I online nucleus without program isolation or, regenerate CICS/VS with dynamic transaction backout active or, do not suppress CICS/VS journaling when reinitializing DL/I online.
2. GO - Initialization of DL/I will continue with a dummy nucleus which in effect no-ops all DL/I calls.

DLZ032A THE SYSTEM LOG JCT ENTRY IS MISSING

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

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

DLZ033I INTERSYSTEM COMMUNICATION FAILURE. REASON IS xxxxxxx

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

xxxxxxx	Meaning
ISC MOD	CICS/VS ISC support not generated.
INVPARM	ISC detected invalid call parameter list.
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.
INT ERR	Internal ISC error.

Action: If xxxxxxx is:

ISC MOD Consult the *CICS/VS System Programmers Reference Manual* 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 additionl 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 CICS/VS 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.

INT ERR Save the log and dump for your IBM Programming System Representative.

DLZ034I MODULE xxxxxxxx NOT IN CICS PPT

Explanation: During DL/I online initialization, module xxxxxxxx was specified in the online nucleus generation ACT statement, but could not be found in the CICS program processing table (PPT).

Action: This message is issued each time a specified module is not found in the PPT. After all missing modules are identified, message DLZ051I is issued. See message DLZ051I for appropriate action.

DLZ039I FIELD EXIT ROUTINE xxxxxxxx NOT FOUND

Explanation: During DL/I online initialization, field exit routine xxxxxxxx could not be found in a DOS/VS core image library. This message is followed by DLZ056I.

Action: 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, OR GO TO CONTINUE

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

Action: Take the appropriate action based on the previous error message printed:

1. CANCEL causes the DOS/VS CANCEL macro to be invoked for the partition.
2. DUMP causes the DOS/VS DUMP macro to be issued for the partition.
3. GO causes the program to continue.

DLZ041I INVALID RESPONSE

Explanation: A response of other than GO, CANCEL, or DUMP was received after message DLZ040A.

Action: Reply correctly to DLZ040A.

DLZ042I HSAM INVALID ATTRIBUTE ON-LINE - PSBNAME=psbname

Explanation: During DL/I online initialization, a PSB was found that accessed an HSAM DBD. The PSB is identified by psbname.

Action: Respond to message DLZ040A as follows:

1. DUMP or CANCEL - Regenerate a DL/I online nucleus without the named PSB, or change the DL/I access method prior to reinitializing DL/I online.
2. GO - Application programs referencing the named PSB will not be scheduled.

DLZ043I LOAD INVALID ATTRIBUTE ON-LINE - PSBNAME=psbname

Explanation: During DL/I online initialization, a PSB was found that contained a PCB with PROCOPT=L specified. The PSB is identified by psbname.

Action: Respond to message DLZ040A as follows:

1. DUMP or CANCEL - Regenerate a DL/I online nucleus without the named PSB, or change the PCB PROCOPT specification prior to reinitializing DL/I online.
2. GO - Application programs referencing the named PSB will not be scheduled.

DLZ044I PSB NOT FOUND BY LOCATE PSBNAME=psbname

Explanation: During DL/I online initialization, the PSB identified by psbname was not found in a DOS/VS core image library.

Action: Respond to message DLZ040A as follows:
1. DUMP or CANCEL - Link-edit the named PSB into a core image library prior to reinitializing DL/I online.
2. GO - Application programs referencing the named PSB will not be scheduled.

DLZ045I CICS FCT NOT LOCATED

Explanation: During DL/I online initialization, the CICS file control table could not be found.

Action: Respond to message DLZ040A as follows:
1. DUMP or CANCEL - Restart CICS specifying a CICS file control table.
2. GO - Field TCAFCTR (TCAFCRC in ANS COBOL) contains X'08'. Application programs should test this field after requesting DL/I services, by specifying the INVREQ parameter of the DFHFC TYPE=CHECK macro, and further processing should be based on the result.

DLZ046I DBD NOT DEFINED IN CICS FCT DBDNAME=dbdname

Explanation: During DL/I online initialization the named DBD could not be matched with a CICS FCT entry.

Action: Respond to message DLZ040A as follows:
1. DUMP or CANCEL - Generate a CICS FCT which includes the named DBD and reinitialize CICS.
2. GO - Application programs referencing PSBs that depend on the named DBD will not be scheduled.

DLZ047I DBD NOT FOUND BY LOCATE DBDNAME=dbdname

Explanation: During DL/I online initialization, a DMB was not found in a DOS/VS core image library for the named DBD.

Action: Respond to message DLZ040A as follows:
1. DUMP or CANCEL - Link-edit the named DMB into a core image library prior to reinitializing DL/I online.
2. GO - Application programs referencing PSBs which depend on the named DBD will not be scheduled.

DLZ048I RANDOMIZING MODULE NOT FOUND BY LOCATE DBDNAME=dbdname

Explanation: During DL/I online initialization, the randomizing routine for the named DBD was not located in the DOS/VS Core Image Library.

Action: Respond to message DLZ040A as follows:
1. DUMP or CANCEL - The randomizing routine must be link-edited into a core image library prior to reinitializing DL/I online.
2. GO - Application programs referencing PSBs that depend on the named DBD will not be scheduled.

DLZ049I NO VALID DMBS FOUND

Explanation: During DL/I online initialization, no DMBs referenced by PSBs were found in a DOS/VS core image library.

Action: Respond to message DLZ040A as follows:
1. DUMP or CANCEL - Link-edit the DMBs into a core image library prior to reinitializing DL/I online.
2. GO - Field TCAFCTR (TCAFCRC in ANS COBOL) contains X'08'. Application programs should test this field after requesting DL/I services, by specifying the INVREQ parameter of the DFHFC TYPE=CHECK macro, and further processing should be based on the result.

DLZ050I DL/I ONLINE NUCLEUS MODULE NOT FOUND

Explanation: During DL/I online initialization the optional features field for DL/I in the CICS CSA was zero.

Action: Respond to message DLZ040A as follows:
1. CANCEL or DUMP - Ensure that the module as specified in the CICS SIT is cataloged in the DOS/VS Core Image Library and reinitialize CICS.
2. GO - Field TCAFCTR (TCAFCRC in ANS COBOL) contains X'08'. Application programs should test this field after requesting DL/I services, by specifying the INVREQ parameter of the DFHFC TYPE=CHECK macro, and further processing should be based on the result.

DLZ051I ONE OR MORE APPLICATION MODULES NOT FOUND IN CICS PPT

Explanation: During DL/I online initialization, one or more of the programs specified in the online nucleus generation ACT statement could not be found in the CICS PPT. The program name has been logged by CICS.

Action: Respond to message DLZ040A as follows:
1. DUMP or CANCEL - Regenerate a CICS PPT with the appropriate entries prior to reinitializing DL/I online.
2. GO - The affected application program(s) may not be executed.

DLZ052I DL/I INITIALIZATION FAILED

Explanation: DL/I online initialization encountered an error which required the DL/I dummy program to be invoked. Field TCAFCTR (TCAFCRC in ANS COBOL) contains X'08'. Application programs should test this field after requesting DL/I services, by specifying the INVREQ parameter of the DFHFC TYPE=CHECK macro, and further processing should be based on the result.

Action: Refer to explanation of the previous error message printed.

DLZ053I DL/I INITIALIZATION COMPLETE

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

Action: None required.

DLZ054I DL/I INITIALIZATION ERROR DETECTED

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

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

DLZ055I DL/I ACTION MODULE NOT FOUND BY LOCATE MODNAME=modname

Explanation: DL/I online initialization could not locate the named DL/I action module in a DOS/VS core image library.

Action: Respond to DLZ040A as follows:
1. CANCEL or DUMP - Link-edit the required DL/I modules to a DOS/VS core image library and reinitialize CICS.
2. GO - Field TCAFCTR (TCAFCRC in ANS COBOL) contains X'08'. Application programs should test this field after requesting DL/I services, by specifying the INVREQ parameter of the DFHFC TYPE=CHECK macro, and further processing should be based on the result.

DLZ056I PSB FAILED TO INITIALIZE PSBNAME=psbname

Explanation: During DL/I online initialization, the named PSB was not successfully initialized.

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

DLZ057I DBD FAILED TO OPEN DBDNAME-dbdname

Explanation: During DL/I online initialization, an open error occurred for a data set referenced by the named DBD. The error is indicated by a preceding open error message in the range DLZ020 to DLZ028.

Action: Respond to message DLZ040A as follows:

1. DUMP or CANCEL - Perform the appropriate action as indicated by the open error message prior to reinitializing DL/I online.
2. GO - Application programs referencing PSBs that rely on the named DBD will not be scheduled.

DLZ058I INSUFFICIENT STORAGE TO INITIALIZE DL/I

Explanation: During DL/I online or batch initialization, the virtual address area allocation was too small to contain the DL/I modules or buffer pool.

Action: For online initialization, respond to message DLZ040A as follows:

1. DUMP or CANCEL - Increase the SIZE parameter in the EXEC statement prior to reinitialization of CICS/VS.
2. GO - Field TCAFCTR (TCAFCRC in ANS COBOL) contains X'08'. Application programs should test this field after requesting DL/I services, by specifying the INVREQ parameter of the DFHFC TYPE=CHECK macro, and further processing should be based on the result.

For batch initialization, this message is followed by an immediate abnormal termination and dump. You can usually resolve the problem by increasing the SIZE parameter in the EXEC statement and resubmitting the job. Other possible actions are to:

1. Check for proper preparation of the buffer parameters in the DLZACT TYPE=BUFFER statement. It is possible for an out of space condition to occur if an HDBFR parameter is used to decrease the number of buffers, but the parameter is not read because of a missing continuation punch in column 72.
2. Ensure that the DL/I action modules have been loaded in the SVA. It is possible to run out of space for the prereorganization, scan, or prefix update utility because the action modules were loaded in the partition instead of in SVA.

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.

Action: Respond to message DLZ061A as specified in that message.

DLZ061A ENTER POOL COUNT, GO, CANCEL, OR DUMP

Explanation: This message follows message DLZ060I.

Action: Respond with one of the following entries:

- pool count — Continues normal DL/I online initialization. You must enter the number of subpools (1 to 255) desired.
- 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 nonzero buffer pool count and reinitialize CICS.
- DUMP — Produces a dump and terminates DL/I online initialization. You must reassemble an ACT with a nonzero buffer pool count and reinitialize CICS.

DLZ062I DL/I HAS BEEN ABNORMALLY TERMINATED

Explanation: A condition has occurred that makes it impossible for DL/I to continue. All active DL/I tasks have been abnormally terminated with code D062, and no further scheduling of DL/I services will be performed. Field TCAFCTR (TCAFCRC in ANS COBOL) contains X'08'. Application programs should test this field after requesting DL/I services, by specifying the INVREQ parameter of the DFHFC TYPE=CHECK macro, and further processing should be based on the result.

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

DLZ063I DL/I NORMAL TERMINATION ENTERED

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

Action: None required.

DLZ064I DL/I NUCLEUS REJECTED

Explanation: An attempt was made to initialize the current level of DL/I with an incompatible level DL/I nucleus.

Action: Regenerate the DL/I nucleus using the DLZACT macro that is provided with your current level DL/I system.

DLZ065I ACTIVE DL/I TASKS .. DL/I ABNORMALLY TERMINATED

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

Action: 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.

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

DLZ067I INSUFFICIENT STORAGE TO TERMINATE DL/I

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

Action: 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.

Action: 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.

Action: None required.

DLZ070I ABNORMAL TERMINATION COMPLETE

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

Action: 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.

DLZ071I PSB VERSION INVALID. PSBNAME=psbname

Explanation: While loading the named PSB, online initialization detected that the block was created prior to V1.1.

Action: Respond to message DLZ040A as follows:
1. DUMP or CANCEL - Rebuild the named PSB using the ACB utility provided with the current level of DL/I DOS/VS prior to initializing DL/I online.
2. GO - Application programs referencing the named PSB will not be scheduled.

DLZ072I DBD VERSION INVALID. DBDNAME=dbdname

Explanation: While loading the named DBD, online initialization detected that the block was created prior to V1.1.

Action: Respond to message DLZ040A as follows:
1. DUMP or CANCEL - Rebuild the named DBD using the ACB utility provided with the current level of DL/I DOS/VS prior to initializing DL/I online.
2. GO - Application programs attempting to schedule PSBs that reference the named DBD will not be scheduled.

DLZ073I COMPRESSION MODULE NOT FOUND. NAME=name

Explanation: While initializing the DL/I control blocks the named user routine was not located in a DOS/VS core image library.

Action: Respond to message DLZ040A as follows:
1. DUMP or CANCEL - Catalog the named module into a DOS/VS core image library prior to initializing DL/I online.
2. GO - Application programs attempting to schedule PSBs that depend on the named module will not be scheduled.

DLZ074I INDEXING MODULE NOT FOUND. NAME=name

Explanation: While initializing the DL/I control blocks the named user routine was not located in a DOS/VS core image library.

Action: Respond to message DLZ040A as follows:
1. DUMP or CANCEL - Catalog the named module into a DOS/VS core image library prior to initializing DL/I online.
2. GO - Application programs attempting to schedule PSBs that depend on the named module will not be scheduled.

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.)

Action: Correct LOG parameter values and resubmit job.

DLZ076A DL/I 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.

Action: 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.

DLZ077I 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.

Action: 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.

DLZ078I 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 cancelled.

Action: 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.

DLZ079I LOG FULL - RUN CANCELED 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.

Action: 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.

DLZ080I 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.

Action: None required.

DLZ081I MPS BATCH DL/I PARTITION STARTED

Explanation: The MPC (master partition controller) was activated (XPOST macro) because a batch partition started executing DL/I using MPS (multiple partition support).

Action: None required.

DLZ082I [id] routname {DEFINE|DELETE|CHECK} UNSUCCESSFUL. XECB=xecbname. RETURN CODE=rc

Explanation: Return code rc was received when the routname routine issued an XECBTAB macro for use in DL/I MPS (multiple partition support). 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 - 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 will send a message when it terminates

Action: The batch partition ends abnormally.

If the return code (rc) is:

00 on a CHECK, the execution of the XECBTAB macro was successful, but the address returned was not the address expected.

04 for a DEFINE, the XECB was previously defined; probably by a program other than DL/I. Other programs should not use XECB names DLZXCBnm, where n is 0-5 and m is 0-3.

04 for a CHECK or DELETE, the XECB was unexpectedly deleted. Other programs should not use XECB names DLZXCBnm, where n is 0-5 and m is 0-3. 04 for a CHECK could also mean the master partition controller (MPC) terminated abnormally.

08 for a DEFINE, a larger XECB table must be generated because it is full.

DLZ083I MPS CANNOT BE STARTED. MPC ATTACH FAILED

Explanation: Multiple partition support cannot be started because the MPC (master partition controller) could not be attached due to insufficient CICS/VS storage.

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

DLZ084I [id] routname {XPOST|XWAIT} UNSUCCESSFUL. XECB=xecbname. RETURN CODE=rc

Explanation: Return code rc was received when the routname routine issued an XPOST or XWAIT macro for use in DL/I MPS (multiple partition support). 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 - Initialization
DLZMPRH - Program request handler
DLZODP01 - Task termination routine

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

If routname is DLZBPC00 or DLZODP01, the partition identifier (id) of the batch partition that the BPC is serving appears in the message text.

Action: The batch partition ends unless the routine is DLZBPC00 , DLZMPC00 or DLZODP01 for XPOST. If there is no other message concerning the ending of a batch partition, the operator will have to cancel it.

If the return code is 04, the XECB table did not contain the expected XECB name. It either, was never defined because MPS was not started, or it was unexpectedly deleted (probably by a program other than DL/I or because the master partition controller (MPC) terminated abnormally). Other programs should not use XECB names DLZXCBnm, where n is 0-5 and m is 0-3.

DLZ085I MPS BATCH PARTITION ENDED ABNORMALLY. BPC ATTACH FAILED

Explanation: The MPS batch partition ended abnormally because the MPC (master partition controller) could not attach the BPC (batch partition controller) that was to service it.

Action: 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 REQUESTED

Explanation: Multiple partition support was signaled to stop and will terminate after all currently active batch partitions are complete. MPS will not allow any more batch partitions to start.

Action: 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.
- The program name was DLZBACK0 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.

Action: 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 CANNOT START WITH ONLY ONE PARTITION DEFINED.

Explanation: Multiple partition support (MPS) cannot run because only one partition was defined at system generation (SYSGEN). MPS requires that at least two DOS/VS partitions be defined; one for CICS/VS and one batch partition.

Action: MPS ends because only one partition was defined at system generation.

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

Explanation: The MPS (multiple partition support) batch partition is unable to start because it could not find a MPC (master partition controller) XECB that is named DLZXCBn0 or DLZXCBn3, where n is 1-5. The XECB could not be found because the MPC was not started, was ending abnormally, was signaled to stop by the stop transaction module, or a program other than DL/I deleted the MPC's XECB.

Action: If MPC was not started, start the MPC and resubmit the batch job. If MPS was attempting to stop, or was ending abnormally, do not submit the batch job until MPS is restarted. Otherwise ensure that non-DL/I programs do not use XECB names DLZXCBn0 and DLZXCBn3, where n is 1-5.

DLZ090I ABNORMAL {XPOST|XWAIT} CONDITION DURING BATCH PARTITION TERMINATION

Explanation: During batch partition termination, an XPOST or XWAIT macro that was issued to the online partition was unsuccessful. If XPOST appears in the message text no TERM record is written. If XWAIT appears in the message text, check the online messages to determine if a TERM record could be written.

Action: Check online partition messages; if there is no TERM record, take the appropriate recovery action.

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.

Action: Make sure that no more than eighteen parameters are used on a CALL statement to DL/I. If less than eighteen parameters were used, save the log and dump, if printed, for your IBM Programming System Representative.

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.

Action: Make sure that all addresses are within the partition. If they are all valid, save the log and dump, if printed, for your IBM Programming Systems Representative.

DLZ093I MPS STARTED

Explanation: The MPC (master partition controller) for DL/I MPS (multiple partition support) started successfully. Batch jobs for MPS can now be started.

Action: None required.

DLZ094I MPS STOPPED {ABNORMALLY|NORMALLY}

Explanation: MPS (multiple partition support) 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.

Action: None required.

DLZ095I MPS BATCH PARTITION ENDED. SCHEDULING RETURN CODE=yyzz

Explanation: The MPS (multiple partition support) 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 TCAFCRC if ANS COBOL) and zz is the contents of TCADLTR. The *DL/I DOS/VS Application Programming Reference Manual,* contains the meanings of the return codes.

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

DLZ096I MPS BATCH PARTITION ENDED ABNORMALLY

Explanation: The MPS batch partition was entered at either the STXIT AB or STXIT PC entry point in the batch abend handler routine (DLZMABND).

Action: Examine the log and dump, if printed. If the problem was not caused by the application program, notify the IBM Programming Systems Representative.

DLZ097I MPS START ATTEMPT WHEN DL/I NOT ACTIVE

Explanation: There was an attempt to start MPS (multiple partition support) 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.

Action: 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 the core image library.

DLZ098I RETURN CODE cde FROM DL/I CALL

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

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

DLZ099I MPS BATCH PARTITION ENDED ABNORMALLY

Explanation: When the batch partition regained control after notifying the online partition that it was starting, the XECB of the online partition could not be found to determine if the start was successful. The batch partition is ended.

Action: Resubmit the job after taking action as specified in the previous message (DLZ082I).

DLZ100I MPS BATCH PARTITION ENDED ABNORMALLY. CODE=xx

Explanation: If CODE=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.

If CODE=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.

Action: If the code is:

01 - Check online messages to determine if the BPC gave a reason for terminating. Save the log and dump for your IBM Programming Systems Representative.

02 - If the online partition terminated abnormally, correct the problem and take any recovery action needed. Otherwise, save the log and dump for your IBM Programming Systems Representative.

DLZ101I MPS START ATTEMPTED WHEN MPS ACTIVE - IGNORED

Explanation: MPS (multiple partition support) start transaction processing was initiated when MPS was active or in the process of stopping. It has no effect.

Action: None required.

DLZ102I BAD RETURN FROM ONLINE PARTITION. RETURN CODE=yyzz

Explanation: The MPS (multiple partition support) 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 TCAFCRC if ANS COBOL) and zz is the contents of TCADLTR. The *DL/I DOS/VS Application Programming Reference Manual,* contains the meanings of the return codes.

Action: 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 cancelled.

Action: 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. If the completion code is ASRA, program interrupt PSW information also appears in the message text. The PSW is from TCAPCPSW.

Action: Restart MPS after correcting the cause for the MPC or BPC abending and taking appropriate recovery action.

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

Explanation: A program has taken a checkpoint.

Action: 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.

Action: Rerun the task.

DLZ108I TASK TERMINATION DUE TO INSUFFICIENT CICS PARTITION SPACE

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

Action: Rerun the task. If this condition persists, an increase in the size of the CICS/VS partition, a decrease in the number of concurrently executing tasks, or a rewrite of tasks to issue checkpoint calls (CHKP) or CICS/VS SYNCH points may be necessary.

DLZ120I DL/I TRACE DISABLED - ERROR NUMBER = n

Explanation: The tracing facility has discovered an error while trying to write the income table to the CICS extrapartition dataset. Tracing is disabled and all trace entries from this point on are lost. The task continues to run, but without tracing.

n=1 The ID (DDIT) used for the output extrapartition dataset can not be located by CICS.

n=2 An I/O error occurred on the trace output device used by CICS.

n=3 There is no more space available on the trace output file.

n=4 The trace output file is not open.

Action: Correct the error condition on the CICS extrpartition dataset device and if necessary rerun the job.

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

DLZ260I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

Explanation: Number of parameters (data items named in USING list) in the application program CALL to DL/I exceeds the allowable limit.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ261I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

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

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ262I DL/I SYSTEM TERMINATED ABNORMALLY

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

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ263I DL/I SYSTEM TERMINATED ABNORMALLY

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

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ264I DL/I SYSTEM TERMINATED ABNORMALLY

Explanation: An I/O error occurred while writing on the journaling tape. The DL/I system is terminated.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ266I DL/I SYSTEM TERMINATED ABNORMALLY

Explanation: Secondary list code is invalid.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ267I DL/I SYSTEM TERMINATED ABNORMALLY

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

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ268I DL/I SYSTEM TERMINATED ABNORMALLY

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.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ301I ERROR - UNABLE TO OPEN FILENAME - filename [VSAM EC=xx]

Explanation: The issuing utility program was unable to open a data set with the filename specified in the message. Additionally, the VSAM error code is printed as a hexadecimal value for VSAM files (DL/I data bases).

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

DLZ302I ERROR - CARD COLUMN 1 NOT {S|D|R|L}

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

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

DLZ303I ERROR - CARD COLUMN 2 NOT A 1 OR 2

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

Action: Correct control card, and resubmit the job.

DLZ304I ERROR - NO DBD NAME SUPPLIED

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

Action: Correct control card and resubmit the job.

DLZ305I ERROR - DBD NAME NOT FOUND IN CORE IMAGE LIBRARY

Explanation: A utility program attempted to find a DBD or DMB with the name specified in the utility control card in the assigned DOS/VS core image libraries. The specified DBD or DMB was not found.

Action: Correct the utility control statement and resubmit the job.

DLZ306I ERROR - FILENAME SUPPLIED 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.

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

DLZ307I ERROR - NO DATA SET FILENAME SUPPLIED

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

Action: 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 filename in card column 22 and the field was blank.

Action: Correct the conrol card, and resubmit job.

DLZ309I ERROR - CONTROL CARD ERRORS DETECTED

Explanation: A utility program found one or more control card errors. (The particular errors have been described by messages preceding this one.)

Action: Examine the output listing, correct the errors, and resubmit the job.

DLZ310I CARD IMAGE FOLLOWS - card image

Explanation: This message usually follows DLZ309I. The control card in error is printed as the second line to identify the particular card in error.

Action: Correct any errors, and resubmit the job.

DLZ311I filename FILE NOT ASSIGNED OR DEVICE INVALID

Explanation: The specified data set was either not assigned or was assigned to an invalid device type. The filename may be incorrect. If filename = TRACEIN, the input file to the Trace Print Utility is involved. The real file name in this case is TAPEIN or DISKIN.

Action: Correct the control card and/or JCL and resubmit the job.

DLZ312I ERROR - DUMP SUPPLIED NOT A DUMPED OR REORGANIZED DATA BASE

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.

Action: Correct the input and resubmit the job.

DLZ313I ERROR - DUMP SUPPLIED NOT FOR DBD SUPPLIED

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.

Action: Correct the input and resubmit the job.

DLZ314I ERROR - CUM 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.

Action: Correct the JCL statements and resubmit 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 number nnnn is the last checkpoint record on the unloaded file.

Action: Resubmit the job and enter the correct restart number when responding to message DLZ318A.

DLZ316I ERROR - DATA BASE ORGANIZATION UNRECOGNIZABLE

Explanation: The DMB for the specified DBD did not contain a valid data base organization code.

Action: 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 DOS/VS core image library. Correct any discrepancies 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 filename specified.

Action: 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 in the format nnnn, where nnnn may be from 1 to 9999.

Action: 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.

DLZ319I ERROR - I/O ERROR ON FILENAME - filename [{PHYSICAL|LOGICAL} RC=xx|ESDS EMPTY]

Explanation: A utility program encountered an unrecoverable I/O error on the named file. If the error was caused by a read or write to a VSAM file and a logical or physical I/O error occurred, the return code (RC) will be printed as a hexadecimal value. If the error occurred on an output volume, and multiple output copies were specified, the program will continue with alternate copy; if multiple output copies were not specified, the program will terminate. If ESDS EMPTY appears in the message text, this message is issued only to warn you of the condition. Processing continues.

Action: Resubmit the job and, if the error persists, call an IBM Programming Systems Representative. If the error is a logical error this may be a DL/I utility error and an APAR should be submitted.

DLZ320I 2 COPIES SPECIFIED - WILL CONTINUE WITH ONE

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.

Action: None required.

DLZ321I ERROR - OPERATION TERMINATED - WILL TRY NEXT CONTROL CARD

Explanation: A utility program encountered errors prohibiting the successful completion of the requested operation. The program attempts to obtain another control card.

Action: None required.

DLZ322I WARNING - NO DUMP INPUT SUPPLIED

Explanation: The data base data set recovery utility found the image copy input file was not assigned.

Action: If none was required, no action is necessary. Otherwise, correct the input and resubmit.

DLZ323I WARNING - NO LOG INPUT SUPPLIED

Explanation: The data base data set recovery utility found that the log input file was not assigned.

Action: If none was required, no action is necessary. Otherwise, correct the input and resubmit.

DLZ324I WARNING - NO CUM INPUT SUPPLIED

Explanation: The data base data set recovery utility found that the accumulated change input file was not assigned.

Action: If none was required, no action is necessary. Otherwise, correct the input and resubmit.

DLZ325I WARNING - CUM PURGE DATE LATER THAN DUMP. MISSING RECORDS POSSIBLE.

Explanation: The data base data set recovery utility found that the purge date specified when the accumulated change input file was created, was later than the date the dump input file was created.

Action: 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.

DLZ326I WARNING - CUM RUN DATE EARLIER THAN DUMP. CUM INPUT IGNORED.

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.

Action: None required.

DLZ327I WARNING - NO RECORDS ON CUM FOR RECOVERED DATA SET

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.

Action: Ensure that the accumulated change input was correct. If not, correct the input and resubmit the job.

DLZ328I WARNING - NO RECORDS ON LOG FOR RECOVERED DATA SET

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.

Action: Ensure that all system log files were submitted. If any were left out, correct the JCL statements, and resubmit the job.

DLZ330I ERROR - CUM RECORD REL BLK NUMBERS NOT IN ASCENDING SEQ

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.

Action: Recreate the accumulated change input file and resubmit the job.

DLZ331I ERROR - CUM RECORD KEYS NOT IN ASCENDING SEQ

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.

Action: Recreate the accumulated change input file and resubmit the job.

DLZ332I ERROR - ESDS RECORD TO BE ADDED IS MORE THAN 1 GREATER THAN LAST RECORD

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.

Action: Ensure that all supplied input files are correct and resubmit the job.

DLZ333I ERROR - DUPLICATE KEY ON LOG AND INSERT

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.

Action: Ensure that all input files are correct, all log tapes are in the proper sequence, and resubmit the job.

DLZ334I ERROR - LOG DATES INDICATE LOGS NOT IN SEQ

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.

Action: If multiple logs were submitted as input, resubmit the job, and ensure the log tapes are submitted in ascending sequence.

DLZ335I ERROR - NO DUMP OR LOG INPUT SUPPLIED

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.

Action: Ensure that the correct input files are assigned and resubmit the job.

DLZ336I ERROR - PERM READ ERROR IN RECOVERED DATA SET

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

Action: Initialize the DASD volume, and resubmit the job. If the error persists, call an IBM Programming Systems Representative.

DLZ337I ERROR - INSUFFICIENT SPACE IN BUFFER POOL

Explanation: The data base data set recovery utility requested buffer pool space, but none was available. This is a DL/I logic error.

Action: Ensure that the DL/I system and all control statements are valid and resubmit the job. If the error persists, call an IBM Programming Systems Representative who will submit an APAR.

DLZ338I ERROR - INSUFFICIENT SPACE IN DATA SET

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.

Action: 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.

DLZ339I NO ERRORS DETECTED - DATA BASE RECOVERY SUCCESSFUL

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

Action: None required.

DLZ340I ERRORS DETECTED - DATA BASE RECOVERY UNSUCCESSFUL

Explanation: The data base data set recovery utility has encountered errors while attempting to recover a file.

Action: Examine error messages, make necessary corrections, and resubmit the job.

DLZ341I ERROR - CATASTROPHIC ERROR IN DL/I

Explanation: The data base data set recovery utility requested DL/I to perform an operation on the file being recovered. The return code indicates the task can not be performed.

Action: Ensure that the DL/I system and all control statements are valid and resubmit the job. If the error persists, call an IBM Programming Systems Representative who will submit an APAR.

DLZ342I NUMBER OF LOG INPUTS INVALID

Explanation: The number of log files in the utility control statement is invalid.

Action: Correct the utility control statement or provide the correct assignment and resubmit the job.

DLZ343I ERROR - CUM SPECIFIED AND SHISAM RECOVERY

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.

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

DLZ345I WARNING - filename NOT ASSIGNED. 1 COPY CREATED.

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

Action: If a second copy is required, correct JCL 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.

Action: Ensure output units and volumes are in good working order, and resubmit the job. If the error persists, call an IBM customer engineer.

DLZ347I ERRORS DETECTED - DATA BASE UNLOAD UNSUCCESSFUL

Explanation: Errors as described in preceding messages have precluded the successful unload of a data base.

Action: Examine listing, correct errors, and resubmit the job.

DLZ348I ERROR - UNEXP RET STATUS FROM CALL. STAT WAS xx

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

Action: Refer to Chapter 5 of this manual for the meaning of the returned status. If this indicates the error, correct and resubmit the job. If not, call an IBM Programming Systems Representative who will submit an APAR.

DLZ349I ERROR - PERM READ ERROR ON FILENAME - filename

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

Action: Resubmit the job. If the error persists, execute the data base data set recovery utility to recover the named file. If the error still persists, call an IBM Programming Systems Representative.

DLZ350I ERROR - A ZEROED CI WAS FOUND

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

Action: If the zeroed CI does not make sense, reorganize the data base to remove the zeroed CI.

DLZ352I NO ERRORS DETECTED - DATA BASE UNLOAD SUCCESSFUL

Explanation: The HD reorganization unload utility has successfully unloaded the data base.

Action: No action required.

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.

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

DLZ354I NO ERRORS DETECTED. DATA BASE RELOAD SUCCESSFUL

Explanation: The HD reorganization reload utility has successfully reloaded the data base.

Action: None required.

DLZ355I ERRORS DETECTED. DATA BASE RELOAD UNSUCCESSFUL

Explanation: Errors as described in preceding messages have precluded the successful reloading of the data base.

Action: Correct the errors, and resubmit the job.

DLZ356I ERROR - NO HEADER FOUND ON FILENAME - filename.

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

Action: 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 unit was assigned; or one copy and two output units were assigned.

Action: 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.

Action: 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.

DLZ360I ERROR - ID CARD ERROR AT OR NEAR COLUMN -

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

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

Action: Correct the control card, and resubmit the job.

DLZ361I ERROR - DB CARD ERROR ENCOUNTERED

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

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

Action: Correct the control card, and resubmit the job.

DLZ362I CARD IN ERROR FOLLOWS -

Explanation: The control card in error is printed as part of the message text.

Action: Correct the card in error, 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 may be used.

Action: Correct control cards, and resubmit the job.

DLZ364I ERROR - A DUPLICATE DBD NAME HAS BEEN ENCOUNTERED

Explanation: The user has specified the same DBD name on more than one control card.

Action: Correct the control card, and resubmit the job.

DLZ365I ERROR - DATE/TIME TABLE HAS OVERFLOWED

Explanation: The accumulation change utility has encountered more DBD names than were specified.

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

DLZ366I ERROR - DATA BASE NAME TABLE HAS OVERFLOWED

Explanation: The accumulation change utility has encountered more data base names than were specified.

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

DLZ367I ERROR - *OTHER SPECIFICATION INVALID.

Explanation: The accumulation change 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.

Action: Correct the control cards, and resubmit the job.

DLZ369I ERROR - DMB NOT FOUND IN CORE IMAGE LIBRARY. 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.

Action: Ensure that the control statements are correct, and that all DMBs for the corresponding DBDs reside in an assigned core image library. 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 the first checkpoint record on the unloaded file.

Action: 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.

Action: 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 32,767
- The value was less than 1024

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

DLZ373I ERROR - FILENAME - CUMIN NOT A VALID INPUT ACCUMULATED DATA SET

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.

Action: 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.

Action: 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.

Action: Ensure that all inputs are valid, and resubmit the job. If the error persists, call an IBM Programming Systems Representative who will submit an APAR.

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.

Action: 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.

Action: 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 CHECKPOINT RESTART SUCCESSFULLY COMPLETED

Explanation: Self-explanatory.

Action: None required.

DLZ379I ERRORS DETECTED - CHECKPOINT RESTART UNSUCCESSFUL

Explanation: The checkpoint restart attempt was unsuccessful because of conditions described by the previous messages printed.

Action: If the error encountered can be corrected, correct the error and resubmit the job. Otherwise, restart the job from the beginning.

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.

Action: Ensure all DOS/VS job control cards 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.

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

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 decimal value between 1 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 segname field is the segment name of the segment being retrieved when the checkpoint was taken.

Action: 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 the DBD was not a valid DBD.

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

DLZ383I ERROR - INPUT RECORDS NOT IN SEQ

Explanation: A root segment has been encountered with a sequence field (key) equal to or lower than the previous root segment processed.

Action: The execution is terminated, and a storage dump was produced. Register 11 is the address of the current key and register 9 is the address of the previous key.

DLZ384I NO ERRORS DETECTED - REQUESTED OPERATION SUCCESSFUL

Explanation: Self-explanatory.

Action: No action required.

DLZ385I ERRORS DETECTED - REQUESTED OPERATION UNSUCCESSFUL

Explanation: Errors have been detected which have prohibited the successful conclusion of the requested operation(s).

Action: If other error messages have preceded this message, correct the indicated errors, and resubmit the job. If this message appears alone, it indicates sort/merge returned with a nonzero return code. Examine sort messages to determine the cause of the error. Correct the errors, 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.

Action: Correct the invalid input, and resubmit the job.

DLZ389I ERROR - STATISTICS RECORD NOT FOUND

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

Action: 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).

Action: Correct log tape input and resubmit the job.

DLZ391I ERROR OCCURRED EXECUTING MACRO {GETVIS|FREEVIS}, RC=n

Explanation: The utility program encountered an error executing either a GETVIS or FREEVIS macro request. The program terminated. A storage dump is produced if a FREEVIS error. The macro return code is indicated as RC=n.

Action: If GETVIS, correct the SIZE parameter in the EXEC statement and reexecute the program. If FREEVIS, call an IBM Programming Systems Representative who will submit an APAR.

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.

Action: Call an IBM Programming Systems Representative who will submit an APAR.

DLZ393I VSAM DEFINE INCOMPATABLE FOR FILENAME - filename

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 - KSDS or ESDS
- control interval size - KSDS or ESDS
- key length - KSDS
- relative key position - KSDS.

Action: 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.

Action: This is a DL/I logic error. Call an IBM Programming Systems Representative who will submit an APAR.

DLZ395I BACKOUT COMPLETE FOR PSB psbname [TO CHKPT id]

Explanation: The data base backout utility has successfully completed the requested data base backout. To CHKPT id identifies the checkpoint to which the backout was completed.

Action: Record the checkpoint-id and the volume serial number in case restart processing must be done.

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.

Action: 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.

Action: Check the DOS/VS job control cards to ensure they are present and identify the data sets properly. Correct any errors and rerun the job.

DLZ398I I/O ERROR DURING BACKOUT IN DATA BASE dbname PROGRAM name.

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

Action: Run the data base recovery utility for the data base in error. Use all log tapes including the output of this backout attempt. 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.

Action: Ensure that the DL/I system and all control statements are valid and resubmit the job. If the error persists, call an IBM Programming Systems Representative who will submit an APAR.

DLZ400I SEQUENCE ERROR IN DATA BASE

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

Action: 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.

Action: 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.

Action: Correct the utility control statement and/or JCL and resubmit the job.

DLZ404I ERROR - UNABLE TO OPEN FILENAME - filename VSAM CODE=xxyy

Explanation: A DL/I utility was unable to open the VSAM data set with the filename specified in the message. xx represents the VSAM hexadecimal return code and yy and VSAM hexadecimal error code.

Action: Ensure that all utility control statements and JCL are correct and resubmit the job.

DLZ405I ERROR - UNABLE TO CLOSE FILENAME - filename VSAM CODE=xxyy

Explanation: A DL/I utility was unable to close the VSAM data set with the filename specified in the message. xx represents the VSAM hexadecimal return code and yy the VSAM hexadecimal error code.

Action: Take the appropriate action for the VSAM code and resubmit the job.

DLZ406I ERROR - I/O ERROR ON FILENAME - filename [VSAM CODE=xxyy]

Explanation: A DL/I utility encountered an I/O error on the data set with the filename specified in the message. If it was a VSAM data set, xx represents the VSAM hexadecimal return code and yy the VSAM hexadecimal error code.

Action: Take the appropriate action for the type of error indicated and resubmit the job. If the error persists, call an IBM Programming Systems Representative.

DLZ407I ERROR - INVALID INPUT CONTROL STATEMENT

Explanation: The input control statement supplied to the utility did not begin with valid characters in column 1.

Action: Correct the utility control statement and resubmit the job.

DLZ408I ERROR - DUPLICATE LO RECORDS

Explanation: More than one LO control statement was encountered by the log print utility.

Action: Remove the extra LO control statement(s) and resubmit the job.

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 alphameric 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.

Action: 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.

Action: 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.

Action: 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.

Action: 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.

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

Action: 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, or S.
- Column 22 was not blank.

Action: 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.

Action: 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 - filename

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

Action: None.

DLZ418I TOTAL FILE RECORDS READ=nnnnnnn, RECORDS PRINTED=nnnnnnn

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.

Action: None.

DLZ419I NO ERRORS DETECTED - LOG PRINT SUCCESSFUL

Explanation: The log print utility completed processing successfully.

Action: None.

DLZ420I TOTAL JOB RECORDS READ=nnnnnnn, RECORDS PRINTED=nnnnnnn

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.

Action: None.

DLZ421I END DATE EXCEEDED ON FILENAME - filename

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.

Action: None.

DLZ422I ERRORS DETECTED - LOG PRINT UNSUCCESSFUL

Explanation: The log print utility was unsuccessful because of conditions described by the previous messages printed.

Action: Correct the errors 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.

Action: 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.

Action: None.

DLZ425I PSBNAME=aaaaaaaa, SEQ NO=xxxxxxxx

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

Action: 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.

Action: 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.

Action: None.

DLZ428I ERROR ON FILENAME - filename, LOG COPY HALTED

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

Action: 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 user requested the copy function by specifying COPY on the LO input control statement. The new log tape file may be used as input to the backout utility.

Action: None.

DLZ430I ERROR - INVALID CICS DUMP PARAMETER

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

Action: Correct the parameter and resubmit the job.

DLZ431I ERROR - DUPLICATE LS RECORDS

Explanation: More than one LS control statement was encountered by the log print utility.

Action: Remove the extra LS control statements 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.

Action: Correct the DBD name and resubmit the job.

DLZ433I ERROR - INVALID TASK ID

Explanation: The CICS task ids 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.

DLZ434I ERROR - INVALID RBN

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 through 9, A through 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.

Action: Correct the RBN and resubmit the job.

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

Action: Correct error and resubmit job.

DLZ441I ERRORS DETECTED BY TRACE PRINT UTILITY

Explanation: One or more errors were detected by the trace print utility.

Action: If the errors prevented proper execution of the trace print utility, correct them and resubmit 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.

Action: 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.

Action: Correct problem with device or assign a different device and rerun job.

DLZ444I TRACE PRINT COMPLETED SUCCESSFULLY

Explanation: The trace print utility finished printing the input file, freed all acquired GETVIS storage, and was able to close all files.

Action: None.

DLZ445I END OF FILE ON TRACE INPUT

Explanation: An End of File condition was returned while reading the trace input.

Action: None.

DLZ476I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

Explanation: An invalid or no PCB address in a DL/I call. Possible cause is that the PSB language specified was not the same as the program language.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ570I INVALID SECONDARY LIST TYPE WAS FOUND IN DBD dbdname

Explanation: This is an internal system error.

Action: 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 an IBM Programming Systems Representative who will submit an APAR.

DLZ571I *WARNING DUPLICATE INPUT PSBNAME --psbname--IGNORED

Explanation: A duplicate PSB name was specified in the PSB operand of the BUILD card.

Action: The second occurrence of the specified PSB name is ignored.

DLZ572I *WARNING* DMB 'dmbname' EXISTS IN CORE IMAGE LIBRARY

Explanation: DMB=YES was coded on the BUILD statement indicating that DMBs were to be unconditionally generated. The DMB is already present in the DOS/VOS core image library and will be overlayed if the output is linkage edited and cataloged.

DLZ573I *WARNING* DMB 'dmbname' DOES NOT EXIST IN CORE IMAGE LIBRARY

Explanation: DMB=NO was coded on the BUILD statement, inhibiting generation of DMBs. The named DMB is not present in the DOS/VOS core image library and is required to successfully use at least one of the generated PSBs.

DLZ583I SYSPCH NOT ASSIGNED

Explanation: Output was designated to the SYSPCH device and SYSPCH was assigned IGN or UA.

Action: 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.

Action: 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.

Action: Correct the errors and reexecute the program.

DLZ587I ERROR BUILDING PSB=psbname IT WILL NOT BE BUILT

Explanation: The block builder returned a nonzero completion code after processing the indicated PSB. Processing continues for any other named PSBs.

Action: 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 LABEL } [symbol]
{OP CODE }
{BLK TYPE }
{OPERAND }
{FORMAT }
{CONTINUATION }

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.

Action: 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.

Action: None required.

DLZ772I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

Explanation: Error in secondary list of an index relationship encountered by DLZDXMT0.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ796I DL/I {SYSTEM|TASK} TERMINATED ABNORNALLY

Explanation: An invalid RBA pointer condition was detected in the process of deleting a segment.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ797I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

Explanation: An invalid RBA pointer condition was detected in the process of inserting a segment.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLA798I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

Explanation: An invalid RBA pointer condition was detected in the process of retrieving a segment.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ799I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

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.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ800I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

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.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ801I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

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 pseudo abend save area.

Name	Description
PSTFNCTN	Requested function
PSTRTCDE	Return code

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ802I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

Explanation: While processing a replace call for a variable-length segment, space management (DLZDHDS0) indicated to DLZDLD00 that no space is available on the data base.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ803I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

Explanation: Invalid pointer conditions were found while deleting a logically related segment. The logical parent or child could not be found or the counter in the prefix of the logical parent was negative.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ804I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

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.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ806I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

Explanation: Cannot access or free work area; or an error occurred attempting to mark buffer altered.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ807I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

Explanation: Delete/replace cannot obtain position for work area (physical path to root required).

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ808I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

Explanation: The LEVTTR field does not contain a valid value for DLZDLD00.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ830I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

Explanation: Requested space larger than block in file. Check DBD generation.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ831I DL/I SYSTEM TERMINATED ABNORMALLY

Explanation: The DL/I system terminated because it did not find an FBA device. Space Management cannot continue because it does not have the track and cylinder capacities to calculate the number of control intervals per track and cylinder to be inserted into the DMB for space management use.

Action:
Either

- Insure 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.

DLZ841I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

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.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ844I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

Explanation: No available space in file. Increase file size allocation.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ845I DL/I SYSTEM TERMINATED ABNORMALLY

Explanation: Unexpected condition in DLZDBH00.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ847I DL/I SYSTEM TERMINATED ABNORMALLY

Explanation: Unexpected return from the enqueue/dequeue routine.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ848I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

Explanation: Request for buffer space exceeds length of largest buffer (subpool buffer size).

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ850I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

Explanation: Unrecognized function in JCBPRESF.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ855I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

Explanation: RBA returned by VSAM not the same RBA value calculated by DLZDDLE0.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ860I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

Explanation: Bad return from call to data base buffer handler. The following pertinent PST fields and contents may be found in the pseudo abend save area:

Name	Description
PSTFNCTN	Requested function
PSTRTCDE	Return code.

Within save trace; find save area where R12 = entry point to DLZDDLE0.

R8 - Return within DLZDDLE0
R4 - SDB
R1 - PST
R6 - JCB.

Probable causes:

Bad insert position in SDBs from retrieve (DLZDLR00).
Data base had bad pointers.
Error in DLZDDLE0.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ861I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

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 save area where register 12 is the entry point to DLZDDLE0.

R1 = PST
R4 = SDB
R6 = JCB
R9 = INVALID SEGMENT CODE
R11 = INSERT POINT.

Probable causes:

DBD generation or VSAM access method services command DEFINE changed length of segments and data base was not reloaded.

Bad insert position in SDBs from retrieve (DLZDLR00).

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ862I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

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.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ863I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

Explanation: An error in the user's edit/compression routine resulted in one of the following:

1. The segment length is greater than the defined maximum (if variable length).
2. The segment length is smaller than specified in DMBDL for fixed length.
3. The key was changed by this routine. Registers in PSTSV3 in the DL/I ABEND save area:

R1 -	PST address.
R2 -	Segment source address in user's I/O area.
R4 -	PSDB address.
R5 -	Segment compaction table address entry.
R10 -	Segment destination address.
R12 -	Entry point DLZDDLE0.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ864I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

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.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

DLZ868I DL/I {SYSTEM|TASK} TERMINATED ABNORMALLY

Explanation: DLZDXMT0 found bad control blocks for a secondary index.

Action: See "DL/I System/Task Abnormal Termination" at the end of this chapter.

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.

Action: 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.

Action: 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 it was, call an IBM Programming Systems Representative who will submit an APAR.

DLZ901I APPLICATION ORDER OF SUBFIELDS OF FIELD field1 NOT EQUAL PHYSICAL ORDER - SEGMENT segname

Explanation: Field 'field1' in segment 'segname' 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.

Action: 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 segname, 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.

Action: Define the subfields in the application view.

DLZ903I PRIOR FIELD fldname1 START LOCATION NOT VALID FOR ALTERNATE SEQUENCE KEY FIELD fldname2 -- SEGMENT segname

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.

Action: Either redo the DBDGEN and define the field or correct the name of the field in the START parameter.

DLZ904I DMB dmbname FOUND IN CIL. IT WILL NOT BE BUILT

Explanation: The DMB named in the above message already exists in a DOS/VS core image library. This is a warning message.

Action: If the DBD has changed and should be used to create a new DMB, delete the old DMB from the DOS/VS core image library and reexecute the application control blocks utility.

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 DOS/VS EXEC statement.
2. The utility program did not execute in a virtual partition.
3. Insufficient virtual storage was available.

Action: Either allocate or increase the virtual address space and 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.

Action: Correct the PSB and reexecute the program.

DLZ907I PSB -- psbname -- REFERENCES LOGICAL DBD -- dbdname -- FOR LOAD

Explanation: The PSB named psbname contains a PCB which references the logical DBD dbdname. The PROCOPT in that PCB is LOAD. This is invalid.

Action: 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.

Action: 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; the blocks were built unless an error occurred. The segment named senseg-name in the named PSB 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.

In either case, neither ISRT nor DLET sensitivity is allowed. The processing option(s) have been changed accordingly.

Action: 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.

Action:

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.

Action: This is an internal program logic error. Remove the offending PSB, and reexecute the program. An APAR should be submitted. To do this, please call an IBM Programming Systems Representative.

DLZ912I PSB psbname REFERENCES SEGM segmname 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.

Action: 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.

Action: 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.

Action: This is an internal program logic error. Remove the offending PSB, and reexecute the program. An APAR should be submitted. To do this, please call an IBM Programming Systems Representative.

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.

Action: 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 nonunique.

Action: 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.

Action: 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.

Action: 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.

Action: 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.

Action: 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.

Action: 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.

Action: 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.

Action: Correct the DBD and reexecute the program.

DLZ924I INDEX SEGMENT IN DBD dbdname TOO SHORT. LENGTH SHOULD BE -- nnn 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.

Action: Correct the DBD in error, and reexecute the program.

DLZ925I LC OF SEGM segmname IN DBD dbdname HAS A SEQ FIELD 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.

Action: Correct the DBD in error, and reexecute the program.

DLZ926I INTERNAL PROG ERROR PROCESSING PSB psbname AT DISPLACEMENT nnnnnn IN ppppmddy

Explanation: An internal program logic error has occurred in program pppp (DLBL or UAMB) with the latest change date mddy (m = month 1 to C, dd = day, y = last digit of year) while processing the named PSB.

Action: Remove the offending PSB from the processing list, and reexecute the program. An APAR should be submitted including the information in this message. To do this, call an IBM Programming Systems Representative.

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.

Action: 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.

Action: Correct the DBD in error, and reexecute the program.

DLZ929I REQUIRED MODULE NOT FOUND IN CIL... {psbname|dbdname}

Explanation: A DLZBLDL macro was issued for the named PSB or DBD which was not found in a core image library that was assigned.

Action: Link-edit the PSB or DBD into the core image library or assign the private core image library in which it already exists and reexecute the program.

DLZ930I NOT SENSITIVE TO ALL SEARCH FIELDS FOR ALTERNATE SEQUENCE .. SEGMENT segname

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.

Action: Redo 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.

Action: Correct the appropriate PSB and/or DBDs, and reexecute the program.

DLZ932I INDEX DBD dbdname HAS NO SEQ FIELD

Explanation: The named INDEX DBD does not have a sequence field defined for the index segment.

Action: 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.

Action: Correct the named DBD and reexecute the program.

DLZ934I SENSITIVITY SPECIFIED FOR MORE THAN ONE SECONDARY INDEX .. SEGMENT segname

Explanation: More than one SENSEG statement referencing a XDFLD field was supplied for a segment.

Action: Redo the PSBGEN and remove the SENFLD statement(,) for the addidional XDFLD(,).

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 Chapter 5 of the *DL/I DOS/VS System/Application Design Guide* 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.

Action: 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.

Action: Correct the PSB and reexecute the program.

DLZ938I INCOMPLETE SENSITIVITY TO CONCATENATED KEY FOR SEGMENT segname

Explanation: Insert sensitivity was specified for a concatenated segment with field sensitivity specified but the sensitivity to the entire concatenated key was not specified.

Action: Do the PSBGEN over and either,

- Drop insert sensitivity, or
- Include sensitivity to the concatenated key.

DLZ939I 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.

Action: Correct the DBD and reexecute the job.

DLZ940I INVALID ALT PROCSEQ SPECIFIED FOR SENSEG--sensegname 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 DBD's for the index and the data base do not match the segment structure defined in the SENSEG statements of the PCB.

Action: Correct the DBD or PCB and reexecute the job.

DLZ941I PHYSICAL FIELD name WAS NOT FOUND IN SEGMENT segname

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.

Action:
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.

DLZ942I INCOMPLETE SENSITIVITY TO SEQ FIELD fldname IN SEGMENT segname

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.

Action:

- Specify sensitivity to the required field at PSBGEN time.
- Redesign and compile the application module to supply the identified field.

DLZ943I 't'1 TO 't'2 FIELD CONVERSION NOT SUPPORTED - SEGMENT segname

Explanation: The type of field supplied in the TYPE=t parameter in the SENFLD statement and the type of field supplied in the TYPE=t parameter in the FIELD statement specify a conversion that is not supported by DL/I.

Action:
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.

DLZ944I SENSEG LENGTH INVALID FOR FIXED LENGTH SEGMENT - segname

Explanation: Specifications of a length on a SENSEG statement is valid when the related physical segment is variable length. For concatenated segments, the length must include at least part of the destination parent and the destination parent must be variable length.

Action:
Either,

- Redesign the associated segment as variable length, or
- Increase the length specification to include part of the destination parent.

DLZ945I PCB dbdname IN PSB psbname HAS NO SENSITIVE SEGMENTS

Explanation: A PCB with no SENSEG statements was submitted.

Action: Correct the PCB and perform a new PSB generation. Then reexecute the application control blocks utility.

DLZ946I FIELD fieldname IN SEGMENT segname 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).

Action: Redo the PSBGEN (and DBDGEN if necessary) to define sensitivity to the additional fields.

DLZ947I FIELD fldname/fldname2 INTERSECT IN SEGMENT segname 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.

Action: Redo 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 segname

Explanation: 'fldname1' and 'fldname2' appear in both views of a segment and either 'fldname1' is not 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 and not in the physical view.

Action: Redo 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 segname

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.

Action: Redo 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 xx

Explanation: The issuing utility received an error return code from the work data set generator module DLZDSEH0. The xx is replaced by the code which is one of the following:

Code	Meaning
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.

Action:

Code	Possible Error
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 being performed as described in Chapter 6 "Data Base Reorganization/Load Processing" of the *DL/I DOS/VS Utilities and Guide for the System Programmer* (see Note 1 of the section titled "Data Base Logical Relationship 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 job control requirements in the *DL/I DOS/VS Utilities and Guide for the System Programmer.*

DLZ953I DATA BASE OPEN FAILURE, data-base-name

Explanation: The issuing program received an error code from DLZDLOC0 while attempting to open a VSAM data base. The job is terminated.

Action: 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.

Action: 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 defined in the *DL/I DOS/VS Logic Manual.*

Action: 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 BLOCKS, 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.

Action: 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.

Action: Ensure that a valid control file is provided to the issuing program. Reexecute DLZURPR0, if necessary.

DLZ958I BUFFER HANDLER ERROR RETURN, code

Explanation: The issuing program received a nonzero return code from the buffer handler. The code is an error code. The meaning may be found in the PST DSECT under the label 'PSTRTCDE'.

Action: 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. The meaning of the return code may be found in Chapter 5 of this manual.

Action: 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.*

Action: 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.

Action: None.

DLZ962I DBS=database-name,segment-name

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.

Action: 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.

Action: 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.

Action: Supply the required control cards and resubmit the job.

DLZ965I UNABLE TO CONSTRUCT CONTROL LIST ENTRY, file-name1 segment-name2 file-name2 segment-name2

Explanation: The issuing program was attempting to construct a control list entry for the logical relationship that was defined to exist between the above-named files and segments. The definition was not verified in the control blocks available to DL/I.

Action: 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.

Action: None.

DLZ967I UTILITY CHECKPOINT, pgm-name chkpt-number

Explanation: The issuing program completed a checkpoint at one of the intervals specified by the user. The name of the program issuing the message and the checkpoint number are shown.

Action: Record the checkpoint number in case a restart action must be requested.

DLZ968I BEGIN EXECUTION, progname date time

Explanation: The program named is starting to execute. The date and time are also printed.

Action: None.

DLZ969I SEGMENT NOT FOUND IN CONTROL BLOCKS, segment-name

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.

Action: Ensure that valid DMBs are available for the supplied input control cards and/or control file. Resubmit the job.

DLZ970I SCAN PROCESSING STARTED, database-name

Explanation: The issuing program started the scan of the named data base. This message is used by DLZURGS0.

Action: None.

DLZ971I SCAN PROCESSING COMPLETED

Explanation: The issuing program completed the scan of the data base named by the preceding DLZ970I message.

Action: None.

DLZ972I DATA BASE NOT SCANNED, database-name

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.

Action: Ensure that valid DMBs are available for the supplied input control cards and/or control file.

DLZ973I SEGMENT NOT SCANNED, database-name segment-name

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.

Action: 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.

Action: 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 the *DL/I DOS/VS Utilities and Guide for the System Programmer* for instruction regarding 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 Utilities and Guide for the System Programmer* pertaining to the program issuing this message.

Action: 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.

Action: 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 type-00 input work file record that was for the same occurrence of a logical parent/logical child pair as a previous input work file record. 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* for description of the record format. Prefix resolution may be invalid if this message is issued. The affected data base(s) must not be used.

Action: 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. Ensure that only work files from data bases currently being initially loaded, reorganized, or scanned are supplied as input to the issuing program.

DLZ978I CAUTION – NO LOGICAL CHILD RECORD FOUND 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 the user intended to load or reload the logical children belonging to the logical parent.

Action: 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.

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* 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.

Action: 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* 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.

Action: 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* for description of the record format. Prefix resolution may be invalid if this message was issued. The affected data base(s) should not be used.

Action: 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 resolution utility, the number indicates in which sort phase the last-detected error occurred and the code indicates the sort/merge return code. Prefix resolution may be invalid if this message is issued. Any affected data base(s) should not be used.

If the message was issued by the prefix update utility, the number is the number of input work file records processed.

Action: Determine the cause of the error (either return codes or error messages) 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* for a description of the record format.

Action: 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.

Action: 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, database-name1, segment-name1 database-name2 segment-name2

Explanation: The issuing program detected a failure in the limit check for the logical parent/logical child indicated by the above-named database-names and segment-names. The limit check performed is as described in Chapter 6 of the *DL/I DOS/VS Utilities and Guide for the System Programmer* under the heading "Data Base Logical Relationship Resolution Utilities", "Restrictions". This message is cautionary only, since the issuing program assumes worst case in computing the limit check.

Action: The user should determine the cause of limit check failure. If none of the components of the limit check can be omitted, the user 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 MULTIPLE LOGICAL CHILD RECORDS/LP. NO LT PTRS SPECIFIED
One or more lines containing input record in hexadecimal and EBCDIC format.

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* 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 accessable through the physical path and have proper pointers to the logical parent.

Action: The user should determine if this condition is to his 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 I/O ERROR OCCURRED ON FILE - dtfname

Explanation: An unrecoverable I/O error was detected for the named file through an error exit in the DTF.

Action: 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 prereorganization 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.

Action: 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.

A dump of the partition will be produced. Check your control statements and resubmit the job after correcting any errors.

DL/I System/Task Abnormal Termination

The numbers of the DL/I DOS/VS 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 will indicate that the DL/I 'system' terminated abnormally. If, however, DL/I was running in an online environment when the condition was detected, the message text will indicate that the DL/I 'task' terminated abnormally. In either case, possible causes for the contradictory conditions are:

- Contradictions between application program and application control blocks.
- 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, the user should make sure that he is using the correct ACBs with his application, that he was not working with a previously destroyed (not backed-out and/or recovered) data base, and that the application program did not destroy DL/I code or control blocks.

If the above is true, the user should contact his local IBM programming systems representative and request that an APAR (Authorized Program Analysis Report) be submitted along with appropriate supporting documentation. In any case, the data base backout utility must be executed to guarantee that the data bases involved are restored to a valid status.

With a few exceptions, the individual explanations for these messages provide assistance for APAR preparation and processing. For this reason, the explanations describe failure conditions in DL/I internal terminology.

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 message, the numeric part of the code that follows the word ABEND corresponds to the numeric portion of the applicable DL/I message number as listed in Chapter 1 of this publication. The code normally begins with D but it begins with E if the termination cannot be noted on the transient data destination CSMT.

In addition, certain ABEND codes are used by DL/I which do not abide by the above conventions. These codes are:

Code	Meaning
DLPV	System scheduling call issued with invalid password.
DBPC	DLZBPC00 (MPS batch task) terminated abnormally.
DMPC	DLZMPC00 (MPS master partition controller task) terminated abnormally.
DACT	TERM call issued by program not in the ACT.

CICS/VS Dump Codes

The following codes are used to identify CICS/VS dumps taken by DL/I:

Code	Meaning
DLI,DLI2	Identifies 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 ODP when a task with no PST acquired abends 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 DOS/VS 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.

DBDGEN Error Messages

DBD001 DBD CARD MUST OCCUR 1ST IN A DBDGEN.
Explanation: Self-explanatory.

EXT001 EXTERNAL REFERENCE TABLE FULL.
Explanation: More than 255 data bases were specified in this DBDGEN.

DBD Macro

DBD100 MORE THAN ONE DBD CARD SPECIFICATION.
Explanation: Self-explanatory.

DBD110 NAME OPERAND IS OMITTED OR INVALID.
Explanation: The name operand has been omitted, contains more than seven characters, or the first character of the entered name does not begin with an alpha character.

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.

DBD130 RMNAME OPERAND IS INVALID.
Explanation: The entered randomizing module name does not begin with an alpha character or the ACCESS operand value does not equal HDAM, or is missing for HDAM.

DATASET Macro

DMAN100 MORE THAN ONE DATA SET STATEMENT PROVIDED.
Explanation: More than one DATASET statement has been specified.

DMAN110 DD1 OPERAND IS OMITTED OR INVALID.
Explanation: Self-explanatory.

DMAN120 DD2 OPERAND IS OMITTED OR INVALID.
Explanation: Self-explanatory - occurs in HSAM or SHSAM.

DMAN140 OVFLW OPERAND IS OMITTED OR INVALID.
Explanation: Self-explanatory.

DMAN150 DEVICE OPERAND IS OMITTED OR INVALID.
Explanation: Self-explanatory.

DMAN160 'TAPE' DEVICE IS INVALID FOR ACCESS=access.
Explanation: Tape devices are only supported for HSAM or HISM.

DMAN170 ENTERED DDNAME IS A DUPLICATE - name.
Explanation: Self-explanatory.

DMAN200 RECORD OPERAND IS INVALID.
Explanation: The entered RECORD operand is not a self-defining term.

DMAN210 BLOCK OPERAND IS INVALID.
Explanation: The entered BLOCK operand is not a self-defining term. This error message may occur if the specified or implied block size exceeds 4086 bytes.

DMAN211 BLOCK PARAMETER PLUS ROOT ANCHOR POINTS EXCEED TRACK LENGTH
Explanation: The record length specified on the BLOCK parameter plus the size of the root anchor points exceeded the track length for the specified HDAM or HIDAM device.

DMAN215 BLOCKSIZE FOR H-D MUST BE MULT OF 512 NOT GT 4096. BLOCKSIZE SET TO 2048.
Explanation: Self-explanatory.

DMAN220 LOGICAL DATASET ASSUMED
Explanation: Self-explanatory; warning message.

DMAN230 KEYWORD SPECIFICATIONS ARE INVALID FOR A LOGICAL DATASET DEFINITION. IGNORED.
Explanation: Self-explanatory.

DMAN240 DD2 PARAMETER NOT ALLOWED FOR org
Explanation: The DD2 parameter is only valid for HSAM or HISAM organization.

DMAN280 SCAN OPERAND IS INVALID.
Explanation: The entered value is not a decimal integer value, or the entered value exceeds 255.

DMAN300 1ST RECORD LENGTH MUST BE EVEN. RESET TO xxx
Explanation: The entered record length value must be a multiple of 2.

DMAN302 2ND RECORD LENGTH MUST BE EVEN. RESET TO nnn.
Explanation: Extent record lengths must be a multiple of two. Warning.

DMAN310 OVFLW PARAMETER NOT ALLOWED
Explanation: Self-explanatory.

DMAN340 ACCESS METHOD LOGICAL MUST BE CODED FOR A LOGICAL DATA BASE. '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.

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.

DMAN400 ESDS REC LENGTH LT KSDS REC LENGTH -- RESET TO KSDS REC LENGTH (xxx)
Explanation: The entered value for ESDS cannot be less than the KSDS record length. Neither the ESDS nor the KSDS record length can be zero.

DMAN420 RECORD EXCEEDS TRACK LENGTH.
Explanation: Self-explanatory.

DMAN430 VSAM KSDS REC LEN CANNOT EXCEED 4086 RECORD LENGTH SET TO 4086.
Explanation: Self-explanatory.

DMAN440 VSAM ESDS REC LEN CANNOT EXCEED 4086 RECORD LENGTH SET TO 4086.
Explanation: Self-explanatory.

DMAN450 DEVADDR OPERAND OMITTED OR INVALID.
Explanation: Self-explanatory.

DMAN460 VALUE FOR NUMBER OF FREE BLOCKS IS INCORRECT IN FRSPC PARAMETER
Explanation: The free block (fbuff) value specified in the FRSPC parameter is not a value from 0 to 100, excluding 1.

Action: Correct the FRSPC parameter value and rerun the DBD generation.

DMAN470 FREE SPACE PERCENTAGE VALUE IS INCORRECT IN FRSPC PARAMETER
Explanation: The free space percentage (fspf) value specified in the FRSPC parameter is not a value from 0 to 99.

Action: Correct the FRSPC parameter value and rerun the DBD generation.

DMAN480 THE FRSPC PARAMETER CONTAINS MORE THAN TWO SUBPARAMETERS
Explanation: The FRSPC parameter can contain only two subparameters: free block value (fbuff) and free space percentage (fspf).

Action: Correct the FRSPC parameter value and rerun the DBD generation.

DMAN490 FRSPC PARAMETER IS VALID ONLY FOR HD ORGANIZATION - 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 HDAM or HIDAM.

Action: None required. The FRSPC parameter is ignored.

SEGM Macro

SEGM099 DBD GENERATION TERMINATED
Explanation: All macros following this message, except DBDGEN, are not processed, DGEN111 will also print.

SEGM100 DATASET CARD MUST PRECEDE SEGM.
Explanation: The name operand does not contain a source segment field and no DATASET card has been encountered.

SEGM110 SEGM SPECIFICATION LIMIT EXCEEDED.
Explanation: Self-explanatory.

SEGM111 ONE SGMT ONLY FOR access DB.
Explanation: Access = data base access method.

SEGM120 NAME OPERAND IS OMITTED OR INVALID.
Explanation: Self-explanatory.

SEGM140 SEGMENT NAME IS A DUPLICATE NAME.
Explanation: Self-explanatory.

SEGM150 PHYSICAL PARENT PARAMETER OMITTED
Explanation: The PARENT= ... parameter was omitted

SEGM152 ROOT SEGMENT CANNOT BE LOGICAL CHILD
Explanation: The first SEGM statement occurring in any DBD generation defines a root segment; therefore, the PARENT= parameter either should be omitted or the PARENT=0 parameter should be specified. If the PARENT=((seg-name2,...)) parameter is specified for a root segment, a second parent appears to be specified on the PARENT= operand for a root segment.

SEGM153 'PARENT=0' ASSUMED. PARAMETER IGNORED.
Explanation: More than one operand was specified in the first parent sublist parameter for the PARENT= keyword parameter for a root 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= parameter either should be omitted or the PARENT=0 parameter should be specified.

SEGM155 'ptr' POINTER SPECIFICATION IN PHYSICAL PARENT PARAMETER INVALID. DBLE ASSUMED.
Explanation: The second operand of the PARENT parameter must be omitted, 'SNGL' or 'DBLE'. Warning.

SEGM156 'oper' INVALID. IGNORED.
Explanation: A third parameter was supplied for the first PARENT sublist. Warning.

SEGM157 ONLY TWO SUBLISTS ALLOWED FOR PARENT PARAMETER. REMAINDER IGNORED.
Explanation: The PARENT parameter included an additional sublist. The acceptable two sublists are indicated by ,PARENT=((seg-name2, SNGL|DBLE), (lpseg-name,V,db-name1)). Sublist 1 is designated by (seg-name2,SNGL|DBLE) and sublist 2 is designated by (lpseg-name,V, db-name1). Any additional sublists are not allowed. Warning.

SEGM162 PARENT SUBPARAM LIST MUST BE ENCLOSED IN 2 PAIRS OF PARENTHESIS.
Explanation: When specifying both operands in the PARENT=0|((seg-name2,SNGL|DBLE,(lpseg-nam,V,db-name1)). Both pairs of double parentheses are required.

SEGM164 LEVEL GT 15 IS NOT ALLOWED.
Explanation: Self-explanatory.

SEGM170 SEGMENT PHYSICAL PARENT IS NOT DEFINED.
Explanation: Self-explanatory.

SEGM180 IF SOURCE SEGMENT PARAMETER IS SPECIFIED, ALL OTHER PARAMETERS ARE INVALID EXCEPT THE PHYSICAL PARENT NAME AND POINTER PARAMETER.

Explanation: Only the NAME=, PARENT=, and POINTER= paramaters may be used with SOURCE= parameter.

SEGM190 LOGICAL PARENT PARAMETER IS INVALID.

Explanation: The entered logical parent parameters are invalid.

SEGM191 'oper' INVALID. V ASSUMED.

Explanation: The second operand, V, of the second sublist, (lpseg-name, V, db-name), of the PARENT= parameter was not V. The 'V' operand was assumed.

SEGM192 LOGICAL CHILD CANNOT BE ITS OWN LOGICAL PARENT

Explanation: The name specified for the parameter PARENT=((seg-name2,...)) and for NAME=seg-name1 are the same.

SEGM200 LOGICAL PARENT DBNAME IS INVALID - name.

Explanation: The entered logical parent data base name does not begin with an alpha character or contains more than seven characters.

SEGM210 BYTES OPERAND IS INVALID.

Explanation: For the SEGM macro, BYTES must not be specified for a virtual logical child segment. The length of the segment must not exceed the size of the control interval used for this data base.

SEGM211 VARIABLE LENGTH STATEMENTS NOT SUPPORTED FOR org. 'n' BYTES FIXED LENGTH ASSUMED.

Explanation: Two operands were supplied for a segment in a data base for which variable length segments are not supported. An operand, 'n', was assumed.

SEGM212 MAXIMUM OF TWO OPERANDS FOR 'bytes' PARAMETER. REMAINDER IGNORED.

Explanation: The BYTES=(max-bytes,min-bytes) parameter had more than the two operands supplied. The first two operands were used as the 'max-bytes' and the 'min-bytes' values. All additional operands were ignored.

SEGM250 POINTER AND PTR OPERANDS CANNOT BOTH BE SPECIFIED.

Explanation: Either POINTER= or its abbreviation PTR= can be used for this 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.

SEGM271 LTWIN POINTERS ONLY VALID FOR LOGICAL CHILDREN.

Explanation: The POINTER=...,LTWIN parameter may only be specified if the segment being defined is a logical child in a bidirectional relationahip. This is indicated by the PARENT=((seg-name2, [SNGL|DBLE]), (lpsegname[V,db-name1])) parameter.

SEGM273 TWINBWD IMPLIES TWIN

Explanation: The POINTER=TWIN,TWINBWD parameter was used. Both TWINBWD and TWIN need not be specified.

SEGM274 LTWINBWD IMPLIES LTWIN

Explanation: The POINTER=LTWIN,LTWINBWD parameter was specified. Both LTWIN and LTWINBWD need not be specified.

SEGM280 RULES OPERAND IS INVALID.

Explanation: Self-explanatory.

SEGM281 RULES PARAMETER 'xxx' INVALID FOR org DATA BASE. IGNORED.

Explanation: Logical relationships are not supported for the indicated data base organizations. The specified logical insert, replace, or delete rules have no meaning.

SEGM290 SEGMENT LENGTH IS GREATER THAN SPECIFIED RECORD LENGTH PLUS ROOT SEGMENT KEY.

Explanation: Self-explanatory.

SEGM300 SOURCE SEGMENT IS INVALID FOR org DBD'S.

Explanation: A SOURCE operand is specified for a segment in a data base organization other than HDAM, HIDAM, or LOGICAL.

SEGM310 SOURCE SEGMENT IS INVALID.

Explanation: More than two segments were specified in the SOURCE operand.

SEGM320 HIERARCHY SEQUENCE ERROR.

Explanation: The named parent of the current segment violated hierarchical sequence.

SEGM330 SEGM (name) LENGTH + PREFIX + LRECL EXPANSION (length1) EXCEEDS REC LENGTH (length2)

Explanation: name = segment name
length1 = segment name + record prefix + LRECL expansion
length2 = record length.

SEGM350 SEGMENT (name) SIZE EXCEEDS CI LENGTH SEGMENT SIZE=n CI LENGTH=n

Explanation: Self-explanatory.

SEGM410 INVALID OR INCOMPLETE SEGMENT PAIRING.

Explanation:

1. SEGM statement had PAIRED specified in its PTR operand and its parent did not have an LCHILD statement naming it in the LCHILD's PAIR operand.
2. Parent segment has no associated LCHILD statements.
3. Segment is the root segment.
4. Segment's parent is invalid.

SEGM430 PTR/POINTER OPERAND IS INVALID.

Explanation: Self-explanatory.

SEGM431 'POINTER' OPERAND INVALID FOR LOGICAL DATA BASE. IGNORED.

Explanation: When constructing a logical data base (DATASET LOGICAL), the POINTER= parameter is not used with the SEGM macro statement. Rather than terminate your program a warning message was given to bring to your attention the unused POINTER= parameter.

SEGM432 'POINTER' OPERAND INVALID. 'PAIRED' ASSUMED.

Explanation: When defining a virtual logical child segment of a bidirectional logical relationship the POINTER=PAIRED parameter should be used. In this case, the correct parameter was not used but the macro expansion inserted the correct option and issued a warning message to the user.

SEGM433 'POINTER' OPERAND OMITTED. 'PAIRED' ASSUMED.

Explanation: When defining a virtual logical child segment of a bidirectional logical relationship the POINTER=PAIRED parameter was omitted. In this case, the missing parameter was inserted by the macro expansion and a warning message was issued to the user.

SEGM440 COMPRTN OPERAND IS INVALID.

Explanation: The COMPRTN=(mod-name,...) parameter was not specified as a valid 1-8 character alphanumeric value.

SEGM441 COMPRESSION NOT SUPPORTED FOR org DATA BASES

Explanation: Compression is supported only for HDAM and HIDAM data bases.

SEGM442 COMPRESSION ONLY VALID FOR VARIABLE LENGTH SEGMENTS

Explanation: Compression allows the reduction in length of variable length segments to increase the effective utilization of secondary storage. VLCs and secondary index ISS segments cannot be variable length.

SEGM460 INDEX SOURCE SEGMENT IS A LOGICAL SEGMENT. REFERENCE XDFLD STATEMENT NAME - name

Explanation: An index source segment may not be a logical segment.

SEGM461 INDEX SOURCE SEGMENT CANNOT BE COMPRESSED. REFERENCE XDFLD STATEMENT NAME - name

Explanation: The segment is a fixed-length segment and was specified in the XDFLD statement 'name'. Compression can be performed on variable-length segments which are not virtual logical child segments or secondary index source segments. Compression is allowed for HDAM or HIDAM data bases only.

SEGM462 'DATA' OPERAND MISSING. 'DATA' ASSUMED.

Explanation: The second operand (D) of the COMPRTN parameter was missing. 'DATA' is used by default.

SEGM463 'oper' OPERAND INVALID. 'DATA' ASSUMED.

Explanation: The second operand (D) of the COMPRTN parameter was not the expected operand. 'DATA' is assumed to be the correct operand.

SEGM464 'oper' OPERAND INVALID. 'INIT' ASSUMED.

Explanation: The third operand (INIT) of the COMPRTN parameter was not the expected operand. 'INIT' is assumed to be the correct operand.

SEGM470 SOURCE SEGMENT TABLE LIMIT EXCEEDED.

Explanation: More than 255 source operand values encountered in a single DBDGEN.

SEGM480 SOURCE SEGMENT NAME IS OMITTED.

Explanation: Self-explanatory.

SEGM490 SOURCE SEGMENT DATA OPTION IS INVALID. OPTION='option'. 'DATA' ASSUMED.

Explanation: The second operand (D) of SEGM500 the COMPRTY parameter was not the expected operand (D). 'DATA' is assumed to be the correct operand

SEGM500 SOURCE SEGMENT DATABASE IS INVALID - name.

Explanation: Entered data base name is not an ordinary symbol.

XDFLD Macro

XDFLD001 XDFLD IS VALID FOR HIDAM OR HDAM ONLY.

Explanation: An XDFLD statement is invalid when the access method specified for the DBD generation is not HIDAM or HDAM.

XDFLD002 LCHILD CARD MUST PRECEDE XDFLD CARD. FIRST LCHILD CARD IN HIDAM DBD GEN BELONGS TO THE PRIMARY INDEX

Explanation: In a HDAM DBD, or in a HIDAM dependent segment, no LCHILD statement with the POINTER=INDX parameter preceded this XDFLD macro statement. For a HIDAM root segment, either none or one LCHILD macro statement preceded the XDFLD macro statement; a LCHILD macro statement should precede the XDFLD statement to define the index pointer segment type in the Index data base and the index target segment type in the HDAM or HIDAM data base. An additional LCHILD must precede the first XDFLD statement for a HIDAM root segment to define the index pointer segment for the primary index.

XDFLD003 SEGM CARD MUST PRECEDE XDFLD CARD.

Explanation: No SEGM statements have yet been encountered in this DBD.

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.

XDFLD005 SRCH OPERAND IS OMITTED OR INVALID.

Explanation: Either the SRCH operand was not specified, or more than five parameters were specified, or a specified parameter contained more than eight characters.

XDFLD006 SUBSEQ OPERAND IS INVALID.

Explanation: Either more than five parameters were specified or a specified parameter contained more than eight characters.

XDFLD007 DDATA OPERAND IS INVALID.

Explanation: Same as message XDFLD006.

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.

XDFLD010 NULLVAL OPERAND IS INVALID.

Explanation: The NULLVAL operand was not specified as a self-defining term, or zero, or blank.

XDFLD011 SEGMENT OPERAND IS INVALID.

Explanation: Either more than one parameter was specified or the specified parameter contains more than eight characters.

XDFLD012 SRCH/DDATA/SUBSEQ/ EXTRTN/NULLVAL NAME TABLE LIMIT EXCEEDED.

Explanation: The total listed specifications may not exceed 255.

XDFLD013 SEGMENT XDFLD LIMIT EXCEEDED.

Explanation: Only 32 XDFLD statements per SEGM statement are allowed.

XDFLD014 SPECIFIED NAME MAY NOT BE OBJECT OF PREVIOUS XDFLD SRCH, DDATA, OR SUBSEQ.

Explanation: The specified NAME operand is invalid.

XDFLD015 INDEX SOURCE STATEMENT MAY NOT PRECEDE INDEX TARGET SEGMENT.

Explanation: The source segment for a secondary index must be either the target segment or a dependent of the target segment.

XDFLD020 PREVIOUS LCHILD POINTER OPERAND INVALID.

Explanation: The LCHILD statement preceding this XDFLD statement did not specify an index relationship by including a PTR=INDX operand specification.

XDFLD022 INDEX SOURCE SEGMENT A LOGICAL SEGMENT.

Explanation: An index source segment may not be a logical segment.

XDFLD023 XDFLD STATEMENT IS INVALID FOR A LOGICAL CHILD SEGMENT OR A PHYSICAL DEPENDENT OF A LOGICAL CHILD SEGMENT. REFERENCE SEGM STATEMENT - name of segment.

Explanation: Self-explanatory.

FIELD Macro

FLD100 SEGM CARD MUST PRECEDE FIELD.

Explanation: Self-explanatory.

FLD110 NAME OPERAND IS OMITTED

Explanation: The "NAME=" parameter or the fld-name1 operand was omitted in the NAME=(fld-name1,...) parameter.

FLD111 NAME OPERAND IS NOT 1-8 CHARACTERS

Explanation: The NAME=(fld-name1,...) parameter specifies the name of a field that may be referred to by an application program in a DL/I call SSA. The 'fld-name1' parameter must be a 1-to-8 character alphanumeric value.

FLD112 'SEQ' OPERAND INVALID. 'SEQ' ASSUMED.

Explanation: The second operand (SEQ) of the NAME=(fld-name1,SEQ,...) parameter identifies the field as a sequence (key) field in the segment with which it is associated. In this case, SEQ was assumed to be the correct parameter rather than terminating the compilation.

FLD113 'SEQ' OPERAND INVALID. IGNORED.

Explanation: The second operand (SEQ) of the NAME=(fld-name1,SEQ,...) parameter identifies the field as a sequence (key) field in the segment with which it is associated. In this case, SEQ was assumed to be the correct parameter rather than terminating the compilation.

FLD114 'M/U' OPERAND INVALID. 'M' ASSUMED.

Explanation: The third operand (U|M) of the NAME=(fld-name1,SEQ,U|M) parameter either unique (U) values or duplicate (M) values of this segment field can occur in multiple occurrences of the segment. The third operand was present and was neither 'U' nor 'M'. In this case, 'M' was assumed to be the correct paramater rather than terminating the compilation.

FLD115 TOO MANY OPERANDS IN NAME FIELD. REMAINDER IGNORED.

Explanation: The NAME=(fld-name1,SEQ,U|M) parameter allows for a maximum of three operands. This message indicates the user supplied more than three operands and should check the FIELD macro statement for correct syntax. In this case, the remaining operands were ignored rather than terminating the compilation.

FLD120 SEGMENT name HAS MULTIPLE SEQUENCE FIELDS.

Explanation: Self-explanatory.

FLD125 OPTION M NOT ALLOWED

Explanation: The second subparameter of NAME parameter must not be M for a root segment of HISAM, HIDAM or INDEX data bases.

FLD130 TYPE OPERAND IS NOT X, H, F, P, C, Z, E, D, or L

Explanation: For the TYPE=t parameter, the only valid values for 't' are X, H, F, P, C, Z, E, D, or L.

FLD132 'TYPE' PARAMETER INVALID FOR '/SX' FIELD TYPE. IGNORED.

Explanation: When defining system related fields for secondary indexing, the TYPE=X|P|C parameter is not used. In this case, the parameter provided was ignored rather than terminating the compilation.

FLD150 BYTES OPERAND IS OMITTED OR NOT IN RANGE 1 - 256

Explanation: The BYTES=bytes parameter must be a valid self-defining term whose value does not exceed 256.

FLD151 BYTES OPERAND IS OMITTED OR NOT IN RANGE 1 - 3825.

Explanation: Either the BYTES=bytes parameter was omitted or it is not a valid self-defining term whose value does not exceed 3825.

FLD152 'BYTES'PARAMETER INVALID FOR '/SX' FIELD TYPE. IGNORED.

Explanation: When referring to the index source segment's VSAM RBA, the BYTES=bytes parameter is not used. In this case, the parameter was ignored rather than terminating the compilation.

FLD153 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 parameters. Floating point has implicit lengths (E=4, D=8, L=16) that may be overridden.

FLD160 START OPERAND IS LESS THAN 1.

Explanation: The START=pos parameter, if it is a numeric term, must be 1 or greater. START=0 is not a valid parameter.

FLD161 START OPERAND IS GREATER THAN 32767
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.

FLD162 START OPERAND IS GREATER THAN 3825
Explanation: The maximum allowable cannot be greater than 3825 bytes.

FLD163 'START' PARAMETER INVALID FOR '/SX' FIELD TYPE. IGNORED.
Explanation: When referring to the index source segment's VSAM RBA, the START=pos parameter is not used. In this case, the parameter is ignored rather than terminating the compilation.

FLD164 'START' FIELD NOT PREVIOUSLY DEFINED
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.

FLD170 FIELD EXTENDS BEYOND DEFINED SEGMENT LENGTH.
Explanation: Self-explanatory.

FLD180 1020 FIELDS/DBD LIMIT EXCEEDED.
Explanation: The maximum number of FIELD and XDFLD statements together cannot exceed 1020.

FLD190 DUPLICATE FIELD NAME IN SEGMENT name.
Explanation: Self-explanatory.

FLD195 MORE THAN ONE /SX FIELD SPECIFIED WITHIN SEGMENT segname.
Explanation: The field name starting with the characters '/SX' is reserved to indicate a name by which the VSAM RBA of an index source segment for a secondary index may be referenced as a SUBSEQ field to quantify the search field. Only one such field may be identified per segment.

FLD200 225 FIELDS/SEGM LIMIT EXCEEDED.
Explanation: The maximum number of FIELD and XDFLD statements together is 255 per SEGM macro statement.

FLD210 FIELD STATEMENT IS NOT ALLOWED IN LOGICAL DATA BASE DEFINITIONS. IGNORED.
Explanation: Self-explanatory.

FLD225 SEQUENCE FIELD SPECIFICATION IS INVALID FOR '/CK' OR '/SX' FIELD TYPES.
Explanation: A systems related field name may not be specified as a sequence field.

FLD240 LENGTH OF SEQUENCE FIELD IN ROOT SEGMENT EXCEEDS 236.
Explanation: Sequence field length value for a root segment cannot exceed 236. This restriction is necessary to allow DOS/VS sort to function properly in the change accumulation utility.

FLD999 START+BYTES EXCEEDS CONCATENATED KEY LENGTH CONCATENATED KEY LENGTH = length.
Explanation: Self-explanatory.

LCHILD Macro

LCHD100 SEGM CARD MUST PRECEDE LCHILD.
Explanation: Self-explanatory.

LCHD110 LCHILD SPECIFICATION LIMIT EXCEEDED.
Explanation: The number of LCHILD macros plus the number of SEGM's for logical children exceeded 255.

LCHD111 PREVIOUS LCHILD STATEMENT MUST BE FOLLOWED BY A XDFLD STATEMENT
Explanation: An LCHILD macro statement specifying the POINTER=INDX parameter (other than the first LCHILD macro statement for the root segment in HIDAM) must be followed by a XDFLD macro statement before the next SEGM, LCHILD or DBDGEN macro statement.

LCHD112 ONLY 1 LCHILD ALLOWED
Explanation: An INDEX DBD may have only one LCHILD statement.

LCHD113 LCHILD NOT ALLOWED FOR org DATA BASES
Explanation: The LCHLD statement is only allowed in a HIDAM, HDAM or INDEX generation.

LCHD120 NAME OPERAND IS OMITTED OR INVALID.
Explanation: Self-explanatory.

LCHD130 DATABASE NAME PARAMETER IS INVALID.
Explanation: The entered data base name parameter of the NAME operand does not begin with an alpha character or is longer than seven characters.

LCHD131 INDEXED DATA BASE CANNOT HAVE THE SAME NAME AS INDEX DATA BASE
Explanation: The data base name specified in the NAME=(seg-name1,db-name) parameter is the same as the name specified in the "DBD NAME=dbname" with "ACCESS=INDEX". They cannot be the same.

LCHD140 POINTER/PTR OPERAND IS INVALID
Explanation: The valid POINTER=... parameter operands are DBLE, INDX, NONE, or SNGL.

LCHD141 MUST SPECIFY POINTERS FOR BI-DIRECTIONAL RELATIONSHIP. 'POINTER=SNGL' ASSUMED
Explanation: The PAIR=seg-name2 parameter was specified and no pointers, the POINTER=NONE|SNGL|DBLE parameter, were identified in the logical parent to the logical child. A single pointer was assumed rather than terminating the compilation.

LCHD142 POINTER AND PTR MAY NOT BOTH BE SPECIFIED
Explanation: Both the POINTER=NONE|SNGL|DBLE parameter and the PTR=NONE|SNGL|DBLE parameter were specified. Only one may be specified.

LCHD143 'POINTER' MUST BE 'SNGL' FOR INDEX. RESET.
Explanation: When using the LCHILD macro statement to define the INDEX data base, the POINTER=SNGL parameter must be used. The pointer specification was reset to SNGL rather than terminating the compilation.

LCHD144 FIRST LCHILD FOR HIDAM MUST BE FOR PRIMARY INDEX.

Explanation: Following the SEGM statement that defines the root segment in a HIDAM data base generation there must be an LCHILD macro statement that names the index pointer segment type in an INDEX data base with "POINTER=INDEX".

LCHD145 INDICES NOT SUPPORTED FOR org DATA BASE

Explanation: Indexes are only supported for HIDAM and HDAM data bases.

LCHD150 PAIR OPERAND IS INVALID.

Explanation: The entered value is greater than eight characters.

LCHD160 INDEX OPERAND IS INVALID.

Explanation: The ACCESS=INDEX operand does not appear on the DBD statement or the INDEX operand appears on an LCHILD statement not associated with ACCESS=INDEX.

LCHD165 INDEX OPERAND IS MISSING.

Explanation: Self-explanatory.

LCHD170 LCHILD STMT IS INVALID FOR LOGICAL SEGMENTS.

Explanation: Self-explanatory.

LCHD180 SEGMENTS WITH A LOGICAL PARENT CANNOT ALSO HAVE LOGICAL CHILDREN.

Explanation: Self-explanatory.

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.

LCHD200 REFERENCED LCHILD CANNOT PRECEDE LCHILD STMT.

Explanation: Self-explanatory.

DBDGEN Macro

DGEN100 DBD CONTAINS NO SEGMENTS.

Explanation: Self-explanatory.

DGEN110 DBDGEN TERMINATED; ERRORS IN DBD.

Explanation: Self-explanatory.

DGEN120 SEGMENT name NOT IN DATABASE name.

Explanation: The named segment has been defined as existing in the data base currently being defined. No SEGM statement was found to identify the named segment in its NAME operand.

DGEN121 ROOT SEGMENT CONTAINS NO LCHILD

Explanation: HIDAM or INDEX organization root segments must have LCHLD statements.

DGEN130 REFERENCED PAIRED SEGMENT DOES NOT SPECIFY 'PAIRED' IN ITS POINTER OR PTR OPERAND. PAIRED SEGMENT NAME IS name.

Explanation: Self-explanatory.

DGEN160 ROOT SEGMENT HAS NO SEQUENCE FIELD.

Explanation: Self-explanatory.

DGEN180 REFERENCED PAIRED SEGMENT CANNOT HAVE ANY CHILDREN, SEGMENT NAME = name.

Explanation: The named logical segment has physical children.

DGEN200 {INPUT} BLOCK SIZE (xxx)
{OUTPUT}
{KSDS}
{ESDS}
EXCEEDS MAXIMUM (yyy)

Explanation: The specified blocking factor caused block size to exceed track capacity (SHSAM or HSAM) or CI size.

DGEN201 {KSDS} BLOCKING FACTOR RESET TO xxx
{ESDS}
xxx = new blocking factor.

Explanation: The specified blocking factor was invalid and DL/I has reset the value.

DGEN202 ESDS RECORD LENGTH RESET TO xxx
xxx = new record length.

Explanation: The specified record length was invalid and DL/I has reset the value.

DGEN210 SPECIFIED INDEX SOURCE SEGMENT NAME NOT FOUND. NAME EQUALS - name.

Explanation: A XDFLD macro statement identified a source segment name for an index that could not be found at DBDGEN time.

DGEN220 SPECIFIED SRCH FIELD NAME name NON-EXISTANT IN INDEX SOURCE STATEMENT

Explanation: A XDFLD macro statement identified a SRCH field by name that could not be located in the identified source segment at DBDGEN time. A field by the specified name exists in another segment.

DGEN230 SPECIFIED SRCH FIELD NAME NON-EXISTANT. FIELD NAME EQUALS - name.

Explanation: A XDFLD macro statement identified a SRCH field by name that could not be located in the identified source segment at DBDGEN time.

DGEN240 SPECIFIED SUBSEQ FIELD NAME - name NON-EXISTANT IN INDEX SOURCE STATEMENT

Explanation: A XDFLD statement identified a SUBSEQ field by name that could not be located in the identified source segment at DBDGEN time. A field by that name exists in another segment.

DGEN250 SPECIFIED SUBSEQ FIELD NON-EXISTANT. FIELD NAME EQUALS - name.

Explanation: A XDFLD statement specified a SUBSEQ field by name that could not be located in any segment at DBDGEN time.

DGEN260 SPECIFIED SOURCE FIELD NAME - name NON-EXISTANT IN INDEX SEGMENT.

Explanation: A XDFLD macro statement specified a source field by a name that could not be located in the identified source segment at DBDGEN time. A field by this name exists in another segment.

DGEN270 SPECIFIED SOURCE FIELD NAME NON-EXISTANT. FIELD NAME EQUALS - name.

Explanation: A XDFLD macro statement identified a SOURCE field by a name that could not be located in any segment at DBDGEN time.

DGEN340 SEGM (name) LENGTH (length) EXCEEDS INPUT REC LENGTH (length)

Explanation: The input record length for the SHSAM data base cannot accomodate the root segments of this data base.

DGEN345 SEGMENT (name) LENGTH (length 1) EXCEEDS OUTPUT RECORD LENGTH (length 2)

Explanation: The output record length for s SHSAM data base cannot accomodate the root segments of this data base.

FINISH Macro

FIN100 NO SUCCESSFUL DBD IN THIS RUN.

Explanation: Self-explanatory.

Chapter 3: Program Specification Block Generation (PSBGEN) Error Messages

As PSB generation is composed of DOS/VS 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 DOS/VS during PSB generation.

The following are the PSB generation error conditions and the messages displayed for these conditions:

PCB Macro

PCB100 PCB SPECIFICATION LIMIT 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.

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: Self-explanatory.

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: Self-explanatory.

PCB230 DBDNAME OPERAND MUST BE ALPHA
Explanation: The entered data base name must begin with an alpha character for DB PCBS.

PCB260 PROCSEQ OPERAND IS INVALID.
Explanation: Self-explanatory.

PCB270 PROCOPT SPECIFICATION IS INVALID.
Explanation: PROCOPT=L or PROCOPT=LS was specified along with a PROCSEQ specification.

PSBGEN Macro

PGEN100 PSBNAME OPERAND IS OMITTED OR INVALID.
Explanation: Self-explanatory.

PGEN110 LANG OPERAND IS OMITTED OR INVALID.
Explanation: Self-explanatory.

PGEN120 PSBGEN TERMINATED; ERRORS IN PSB.
Explanation: Self-explanatory.

PGEN130 PCB name HAS NO SENSITIVE SEGMENTS.
Explanation: Warning message, self-explanatory.

PGEN140 SYSTEM ERROR; PSBGEN TERMINATED.
Explanation: An error occurred during source segment table generation (should not occur).

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.

SFLD150 START PARAMETER IS NOT IN RANGE 1 - 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 through 32767.

SFLD155 'START' FIELD NOT PREVIOUSLY DEFINED
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.

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 designated alphabetic characters (X, H, F, C, Z, P, E, D or L) substituted for 't'.

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 that may not be overridden. (E=4, D=8, L=16)

SFLD190 BYTES PARAMETER NOT IN RANGE 1 - 256
Explanation: The length of the field must be within the range of 1 to 256 bytes. It must be numeric or 'p'.

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=prog parameter, 'prog' is the name of a module in the DOS/VS core image library that is given control whenever this field is retrieved or stored. It must be a 1 - 8 character name; the first character must be alphabetic, and the remaining characters may be alphanumeric.

SFLD230 OPTION AND OPT NOT ALLOWED ON ONE SENFLD
Explanation: Do not specify both OPT=... and OPTION=... 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.

SENSEG Macro

SEG100 SENSEG SPECIFICATION LIMIT EXCEEDED.
Explanation: More than 255 SENSEG statements have been encountered for a PCB.

SEG110 PCB STMT MUST PRECEDE SENSEG STMT.
Explanation: Self-explanatory.

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: Self-explanatory.

SEG190 PARENT OPERAND IS OMITTED OR INVALID.
Explanation: Self-explanatory.

SEG230 PROCOPT OPERAND IS INVALID.
Explanation: Self-explanatory.

SEG240 BYTES PARAMETER NOT IN RANGE 1 - 32768
Explanation: The maximum range of 32768 for segments has been exceeded.

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 applicabel 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. Fldname 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 - 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 through 32767.

VFLD155 'START' FIELD NOT PREVIOUSLY DEFINED
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.

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 designated alphabetic characters (X, H, F, C, Z, P, E, D or L) substituted for 't'.

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 '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 that may not be overridden.

VFLD190 BYTES PARAMETER NOT IN RANGE 1 - 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=prog parameter, 'prog' is the name of the module in the DOS/VS core image library that is given control whenever this field is retrieved or stored. It must be a 1 - 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 't' 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 EITHER RTNAME OR VALUE PARAMETER REQUIRED

Explanation: The parameter VALUE=value or RTNAME=prog is required in the statement.

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 DOS/VS job. Therefore, in addition to the messages listed in this chapter, other messages are issued by DOS/VS if error conditions are found during the processing of DOS/VS job control statements.

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 2-alphameric 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 CONFIG STATEMENT ALLOWED, IGNORED

Explanation: There may be only one DLZACT TYPE=CONFIG statement for each DL/I ACT generation.

DLZ1005 MAXTASK OPERAND MISSING, 10 DEFAULTED FOR MAX TASK 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 CONFIG STATEMENT MUST PRECEDE FIRST ACT

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 SLC NAME GT 8 CHARACTERS - TRUNCATED.

Explanation: The phase name of the storage layout control specified in the SLC operand of the DLZACT TYPE=CONFIG macro instruction exceeded 8 alphameric 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 PGMNAME PARAMETER TOO LONG, PGM IGNORED

Explanation: The application program name specified in the PGMNAME operand of the DLZACT TYPE=PROGRAM macro instruction exceeds 8 alphameric 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.

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.

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.

DLZ1019 PSBNAME name EXCEEDS 7 CHARS, ENTRY IGNORED

Explanation: The PSB name specified in the PSBNAME operand of the DLZACT TYPE=PROGRAM macro instruction exceeded 7 alphameric characters. The macro instruction is ignored.

DLZ1020 NUCLEUS GENERATION ERROR, GENERATION TERM.

Explanation: A severe error condition occurred that caused nucleus generation processing to terminate. The error condition is given in the preceding message(s).

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.

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).

DLZ1023 PASSWORD TRUNCATED TO 8 CHARACTERS

Explanation: The password specified in the PASS operand of the DLZACT TYPE=CONFIG macro instruction exceeded 8 alphameric characters. It was truncated to 8 characters.

DLZ1024 PSB COUNT 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.

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.

DLZ1026 MORE THAN 255 HDBFR STMTS. - 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.

DLZ1027 NBR. OF HDBFR STMTS EXCEEDS BFRPOOL SIZE - POOL WILL BE ADJUSTED AT EXEC TIME.

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.

DLZ1028 BUFFER SPECIFICATION INVALID IN HDBFR OPRD. SET TO number

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.

DLZ1029 DBDNAME PREVIOUSLY DEFINED - dbdname - 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.

DLZ1030 INVALID DBDNAME IN HDBFR OPRD. - dbdname 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 1-7 alphameric characters.

DLZ1031 NO VALID ENTRY IN TYPE=BUFFER, STMT 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 instruciton 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: (1) not specified, (2) invalid (it must be 1-7 alphameric characters), (3) previously specified in another statement, or (4) 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 STATEMENTS MUST BE PRECEDED BY TYPE=PROGRAM STMT, STMT IGNORED

Explanation: A TYPE=BUFFER statement was encountered before any TYPE=PROGRAM statements. The TYPE=BUFFER statement is ignored.

DLZ1039 REMOTE=value INVALID IN TYPE=CONFIG STMT, 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 STMT OUT OF ORDER, 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 STMT NOT DEFINED IN TYPE=PROGRAM STMT, STMT 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 NAME GT 4 CHARACTERS IN TYPE=RPSB STMT, STMT IGNORED

Explanation: The SYSID parameter contained more than 4 characters. The TYPE=RPSB statement is ignored.

DLZ1045 SYSID PARAMETER MISSING IN TYPE=RPSB STMT. STMT IGNORED

Explanation: The SYSID parameter must be specified on a TYPE=RPSB statement. The statement is ignored.

DLZ1046 RNAME name GT 8 CHARACTERS IN TYPE=RPSB STMT, STMT IGNORED

Explanation: The RNAME parameter contained more than 8 characters. The TYPE=RPSB statement is ignored.

DLZ1047 PSB PARAMETER MISSING IN TYPE=RPSB STMT, STMT IGNORED

Explanation: A PSB name was not specified in the PSB parameter of the DLZACT TYPE=RPSB statement. The statement is ignored.

DLZ1048 LANG=value INVALID IN TYPE=RPSB STMT, PARAMETER IGNORED

Explanation: The LANG parameter on the TYPE=RPSB statement must have a value = PL/I if the PSB type is PL/I. Other values are ignored.

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).

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: DL/I Status Codes

Note: When the first character of any of the following status codes is an N, this indicates that an error occurred during an internal DL/I call issued by indexing maintenance.

AB **Error in call.**
Explanation: On a data base call, segment I/O area is required but was not specified in the call.

Action: Correct the program.

AC **Error in call.**
Explanation: SSA(s) contains an error in hierarchical sequence.
Possible causes:

1. No segment name equal to that specified in SSA found within scope of this PCB.
2. Level at which this SSA appears is out of sequence with that specified by the PCB.
3. Two segments of the same level are specified in the same call.

Action: Correct the program.

AD **Error in call.**
Explanation: An invalid function parameter was supplied or a checkpoint call was issued while the task was scheduled to a data base on a remote system.

Action: Correct the program.

AH **Error in call.**
Explanation: No SSA(s) was specified in call. Call type requires at least 1 SSA; none was specified.

Action: Specify SSA in call. Correct the program.

AI **I/O, system, or user error.**
Explanation: Open error.
Possible causes:

1. Error in DOS/VS job control cards.
2. Data base being opened for other than load mode and has not been loaded.

Action: Check job control cards; ensure that filename is the same as the name specified on the DATASET card of the DBD. The segment name area in the PCB has the filename of the file which could not be opened.

AJ **Error in call.**
Explanation: SSA qualification format was invalid.
Possible causes:

1. Invalid relational operators
2. Missing right parenthesis
3. DLET call has qualified SSAs
4. REPL call has qualified SSAs
5. ISRT call has the last SSA qualified
6. Field in the SSA has the wrong length
7. Invalid command codes.

Action: Correct the program.

AK **Error in call.**
Explanation: The field name in the SSA parameter in a call statement does not correspond to any field name specified in the DBD.

Action: Correct the program.

AM **Error in call.**
Explanation: Call function not compatible with processing option or segment sensitivity.

Action: Correct the program, PSB, or system definition.
Possible causes:

1. Processing option of L and call function is not insert.
2. DLET. REPL. or ISRT call without corresponding segment sensitivity.
3. DLET or ISRT call for the root of a secondary data structure or any of its physical parent segment types in the physical data base.
4. A secondary index is processed as a data base itself and attempts are made to delete or insert a segment or to replace system-maintained fields.
5. Command code D used for a path retrieval call without path sensitivity.
6. An attempt was made to unload a secondary index using the HD unload utility.

AO **I/O error.**
Explanation: There is a SAM or VSAM I/O error.

Action: Check and correct.

DA **Error in call.**
Explanation: Segment key field has been changed on a REPL call.

Action: Correct.

DJ **Error in call.**
Explanation: No previous successful GET HOLD call.

Action: Check and correct.

DX **Error in call.**
Explanation: Violated delete rule. Review delete rule in SEGM statement of DBDGEN in Chapter 2 of the *DL/I DOS/VS Utilities and Guide for the System Programmer.*

Action: Correct the program.

GA **Call is completed.**
Explanation: Crossed hierarchical boundary into higher level. (See "Call Functions" in the *DL/I DOS/VS Application Programming Reference Manual.* This status code is returned on unqualified GN calls only.

GB **Call is not completed.**
Explanation: This is the end of the file; the position beyond the last segment is reached.

GE **Call is not completed.**
Explanation: Segment has not been found (get call). For an ISRT call, the parent of the segment to be inserted was not found.

GK **Call is completed.**
Explanation: Different segment type at sáme level returned. This status code is returned on unqualified GN calls only.

GP **Error in call.**
Explanation: No parent established (or parent deleted) for this GNP call, or the requested segment level is not lower than the parent level.

II Call is not completed.

Explanation: The segment that the user tried to insert already exists in the data base.

Possible causes:

- 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.
- For unique segment support, another segment under the same parent at the same type level exists.

Action: Correct the error.

IX Call is not completed.

Explanation: Violated the insert rule. Review the insert rule in SEGM statement of DBDGEN in Chapter 2 of the *DL/I DOS/VS Utilities and Guide for the System Programmer.*

Possible causes:

- 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 user I/O area, the key in the logical parent does not match the corresponding key in the concatenated key in the logical child.

Action: Correct the program.

KA Numeric truncation error.

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.

Action: Correct the program.

KB Character truncation error.

Explanation: During automatic length conversion of a character field the 'to' field was not large enough to contain all the nonblank characters moved from the 'from' field. The field is moved left justified.

Action: Correct the program.

KC Invalid packed decimal or zoned decimal format.

Explanation: During automatic conversion, a 'from' field character was encountered that is not a valid packed decimal or zoned decimal character.

Action: Correct the program.

KD Type conflict for conversion.

Explanation: A field-to-field conversion was requested that was not supported by DL/I.

Action: Do conversion in field exit routine.

KE NOREPL violation.

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.

Action: Correct the program or specify REPLACE=YES.

LB Call is not completed.

Explanation: The segment that the user tried to load already exists in the data base.

Possible causes:

- 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.
- For unique segment support, another segment under the same parent at the same type level exists.

Action: Correct the error.

LC Call is not completed.

Explanation: Key field of segments is out of sequence.

Action: Check and correct.

LD Call is not completed.

Explanation: No parent has been loaded for this segment.

Action: Check and correct.

LE Call is not completed.

Explanation: Sequence of sibling segments is not the same as the sequence in the DBD.

Action: Check and correct.

NA Call is not completed.

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.

Action: Correct the program.

NE Error in some previous insert call or system error. The INSERT, DELETE, or REPLACE call was completed, as if the NE status code were a warning.

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:

- During some previous insert call 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.
- Some error had occurred during reorganization of the data base.

Action:

- For DLET call, none. The index source segment which produced the duplicate key is now removed.
- For REPL call, user determined. For example, delete the segment and reinsert it with proper search and subsequence data.

NI VSAM open error or duplicate key for index data base.

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:

- Error in DOS/VS job control statements.
- Control interval size, keylength, 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.

Action: 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:

1. 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.
2. 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: An I/O error occurring during processing of an index, either in the index or indexed data base.

Action: Check and correct.

RX Error in call.

Explanation: Violated replace rule. Review replace rule in SEGM statement of DBDGEN in chapter 2 of the *DL/I DOS/VS Utilities and Guide for the System Programmer.*

Action: Correct the program.

V1 Program error.

Explanation: Invalid length for variable length segment. The LL field of the variable length segment is either too large or too small. The length of the segment must be equal or less than the maximum length specified in the DBD. The length must be long enough to include the entire sequence field.

Action: Correct the 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.

Action: Check and correct.

XH Call is not completed.

Explanation: Data base logging was inactive during checkpoint (CHKP) call processing.

Action: Ensure that data base logging is active during checkpoint call processing.

bb Call completed.

Explanation: Your call was completed!

Action: Proceed!

SH12-5414-6

DL/I DOS/VS Messages and Codes Printed in U.S.A. SH12-5414-6

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N. Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N. Y., U. S. A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N. Y., U. S. A. 10601

Data Language/I Disk Operating System/Virtual Storage
(DL/I DOS/VS) Messages and Codes
Order No. SH12-5414-6

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

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

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

	Yes	*No*
• Does the publication meet your needs?	□	□
• Did you find the material:		
Easy to read and understand?	□	□
Organized for convenient use?	□	□
Complete?	□	□
Well illustrated?	□	□
Written for your technical level?	□	□

- What is your occupation? ______________________________

- How do you use this publication:

As an introduction to the subject?	□	As an instructor in class?	□
For advanced knowledge of the subject?	□	As a student in class?	□
To learn about operating procedures?	□	As a reference manual?	□

Your comments:

If you would like a reply, please supply your name and address on the reverse side of this form.

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

SH12-5414-6

Reader's Comment Form

Cut or Fold Along Line

Fold and Tape Please Do Not Staple Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 10 ENDICOTT, N.Y. U.S.A.

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department G60
P. O. Box 6
Endicott, New York 13760

Fold Fold

If you would like a reply, *please print:*

Your Name ______________________________

Company Name ______________________ *Department* __________

Street Address ______________________________

City ______________________________

State ______________________ *Zip Code* __________

IBM Branch Office serving you ______________________________

IBM®

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N. Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N. Y., U. S. A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N. Y., U. S. A. 10601

DL/I DOS/VS Messages and Codes Printed in U.S.A. SH12-5414-6