software AG

Adabas Bridge for DL/1

Messages and Codes

Version 2.3.1

June 2014

Adabas Bridge for DL/I

Table of Contents

Messages and Codes

This documentation provides an overview of the Messages and Codes for the Adabas Bridge for DL/I.

The following topics are covered:

Introduction

Error Handling

Problem Analysis

Error Messages and Codes

Appendix A - Glossary of ADL Terms

1 Introduction

This documentation provides you with all information related to error situations in the ADL environment.

The installation of the ADL system is described in the ADL Installation documentation . The conversion of DL/I data bases is described in the ADL Conversion documentation. Finally, the installation and operation of the `CALLDLI` and Consistency Interfaces is covered by the ADL Interfaces documentation .

In addition to a list of error messages issued by ADL, the documentation provides information on the general error handling procedures of ADL and in particular the error handling of the `CALLDLI` and the Consistency Interfaces. The section `Problem Analysis` is intended to help you to cope with error situations.

For your convenience, a glossary of terms used in the ADL documentation is included.

This chapter covers the following topics:

Other documentations You May Need

The following Software AG publications may be useful when installing and operating the ADL Interface:

- *Adabas Utilities* documentation
- *Adabas Operations* documentation
- *Adabas Messages and Codes* documentation
- *Adabas Reference Data* and *Adabas DBA Reference Data* documentation.

For a complete list of Software AG documentation, refer to Software AG's **Empower** web site. If you do not have an Empower user ID and password yet, you will find instructions for registering on this site (free for customers with maintenance contracts).

Documentation Related to non-SAG Products

The documentation mentioned below might be of interest and helpful for the installation and operation of ADL.

For z/OS users:

- IMS/VS Application Programming
- IMS/VS Application Programming for CICS/VS Users
- IMS/VS Utilities Reference documentation

- CICS TS Installation Guide
- CICS TS Operations and Utilities Guide
- CICS TS Resource Definition Guide

For z/VSE users:

- DL/I DOS/VS Guide for New Users
- DL/I DOS/VS Application Programming: CALL and RQDLI Interface
- DL/I DOS/VS Application Programming: High Level Programming Interface
- DL/I DOS/VS Utilities and Guide for the System Programmer
- DL/I DOS/VS Resource Definition and Utilities
- CICS TS Installation Guide
- CICS TS Operations and Utilities Guide
- CICS TS Resource Definition Guide

2 Error Handling

This chapter covers the following topics:

General Error Handling

ADL Response Codes

Whenever ADL detects an error situation, it issues an error or warning message. Each individual message is identified by a unique number in the range from 1 to 1500. The *Error Messages and Codes* documentation lists all messages in numerical sequence together with an explanation and recommended action.

Note that error messages with numbers in the range from 1 to 255 are reserved for and identical to Adabas response codes. A more detailed explanation of these can be found in the *Adabas Messages and Codes* documentation.

Abnormal Endings Caused by ADL

If an unrecoverable error is detected, ADL terminates the user program abnormally. In particular, batch programs are abended with the user completion code 700, or the JDUMP macro in z/OS and z/VSE respectively. IMS/TP applications are abended with the user completion code 700.

In some cases, ADL terminates batch programs abnormally with a system completion code "0C1" (operation exception). These errors can easily be identified by the PSW pointing to a character string like:

```
-> INVALID PCB NUMBER
```

or

```
-> ATTEMPT TO ACCESS NON CONVERTED PCB
```

These error situations, however, are rare and should not occur under normal circumstances.

CICS applications are terminated abnormally either with an abend code DAZA, accompanied by an ADL error message, or any other abend code in the form DAZx where x may be "B", "C", "D" or a number between 1 and 9. These abend codes are explained in the section *Error Messages and Codes*.

Adabas Response Codes

ADL treats Adabas response codes in two different ways. The first class of Adabas response codes is considered to be unrecoverable and thus leads to an abnormal termination of the user application. An example of an unrecoverable Adabas response code is the response code 148 (database not active).

The second class of Adabas response codes is considered to be recoverable. In particular, these are the response codes 3 (end-of-file or end-of-list), 9 (time limit has been exceeded), 145 (the hold queue is full) and 198 (duplicate value for unique descriptor).

System Request Failures

Failures during system requests from ADL (like `GETMAIN`, sequential I/O etc.) are considered to be unrecoverable and generally cause an abnormal termination of the application (see above in this section). You should try to determine the cause of the error. Most of these are related to improperly defined sequential files or insufficient region sizes.

Internal Errors

Software AG tries to minimize potential sources of internal errors like data exceptions, buffer overflows etc. Thus, most of the internal errors can be solved by adjusting the relevant parameters, for example buffer sizes. Should there be any internal error which requires some action by Software AG, this is clearly marked in the section *Error Messages and Codes* in the "`Action:`" clause of the particular error message.

Error Handling of the CALLDLI Interface

Adabas Response Codes

Due to the fundamental differences between the two data base systems, not all Adabas response codes can be translated into DL/I status codes. ADL transforms some of the recoverable Adabas response codes into DL/I status codes.

The Adabas response code 3 normally results in a DL/I status code "`GB`" (end of dataset) or "`GE`" (segment not found) being returned to the application. The response code 198 results in a DL/I status code "II" (segment to insert already exists in the data base).

The Adabas response code 9 normally occurs in online applications only. The typical reason for this is that the user at a particular terminal has been inactive for a period of time longer than the Adabas time limit. If ADL finds that no data has been backed out by Adabas, it simply re-issues the command without any effect on the application. Otherwise it abnormally terminates the transaction, causing the online system to back out its resources as well.

If the `RETRY` parameter is set to `"WAIT"`, ADL receives a response code 145 only when the Adabas hold queue is full. In this case, Adabas attempts to issue the command a second time. If the second attempt fails as well, the application terminates abnormally. Should you observe that ADL terminates applications frequently for this reason, you should increase the size of the Adabas hold queue. If the `RETRY` parameter is set to a numeric value, ADL terminates the application abnormally if a required record is in hold status. See the section ADL Parameter Module in the *ADL Installation* documentation for more details on the `RETRY` parameter.

DL/I Status Codes

The ADL `CALLDLI` Interface generates status codes which are identical to and have the same meaning as those of DL/I. This is in particular true for all status codes which are related to the contents of the data base (like `"GB"`, `"GE"`, `"GK"` etc.) or caused by invalid call or segment qualification formats (like `"AM"` or `"AJ"`).

There are, however, a couple of DL/I status codes which are strongly related to the physical design of DL/I. Examples for these are `"XD"` (error during data base buffer write) and `"XH"` (data base logging not active). Since these do not correspond to any Adabas response code, they will never be generated by ADL.

The meaning of the DL/I status codes can be viewed in the ADL Online Services with the "Messages and Codes" function.

Error Handling of the Consistency Interface

Consistency Response Codes

If a data base request originating from a Natural program or an Adabas direct call does not fulfill the rules and conditions required for the integrity of the database, the Consistency Interface returns the response code 216 in the response code field of the Adabas command block. In Natural programs, this corresponds to the `*ERROR-NR 3216`.

If this error occurs, the ADL Consistency Interface provides a more detailed error message on request. The corresponding error codes and message are explained in the section *Error Messages and Codes*. The section *How to Retrieve an Error Code and Message* below describes how an application program may obtain these comprehensive error messages.

How to Retrieve an Error Code and Message

Natural programs may retrieve the full ADL error message by a "`CALLNAT`" to the ADL-supplied Natural subprogram `ADLERROR`, supplying an 80 character string as output field.

Example:

```
IF *ERROR-NR = 3216
    CALLNAT 'ADLERROR' #ERRMES
END-IF
```

The layout of the error message is

```
ADL xxxx - error text
```

where "`xxxx`" is the actual error number. The meanings of the individual error numbers and messages are explained in the section *Error Messages and Codes* together with a recommended action for each error.

Programs using Adabas direct calls may retrieve an ADL Consistency Interface error message by issuing an Adabas "`S1`" call as shown below:

```
Command Code        : S1
Command ID          : blank or binary zero
File Number         : filled in by ADL
ISN Lower Limit     : zero
Command Option1     : blank
Command Option2     : blank
FB length           : 26 bytes minimum
RB length           : 80 bytes minimum
SB length           : 3 bytes minimum
VB length           : 2 bytes minimum
IB length           : not used
Format Buffer       : must contain CL8'ADLERROR'
Record Buffer       : on return contains the ADL error message
Search Buffer       : filled in by ADL
Value Buffer        : filled in by ADL (error number in binary format)
```

Any fields of the Adabas control block not mentioned above are not used.

3   **Problem Analysis**

The information in this section enables you to react to and in many cases solve ADL related problems. It also tells you which information is required by Software AG representatives to effectively assist you and how to obtain this required information.

In general, when you request Software AG's technical support you should be able to specify which version of operating system, TP monitor, Adabas and ADL is installed at your site.

This chapter covers the following topics:

Failure of DL/I Calls

A DL/I call is considered a "failure" if the result of the call is not what the application expects. Program abends during DL/I calls are discussed in a further paragraph below.

A failure of a DL/I call could for example be

- an invalid status code returned in the PCB (or DIB)
- invalid or no data returned in the user I/O area
- invalid or no data returned in the key feedback area
- the segment requested is not retrieved but is in the data base.

In order to solve the problem of a DL/I call failure

- make sure that the data base was correctly loaded
- you may use the ADL supplied `CALLDLI` test program, `DAZZLER`, to verify the contents of the data base (the `DAZZLER` program is described in the ADL Interfaces documentation).
- make sure that the DL/I call is issued correctly
- make sure that the `PSB` and `DBD` definitions are correct and have been converted successfully and without errors
- with the help of the ADL Trace Facility, find and document the DL/I call which caused the problem.

The ADL Trace Facility, which is described in the ADL Interfaces documentation , will help you to find out exactly what is wrong with the call and to document the problem.

When calling Software AG technical support, you should have the listings of the ADL Trace Facility and the `PSB` and `DBD` sources available.

Failure of Adabas Calls

An Adabas call is considered a "failure" if the result of the call is not what the application expects. Program abends during Adabas calls are discussed in another paragraph below.

A failure of an Adabas call could be:

- the call has received a response code which it should not receive
- as a result of the call, invalid data is stored or replaced

You should use the ADL Trace Facility, described in the ADL Interfaces documentation , to isolate the problem.

When calling Software AG technical support, you should have available the listings of the ADL Trace Facility, the source definition of the DBD related to the ADL file and the Adabas file description table of the ADL file.

Abnormal Termination of Application Programs

Abnormal program terminations inside the ADL modules can be identified by the program status word (PSW) pointing to an address inside of one of the ADL modules. All ADL module names start with the characters "DAZ".

Most of the abnormal program terminations are accompanied by one or more ADL error messages printed on the file DAZOUT1 (batch) or routed to the user's terminal and the master console (CICS). These messages are explained in the section *Error Messages and Codes* below, together with the recommended action to be taken.

In rare cases, the PSW may point to a character string inside of one ADL modules. This character string should provide you with sufficient information to correct the error. If not, please call your Software AG technical representative.

In general, when calling Software AG for technical support, you should be able to provide the following information:

- the full ADL error message, if any, accompanying the abnormal program termination
- the version, release and SM level of ADL installed, toGETher with any additional ZAP applied
- in the case of DL/I applications, the mode (call or command level) and lanGUage (COBOL, PL/I, BAL) of the program
- the formatted dump of the abending application program and (CICS only) the ADL tables (as described in Maintaining the ADL Interfaces under CICS, section *ADL Online Services* in the *ADL Interfaces* documentation).

- a description of the data base calls issued by the application program (the ADL Trace Facility may help you to find out and to document which data base calls have been issued by the program prior to the abend)

- the DBD and PSB definition sources used by the application program

- the Adabas file description tables of the ADL files related to the problem

CICS Trace Table Entries

ADL adds entries to the CICS trace table for each data base request, scheduling and termination call from DL/I applications under CICS. These entries are restricted to calls which are actually served by ADL.

You will find the following trace table entries from ADL. For an explanation on how to obtain, format and interpret the CICS trace table, please see the related CICS documentation.

Trace ID	Field A	Field B	Remarks
X'C0'	call function	origin	all call types
X'C1'	PSBNAME	PSBNAME	scheduling call
X'C1'	A(PCB)	A(I/O area)	data base call

4   **Error Messages and Codes**

This chapter covers the following topics:

Introduction

This section contains explanations of messages and codes specific to ADL, plus any recommended action. Note that the ADL messages and codes can also be retrieved online by the ADL Online Services.

The error numbers 1 - 255 are reserved for and are identical to Adabas response codes. For this reason, only those Adabas response codes which are of particular significance to the ADL are mentioned here. A full list is provided in the *Adabas Messages and Codes* documentation .

The messages are reproduced in the form in which they actually appear, i.e. in mixed upper and lower case. Variable output information is represented by lower case "x" in inverted commas. All ADL nucleus messages have the following format:

```
Error ==>    nnnn  from ADL module DAZXXXXX/DAZYYYYY at address aaaaaa
ADLnnnn  message ..................


Warning   nnnn  from ADL module DAZXXXXX/DAZYYYYY at address aaaaaa
ADLnnnn  message ..................
```

where

nnnn	is the number of the message.
DAZXXXXX	is the name of the ADL internal module from which the message was generated.
DAZYYYYY	is the name of the ADL internal routine from which the message was generated.
aaaaa	is the offset within the ADL internal module in which the message was generated.

For convenience, only the lower message line is reproduced in the list below. The messages are arranged in ascending order according to their error numbers.

General ADL Messages and Codes

Number	Message	Cause	Action
ADL0017	Invalid file number	An Adabas call was issued for an Adabas file which was not in the data base. This was probably due to a DL/I call being issued against a DL/I data base which has not yet been converted.	Convert the DL/I data base or define the Adabas file. See the section *ADL Data Conversion Utility* in the *ADL Conversion* documentation for details.
ADL0113	Invalid ISN specified	An invalid ISN was specified for an Adabas call, probably as the result of incomplete data conversion.	Check that all `DAZELORE` runs were completed successfully.
ADL0145	Record not available	An attempt was made to access a record held by another user or a hold queue overflow occurred.	See the `RETRY` parameter in the *ADL Installation* documentation.
ADL0148	Adabas not active	The Adabas nucleus was not active.	Bring up Adabas.
ADL0216	Command rejected by user exit	If the ADL Consistency Interface is active, it has encountered an error situation.	Retrieve the comprehensive ADL Consistency Interface error message and follow the recommended actions for this. See the topic *Error Handling of the Consistency Interface* in the section *Error Handling* for more details on how to obtain the full error message.
ADL0281	Stack overflow	An overflow of the internal stack was detected in the ADL error handler. This message may occur directly after another error message.	See the recommended actions for the preceding error message. If no preceding error message is given, increase the stack size using the `STACK` parameter. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0282	Error Message could not be retrieved	This error only occurs if the error message for a previous error could not be retrieved from ADL directory file. A possible cause may be that the DBID and file number of the ADL directory file have not been defined correctly.	See the recommended actions for the preceding error number.
ADL0283	Error Message could not be retrieved	This error only occurs if the error message for a previous error could not be retrieved from the ADL directory file. In this case, only the number of the previous error was	See the recommended actions for the preceding error number.

Number	Message	Cause	Action
		output without the corresponding text.	
ADL0300	Stack overflow	The ADL internal routine stack is too small.	Increase the stack size using the `STACK` parameter. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0301	`DBD` area overflow	The buffer for the ADL `DBD` control blocks is too small.	Increase the buffer size using the `DBD` parameter. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details. If the error occurred during initialization under CICS, it was probably caused by an incorrectly generated `DAZDBD`. See the section *Generating the Runtime Control Tables* in the *ADL Interfaces* documentation .
ADL0302	Invalid `DBD` ECB encountered	A `DBD` control block read from the ADL directory file is invalid.	Check that all `DBD`s referenced by the `PSB` specified have been successfully run through the conversion process. See the section *ADL Conversion Utilities for DBDs and PSBs* in the *ADL Conversion* documentation for details. Under CICS, the `PSB` in error can be found by looking into `DAZOUT2`. It is the one, which would come alphabetically after the last one in the `PSB` list.
ADL0303	`PSB` area overflow	The buffer for the ADL `PSB` control blocks is too small.	Increase the buffer size using the `PSB` parameter. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details. If this error occurs during initialization under CICS, it is probably caused by an incorrectly generated `DAZBUF` table, for example the largest slot size is too small to hold all internal control blocks for at least one `PSB`. Use the `DAZSHINE` utility to find out the slot sizes needed to hold all `PSB`s that will be accessed in the online environment. (See the topic *Generating the Runtime Control Tables* in the *ADL Interfaces* documentation for details.)
ADL0304	Internal `DBD` table overflow	An internal table used as a `DBD` directory is too small.	Please contact your Software AG technical support representative.
ADL0305	Field name not found for sensitive field	A sensitive field not matching any of the fields in the referenced `DBD` has been encountered in the `PSB`.	Check the `DBD` and `PSB` in question and correct the error.

Number	Message	Cause	Action
ADL0306	Routine name specified for sensitive field	A sensitive field which has a compression routine specified has been encountered in the PSB. This facility is not supported by the current version of ADL.	Check the PSB in question and correct the error.
ADL0307	Secondary index not found	A secondary index DBD was encountered during initialization of the ADL control blocks, but no matching secondary index definition was found for it in the corresponding physical DBD.	Check the DBD in question and correct the error.
ADL0308	Logical relation table overflow	An internal table used for logical relations is too small.	Increase the table size using the NUMLR parameter (see the section *ADL Parameter Module* in the *ADL Installation* documentation for details).
ADL0309	Insufficient work area for ECBs	The buffer for the ADL control blocks read from the ADL directory file is too small.	Increase the buffer size using the EBUF parameter. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.

Number	Message	Cause	Action
ADL0310	Insufficient number of ECB directory entries	The directory for the ADL control blocks buffer read from the ADL directory file is too small.	Please contact your Software AG technical support representative.
ADL0311	Target segment not found	A secondary index was encountered during initialization of the ADL control blocks, but no target segment could be found for it.	Please contact your Software AG technical support representative.
ADL0312	Source segment not found	A secondary index was encountered during initialization of the ADL control blocks, but no source segment could be found for it.	Please contact your Software AG technical support representative.
ADL0313	SEARCH/SUBSEQUENCE/DUPLICATE DATA field not found	A secondary index was encountered during initialization of the ADL control blocks, but no SEARCH, SUBSEQUENCE or DUPLICATE DATA	Please contact your Software AG technical support representative.

Number	Message	Cause	Action
		fields could be found for it.	
ADL0314	Work area not big enough to store ECB	There is not enough space left in the work area to store an ADL control block on the ADL directory file.	Increase the size of the CBC utility work area using the UTI parameter. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0315	Logically deleted record not (uniquely) found	Internal error.	Please contact your Software AG technical support representative.
ADL0316	Error in ADA specification	The `ADA` parameter was specified incorrectly during the `DAZPRINT` utility run.	Check the `ADA` parameter and correct the error. See the section *Debugging Aids - ADL Trace Facility* in the *ADL Interfaces* documentation for details.
ADL0317	Invalid input statement	An invalid input statement was encountered during the `DAZPRINT` utility run.	Check the input parameters and correct the error. See the section *Debugging Aids - ADL Trace Facility* in the *ADL Interfaces* documentation for details.
ADL0318	DLI already specified	The `DLI` parameter has already been specified during the `DAZPRINT` utility run.	Check the `DLI` parameter and correct the error. See the section *Debugging Aids - ADL Trace Facility* in the *ADL Interfaces* documentation for details.
ADL0319	Error in DLI specification	The `DLI` parameter was specified incorrectly during the `DAZPRINT` utility run.	Check the `DLI` parameter and correct the error. See the section *Debugging Aids - ADL Trace Facility* in the *ADL Interfaces* documentation for details.
ADL0320	ADA already specified	The `ADA` parameter has already been specified during the `DAZPRINT` utility run.	Check the `ADA` parameter and correct the error. See the section *Debugging Aids - ADL Trace Facility* in the *ADL Interfaces* documentation for details.
ADL0321	Unexpected function code	Internal error.	Please contact your Software AG technical support representative.

Number	Message	Cause	Action
ADL0322	Error in RPL or bad parameter for PACKY	Internal error.	Please contact your Software AG technical support representative.
ADL0323	Z1 overflow	Internal error.	Please contact your Software AG technical support representative.
ADL0324	Target segment for SEXIC not found in PISSAS	Internal error.	Please contact your Software AG technical support representative.
ADL0325	Packed field length exceeds maximum	Internal error.	Please contact your Software AG technical support representative.
ADL0326	Feature not implemented	Internal error.	Please contact your Software AG technical support representative.
ADL0327	PSB name not found during memory allocation	The load module containing the PSB was given a different name than the PSB itself.	Correct the error and reconvert the PSB.
ADL0328	Binary number too large	Internal error.	Please contact your Software AG technical support representative.
ADL0329	PSB references files in >1 Adabas DB	The PSB references DBDs which have been converted but which specify more than one Adabas DBID. The current version of ADL does not support this facility.	Check the DBDs and correct the error.

Number	Message	Cause	Action
ADL0330	Overflow in DBD list or file list during open	Internal error.	Please contact your Software AG technical support representative.
ADL0331	Overflow in retain ISN list area	The area for the retain ISN list is too small.	Increase the size of the ISN list area or force ADL to issue automatic ET calls more often. Modify the IBSIZ or the ET parameter respectively.
ADL0332	Overflow in ISN buffer	The area for the ISNs is too small.	Increase the IBSIZ parameter.
ADL0333	Internal error	Internal error.	Please contact your Software AG technical support representative.

Number	Message	Cause	Action
ADL0334	Error on statement reader	An error has been detected during an attempt to read an input statement.	Check in JCL/JCS that the input data set/file was specified correctly.
ADL0335	Error on printer	An error has been detected during an attempt to write to the print data set/file.	Check in JCL/JCS that the printer data set/file was specified correctly.
ADL0336	Input statement is empty or starts with a blank	Self-explanatory.	Check the input statement and correct the error.
ADL0337	Routine not found	Internal error.	Please contact your Software AG technical support representative.
ADL0338	Undefined function keyword	The first word of an input statement is not an allowed function keyword for the CBC utility.	Correct the input statement and rerun the job or, where multiple functions are requested within one job, the last step in the generation process.
ADL0339	No parameter(s) found after function	No parameters were defined on the read input statement being processed.	Specify the parameter(s) required for the function being processed and rerun the job.
ADL0340	Continuation statement expected, EOF found	The preceding input statement ended with a comma. As a result, the CBC utility expected a continuation statement for the last function. However, an EOF occurred.	Remove the comma on the last input statement or add a continuation statement. Rerun the job or the last step of the job as appropriate.
ADL0341	Missing continuation from previous statement	A warning message. The preceding input statement ended with a comma. As a result, the CBC utility expected a continuation statement for the last function. It has now treated the last function as complete and tried to execute it. If all necessary parameters were specified, the function will have been executed as normal. If not, an error message for the missing parameter(s) will have been generated. If the function was executed, the present statement will be accepted as the next function.	Check that the conversion process was performed correctly. If not, correct your input statements. Rerun the job, making sure that you delete all control blocks before you do so.
ADL0342	Function not supported in this release	A function has been specified which is not supported by the current release of ADL, but is planned for a future release.	Rerun the job using only those functions which have been released.
ADL0343	Parameter not found	A parameter has been specified which is not allowed. A valid parameter has probably been misspelt.	Correct the input statements as necessary.

Number	Message	Cause	Action
ADL0344	Parameter value too long	The value given for a parameter exceeds the maximum permissible length allowed for it.	Correct the input statement accordingly.
ADL0345	Parameter not allowed for function	A parameter has been given which is not allowed for the function specified in the same or the preceding statement.	Check the input statement for parameters which are not allowed for the function being processed, correct the error and rerun the job or job step.
ADL0346	`NAME` parameter defined more than once	The `NAME` parameter is specified more than once within a single function.	Ensure that the `NAME` parameter is specified only once within the function.
ADL0347	Same name defined more than once	A segment may only be referenced once during generation of a single `DBD`.	Remove the second reference to the segment from the input statements.
ADL0348	Invalid value for `DBID` parameter	The value specified for the `DBID` parameter was invalid. The data base ID must be: ■ numeric ■ less than 65536	Correct the value for the `DBID` parameter.
ADL0349	Invalid value for `FNR` parameter	The value specified for the `FNR` parameter was invalid. The file number must be: ■ numeric ■ less than 65535	Correct the value for the `FNR` parameter.

Number	Message	Cause	Action
ADL0350	Invalid TYPE value: Only `ADA` or `DLI` allowed	Only the values `ADA` and `DLI` are allowed for the `TYPE` parameter, which specifies whether a `DBD` has been converted or not: `"ADA"` means that the `DBD` has been converted. `"DLI"` means that the `DBD` has not been converted. The default value is `ADA`.	Correct the input statement using allowed values for the `TYPE` parameter and rerun the job.
ADL0351	Invalid `CONSI` parameter specification	Self-explanatory.	Check the input statement, correct the error and rerun the job.
ADL0352	Invalid `LOGID` parameter specification	Self-explanatory.	Check the input statement, correct the error and rerun the job.

Number	Message	Cause	Action
ADL0353	DBID greater than or equal to 65536	Self-explanatory.	Check the input statement, correct the error and rerun the job.
ADL0354	`DBID` parameter defined more than once	The `DBID` parameter may only be specified once in a function.	Remove the second definition of the `DBID` parameter from the input statement.
ADL0355	`FNR` parameter defined more than once	The file number parameter may only be specified once in a function.	Remove the second definition of the `FNR` parameter from the input statement.
ADL0356	`LEN` parameter defined more than once	The length parameter may only be specified once in a function.	Remove the second definition of the `LEN` parameter from the input statement.
ADL0357	`POS` parameter defined more than once	The position parameter may only be specified once in a function.	Remove the second definition of the `POS` parameter from the input statement.
ADL0358	Invalid function code	Internal error.	Please contact your Software AG technical support representative.
ADL0359	Invalid value for `LEN` parameter	The length of a user-defined field for this parameter must be greater than zero and less than or equal to 255.	Correct the value of the `LEN` parameter.
ADL0360	Invalid value for `POS` parameter	The length of a user-defined field for this parameter must be greater than zero and less than or equal to 32767.	Correct the value of the `POS` parameter.
ADL0361	Only `YES` or `NO` allowed for `BACKW` parameter	The `BACKW` (backward) parameter may be used to determine whether a Z0 field should be filled in order to allow better performance during the insertion of child segments. The default is `NO`. However, a Z0 field may be generated and filled automatically, depending on the rules specified during `DBD` definition.	Specify a correct value for the `BACKW` parameter.
ADL0362	Preceding function keyword missing	Some functions require another preceding function keyword before they can be performed. For example, the `GENSEG` function requires a preceding `GENDBD` statement.	Define the required function keyword before of the statement causing the error.
ADL0363	File number missing for `DBD`	An extra file number must be specified for the `DBD`, even though file numbers are specified for all `DBD` segments.	Specify a file number (`FNR` parameter) for the `GENDBD` function.
ADL0364	Same parameter defined more than once - ignored	A parameter has been defined more than once in a function. This message is preceded by a printout of the parameter in question.	Remove the second definition of the parameter and rerun the job.

Number	Message	Cause	Action
ADL0365	Work area too small - increase and rerun	The work area specified by the `UTI` parameter at the start-up of the utility is not large enough to contain all the information required for conversion.	Increase the value specified for the `UTI` parameter and rerun the job.
ADL0366	Parameter value missing	A value was not defined for a parameter.	Define a value for the parameter and rerun the job.
ADL0367	Specified control block not found	No control block (`DBD` or `PSB`) with the name specified in the `NAME` parameter could be found.	Ensure that the output of the `DBD` or `PSB` assembled is linked as a load module (object format) of the library assigned to the job. Check that the name of the control block requested is correct. Rerun the job.
ADL0370	Name in loaded `DBD` or `PSB` not equal to module name	The name defined in the loaded `DBD` is not the same as the module name. The name of the module in the load library must be the same as that of the `DBD` or `PSB`.	Check that the name given in the `DBD` or `PSB` assembly is the same as that of the linked load module. Rerun the job.
ADL03701	Requested `ECB` not found	The control block for a secondary index `DBD` referenced in the `DBD` being processed could not be found.	Ensure that the output of the `DBD` or `PSB` assembled is linked as a load module (object format) of the library assigned to the job. Check that the name of the control block requested is correct. Rerun the job.
ADL0373	Specified segment not found in `DBD`	The segment referenced in a `GENSEG` function could not be found in the `DBD`.	Correct the value of the `NAME` parameter in the `GENSEG` statement or reassemble the `DBD` definition if the segment name there is incorrect.
ADL0374	Segment not defined in `DBD` sequence	The `GENSEG` functions must be defined in the same sequence as the segments defined in the `DBD`. Segments may, however, be skipped.	Put input statements in the order of the `DBD` segments and rerun the job.

Number	Message	Cause	Action
ADL0375	File number missing in `GENDBD` and `GENSEG` of root	This is a subsidiary error which occurs when no file number has been specified for the `DBD` (see **ADL0363**).	Specify a file number using the `FNR` parameter for the `DBD`, segment, or root of the segment.
ADL0376	Adabas name already in use or not allowed	The Adabas field name specified is already in use or not allowed (field names `E0` to `E9` are reserved by Adabas, and `Z0` to `Z9` by ADL).	Specify another, unused field name for the segment, or omit the `ADANAME` parameter, thus causing the Adabas field name to be automatically generated.

Number	Message	Cause	Action
ADL0378	CCPRT: invalid printer ID specified. See R1.	Internal error.	Please contact your Software AG technical support representative.
ADL0381	All available field names for this file already used	All field names for a file are reserved.	The only way to solve this problem is to distribute the segments of a `DBD` across different files.
ADL0382	Error on puncher	An error has been detected during an attempt to write to the punch data set/file.	Check that the JCL/JCS specifies the printer data set/file correctly.
ADL0383	Two `SEQUENCE` fields overlap each other	`SEQUENCE` fields may not overlap one another where more than one has been defined for a segment.	Correct the definition of the `DBD`, reassemble the `DBD`, and rerun the conversion job.
ADL0384	Segment not in `RLT`	Internal error.	Please contact your Software AG technical support representative.
ADL0386	`SEQUENCE` field of secondary index has incorrect `LEN`	The `SEQUENCE` field of the secondary index must be the same length as the sum of all `SEARCH` and `SUB SEQUENCE` fields defined for the secondary index.	Check that the definition of the indexed `DBD` is consistent with the definition given in the `INDEX DBD` field, then correct and reassemble the `DBD` as necessary, and rerun the utility.
ADL0387	Segment name in secondary index `DBD` invalid	The name of the only segment in the secondary index `DBD` must be the same as the name of the `LCHILD` that defines the secondary index in the physical `DBD`.	Check that the definition of the indexed `DBD` is consistent with the definition given in the `INDEX DBD` field, then correct and reassemble the `DBD` as necessary, and rerun the utility.
ADL0388	Name of `LCHILD` in secondary index `DBD` invalid	The segment name in the `LCHILD` of the `INDEX DBD` must be the same as that for the index target segment.	Check that the definition of the indexed `DBD` is consistent with the definition given in the `INDEX DBD` field, then correct and reassemble the `DBD` as necessary, and rerun the utility.
ADL0389	`DBD` name of `LCHILD` in secondary index `DBD` invalid	The `DBD` name in the `LCHILD` of the `INDEX DBD` must point to the indexed `DBD`.	Check that the definition of the indexed `DBD` is consistent with the definition that is given in the `INDEX DBD` field, then correct and reassemble the `DBD` as necessary, and rerun the utility.
ADL0390	String too long	Internal error.	Please contact your Software AG technical support representative.
ADL0391	Name of physical `DBD` in secondary index `DBD` invalid	The `DBD` name given in the `NAME` parameter of the `LCHILD` statement in the `INDEX DBD` must be that of the indexed `DBD`.	Check that the definition of the indexed `DBD` is consistent with the definition that given in the `INDEX DBD` field, then correct and reassemble the `DBD` as necessary, and rerun the utility.

Number	Message	Cause	Action
ADL0392	Secondary index name in secondary index DBD invalid	The name given in the INDEX parameter of the LCHILD statement in the INDEX DBD must be that of the XDFLD statement in the definition of the physical DBD.	Check that the definition of the indexed DBD is consistent with the definition that given in the INDEX DBD field, then correct and reassemble the DBD as necessary, and rerun the utility.
ADL0393	Unexpected number of records found	Internal error.	Please contact your Software AG technical support representative.
ADL0394	Referenced segment is not a logical child	The segment referenced as the paired segment of a virtually paired logical child is not a logical child.	Check that the definitions of both DBDs (i.e. the DBD of the virtual segment and the DBD of the paired segment) are consistent, then correct and reassemble the DBDs as necessary, and rerun the utility.
ADL0395	Duplicate field name in referenced DBD	The Adabas field name generated for the partial concatenated key (PCK) of a segment in a DBD referenced by the DBD being processed has already been used for a different purpose. This message is preceded by another message giving the name of the segment concerned and the duplicate Adabas field name.	The field names for the PCK field are generated automatically by the CBC utility and are normally unique for all segments within the DBDs active for a single conversion process. An error can only occur if a DBD references other DBDs which have been converted previously. Typical errors are the re-conversion of a DBD after new segments have been added, or where a particular DBD is referenced by more than one other DBD and any of the previous DBD conversion processes have generated the same PCK names. It is not possible to give a single set of instructions on how to solve this problem, but you will basically have to change the order of your conversion jobs in such a way that the DBD referenced by all other DBDs is converted first. Please contact your Software AG technical support representative for further information.
ADL0396	Invalid flag set in segment	Internal error. A segment is marked both as a virtual segment of a virtually paired logical relationship (flags SEGEC#BC and SEGEC#VC are set), and as a real segment or, alternatively, a Source 1 is not available.	Please contact your Software AG technical support representative.
ADL0397	Referenced segment must not be a virtual child	The segment referenced as the paired segment of a virtual segment of a virtually paired logical relationship must not be a virtual child.	Check that the DBD definitions for both segments of the logical relationship are consistent, then correct and reassemble the DBD as necessary, and rerun the utility.

Number	Message	Cause	Action
ADL0400	Max. seg. length < `PCK-LEN` of source segment's parent	The computed length of the virtual paired segment is not sufficient to contain the concatenated key of the source segment's parent.	Increase the length of the source segment, reassemble the `DBD` of the source segment and rerun the job.
ADL0401	Min. seg. length < `PCK-LEN` of source segment's parent	The computed minimum length of the virtual paired segment is not sufficient to contain the concatenated key of the source segment's parent. Variable length segments must be long enough to contain the concatenated key of the paired segment's parent.	Increase the length of the source segment, reassemble the `DBD` of the source segment and rerun the job.
ADL0402	Flag `#VC` may not be set to ON for this segment	Internal error.	Please contact your Software AG technical support representative.
ADL0403	Flag `#S1` may not be set to ON for this segment	Internal error.	Please contact your Software AG technical support representative.
ADL0405	`LCHILD` missing for segment in referenced DB	The segment referenced as a paired segment in a logical relationship needs a `LCHILD` corresponding to the actual segment.	Check that the definition of the paired logical relationship is correct in both `DBD`s, then correct any errors, reassemble the `DBD`s and rerun the job.
ADL0406	`LCHILD` must be paired to the bidir. logical child	The `LCHILD` of the paired segment which corresponds to the segment being tested does not point to it.	Check that the definition of the paired logical relationship is correct in both `DBD`s, then correct any errors, reassemble the `DBD`s and rerun the job.
ADL0407	Both segments of bidirectional relationship have children	Only one of the two segments of a bidirectional paired relationship may have children. The other will be defined as non-real by the CBC utility and must not have children. If both segments have children, the conversion process cannot be finished.	Redesign and reassemble the `DBD` and rerun the job.
ADL0408	Logical parent of paired segment incorrect	The logical parent of the paired segment must be identical to the parent of the real segment. This message is preceded by a display giving the names of the segment, the parent and the defined logical parent.	Check that the definition of the paired logical relationship is correct in both `DBD`s, then correct any errors, reassemble the `DBD`s and rerun the job.
ADL0409	Invalid file number for generated field name	Internal error.	Please contact your Software AG technical support representative.

Number	Message	Cause	Action
ADL0410	Insufficient cross reference space	The CBC utility saves information on all the `DBD`s referenced by the `DBD` being processed in an additional internal control block. This block is too small.	Increase the size of the `XREF` parameter in the `DBDGEN` statement during `DBD` generation, then reassemble the `DBD` and rerun the job.
ADL0411	Internal error during secondary index deletion	Internal error. All index `DBD`s referenced by a `DBD` are normally deleted at the beginning of `DBD` conversion and then automatically reconverted. Here, the index `DBD` was found but an error occurred during deletion.	Please contact your Software AG technical support representative.
ADL0412	Error during sequential output write	An error occurred during the ADL write to a sequential output data set/file using the `LOAD` function.	Check the JCL/JCS for the output data set/file.
ADL0413	Terminal ID already specified	The `TERM` parameter has been specified more than once.	Check the control statement for `DAZPRINT` and correct the error. See the section *Debugging Aids - ADL Trace Facility* in the *ADL Interfaces* documentation for details.
ADL0414	Task number already specified	The `TASK` parameter has been specified more than once.	Check the control statement for `DAZPRINT` and correct the error. See the section *Debugging Aids - ADL Trace Facility* in the *ADL Interfaces* documentation for details.
ADL0415	Invalid task number specified	An invalid task number has been specified.	Check the control statement for `DAZPRINT` and correct the error. See the section *Debugging Aids - ADL Trace Facility* in the *ADL Interfaces* documentation for details.
ADL0419	Unexpected `EOF` on input for ADL precompiler.	An unexpected `EOF` was detected on the program source input for the ADL precompiler.	Check that the file `DAZIN2` is defined correctly in the JCL/JCS. See the section *Precompiler for EXEC DLI Command* in the *ADL Interfaces* documentation for details.

Number	Message	Cause	Action
ADL0420	Error during read card input	An error occurred during the ADL read of the control for the ADL precompiler.	Check the JCL/JCS for the input data set/file. See the section *Precompiler for EXEC DLI Command* in the *ADL Interfaces* documentation for details.
ADL0421	Missing `LAN` parameter	The `LAN` parameter is missing.	Check the input parameters and correct the error. See the section *Precompiler for*

Number	Message	Cause	Action
			EXEC DLI Command in the *ADL Interfaces* documentation for details.
ADL0422	Illegal `LAN` parameter specification	The `LAN` parameter has not been specified correctly.	Check the `LAN` parameter and correct the error. See the section *Precompiler for EXEC DLI Command* in the *ADL Interfaces* documentation for details.
ADL0423	Illegal ADL precompiler parameters	The parameters for the ADL precompiler have not been specified correctly.	Check the input parameters and correct the error. See the section *Precompiler for EXEC DLI Command* in the *ADL Interfaces* documentation for details.
ADL0424	Error during read source input	An error occurred during the ADL read of the source program for the ADL precompiler.	Check the JCL/JCS for the input data set/file. See the section *Precompiler for EXEC DLI Command* in the *ADL Interfaces* documentation for details.
ADL0425	Illegal source program first statement	The first statement of the source program has to be a `*PROCESS` statement for PL/I, a CBL statement for COBOL or a `*ASM` statement for Assembler.	Check the source program and correct the error.
ADL0426	Illegal `QNUM` or `SNUM` parameter specification	One of the `QNUM` or `SNUM` parameters for the ADL precompiler was not assigned a numeric value.	Check the `QNUM` and `SNUM` parameters and correct the error. See the section *Precompiler for EXEC DLI Command* in the *ADL Interfaces* documentation for details.
ADL0427	Illegal `MARGIN` parameter specification	The `MARGIN` parameter for the ADL precompiler was not specified correctly.	Check the `MARGIN` parameter and correct the error. See the section *Precompiler for EXEC DLI Command* in the *ADL Interfaces* documentation for details.
ADL0428	Error during read trace input	An invalid record has been encountered while reading from the trace input file using the `DAZPRINT` utility.	Check that the input file `DAZIN4` has been correctly defined in the JCL/JCS and that trace data has been written to this file in a previous job or job step. See the section *Debugging Aids - ADL Trace Facility* in the *ADL Interfaces* documentation for details.
ADL0429	Error during write to print dataset.	An error occurred while writing to the printer dataset using the `DAZPRINT` utility.	Check that the output file `DAZOUT1` is correctly defined. See the section *Debugging Aids - ADL Trace Facility* in the *ADL Interfaces* documentation for details.
ADL0430	Invalid `MODE` parameter specified for `DAZPRINT`	An invalid keyword parameter `MODE` was specified for the `DAZPRINT` utility.	Check the `MODE` parameter and correct the error. See the section *Debugging Aids - ADL Trace Facility* in the *ADL Interfaces* documentation for details.
ADL0431	Invalid `'INPUT'` parameter specification	The `'INPUT'` parameter for the ADL precompiler was not specified correctly.	Check the `INPUT` parameter and correct the error. See the section *Precompiler for*

Number	Message	Cause	Action
			EXEC DLI Command in the *ADL Interfaces* documentation for details.

Number	Message	Cause	Action
ADL0450	Invalid checkpoint ID / parameters specified for XRST call	Either the error was caused by an invalid checkpoint ID specification (e.g. a time stamp or a non-existent ID), or the XRST call parameters are inconsistent with those of the CHKP call.	Check the application program and correct the error.
ADL0451	ROLL or ROLB call not permitted for this program	A ROLL or ROLB call may only be issued by programs running as Adabas transaction logic users.	Change the mode in which the program is run. See the sections *Batch Installation and Operation* and *Recovery and Restart Procedures* in the *ADL Interfaces* documentation for details.
ADL0452	Program terminated by ROLL call	A ROLL call was issued by the application program. All data base updates have been backed out and the application program was abended.	None.
ADL0454	Program terminated due to unrecoverable status code 'xx'	The application program received the indicated unrecoverable status code and was terminated.	Check the application program and correct the error.
ADL0455	Unrecoverable PCB found during restart of batch program	No restart data found.	Do not try to restart a program, which has ended normally.
ADL0456	PSB Internal Control Blocks not found on Directory File	Internal Error.	Please contact your Software AG technical support representative.
ADL0457	GETMAIN failed	Internal Error.	Please contact your Software AG technical support representative.
ADL0458	Error during CONVBIN	Internal Error.	Please contact your Software AG technical support representative.
ADL0460	Overflow in last-call save area	Cause: The last-call save area (LCS) is too small.	Increase the size of the LCS, or switch the LCS facility off. Refer to the description of the LCS parameter in the section *ADL Parameter Module* in the *ADL Installation* documentation for more details.
ADL0461	Exit-routine table overflow	The exit-routine table is too small.	Increase the number of possible entries in the exit-routine table using the NUMEXR parameter. See the section *ADL Parameter Module* in the *ADL Installation*

Number	Message	Cause	Action
			documentation for details. If the error occurred during initialization under CICS, it was probably caused by an incorrectly generated DAZDBD. See the section *Generating the Runtime Control Tables* in the *ADL Interfaces* documentation for details.
ADL0462	Exit-routine not found	An user-supplied index maintenance exit routine could not be loaded.	Check that the exit-routine name is correctly specified in the DBD source and that the corresponding module is in the load library.
ADL0474	Internal Error	Internal Error.	Please contact your Software AG technical support representative.

ADL DAZUEX06 / DAZUEXMI Messages

Number	Message	Cause	Action
ADL0501	Parameter not numeric	One of the numeric input parameters was given a non-numeric value.	Check the input parameter statement and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0502	Illegal MODE parameter specification	The MODE parameter was incorrectly specified.	Check the MODE parameter and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0503	Illegal LC parameter specification	The LC parameter was incorrectly specified.	Check the LC parameter and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0504	Illegal FNR parameter specification	The FNR parameter was incorrectly specified.	Check the FNR parameter and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0505	Segment name not found in table	Internal error: a segment was not found in an internal table. This might be caused by either a wrong user exit (UEX6) used for ADACMP, a wrong input file for ADACMP or a wrong LC= parameter specification for DAZELORE.	Please contact your Software AG technical support representative.

Number	Message	Cause	Action
ADL0506	Partial concatenated key name not found in table	Internal error: a partial concatenated key name was not found in an internal table.	Please contact your Software AG technical support representative.
ADL0507	Secondary index name not found in table	Internal error: a secondary index name was not found in an internal table.	Please contact your Software AG technical support representative.
ADL0510	Illegal `DBID` parameter specification	The `DBID` parameter was incorrectly specified.	Check the `DBID` parameter and correct the error. See the section *Migration to ADL 2.3 and Backward Migration* in the *ADL Installation* documentation for details.
ADL0511	File number not found in table	The file number specified in the `DAZUEXMI` run was not found in the User Exit 6 extension table.	Check that the User exit 6 `DAZUEXMI` is linked with the correct extension table. Check that the `FNR` parameter specified in the `DAZUEXMI` run corresponds to a file number in the table. See the section *Migration to ADL 2.3 and Bacjkward Migration* in the *ADL Installation* documentation for details.
ADL0512	Illegal SEQ parameter specification	The SEQ parameter was incorrectly specified.	Check the SEQ parameter and correct the error. See the section *Migration to ADL 2.3 and Backward Migratin* in the *ADL Installation* documentation for details.
ADL0513	4-byte parent ISN found. Backward migration not possible	At a backward migration a 4-byte parent ISN was found. A backward migration is therefore no longer possible.	Please contact your Software AG technical support representative.
ADL0514	Secondary index offsets out of sequence	Internal error.	Please contact your Software AG technical support representative.

DAZELORE Utility Messages

Number	Message	Cause	Action
ADL0601	Logical child segment not found in `PCB1`	The input given as the name of the logical child sensitive segment in `PCB1` could not be found in this PCB.	Check the `PSB` and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0602	Logical child segment not found in `PCB2`	No sensitive segment for the input logical child could be found in `PCB2`.	Check the `PSB` and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.

Number	Message	Cause	Action
ADL0603	Superfluous segments found in `PCB1`	`PCB1` should only contain sensitive segments for the logical child segment and its parents.	Check the `PSB` and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0604	`PCB1` doesn't reference a physical `DBD`	`PCB1` should be based on a physical `DBD`.	Check the `PSB` and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0605	Unexpected status code 'xx' from a `GN` call on `PCB2`	A status code other than " ", `"GB"`, or `"GA"` was received for a `GN` call when `PCB2` retrieved the concatenated segment.	Please contact your Software AG technical support representative.
ADL0606	Unexpected status code 'xx' from a `GN` call on `PCB2`	A status code other than " ", `"GB"`, or `"GA"` was received for a `GN` call when `PCB2` retrieved the concatenated segment.	Please contact your Software AG technical support representative.
ADL0607	Unexpected status code 'xx' from a `ISRT` call on `PCB3`	A status code other than " " was received for a `ISRT` call when `PCB3` inserted physically deleted segments.	Please contact your Software AG technical support representative.
ADL0608	Unexpected status code 'xx' from a `CNCT` call on `PCB1`	A status code other than " " or `"DP"` was received for a `CNCT` call with command code `X` when `PCB1` finished connecting a logical child to its logical parent.	Please contact your Software AG technical support representative.
ADL0609	Missing control input	No control input statement has been given.	Check the job and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0610	Unexpected status code 'xx' from a `CNCT` call on `PCB1`	A status code other than " " or `"DP"` was received for a `CNCT` call when `PCB1` connected a logical child to its logical parent.	Please contact your Software AG technical support representative.
ADL0611	`GETMAIN` failed	A `GETMAIN` to acquire storage for I/O areas was unsuccessful.	Increase the partition/region size.
ADL0612	Unexpected DP status code for DAZELORE procedure used	A logical child segment was encountered without a matching logical parent while the Simplified, Special or Turbo Procedures were being used. These procedures are not applicable to this database.	Rerun the database conversion process using the Standard Procedure for `DAZELORE`. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details
ADL0613	Invalid control statement	The name after the `LC` keyword was not followed by a blank or by `",MODE="`.	Check the input statement and correct the error. See the section *ADL Data*

Number	Message	Cause	Action
			Conversion Utilities in the *ADL Conversion* documentation for details.
ADL0614	Invalid logical child segment name	Invalid input after the `LC` keyword.	Check the input statement and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0615	Incorrect number of `SENSEG`s in `PCB3`	The number of sensitive segments in `PCB3` must be one greater than the number of parent segments of the destination parent segment.	Check the `PSB` and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0616	`PCB4` doesn't reference a logical `DBD`	`PCB4` must be based on a logical `DBD`.	Check the `PSB` and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0617	Input name is not a logical child segment	The segment specified in the input statement is not a logical child segment.	Check the input statement and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0618	`PCB1` should have path calls allowed	`PCB1` should be defined with `"PROCOPT=..P.."`	Check the `PSB` and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0619	Unexpected status code `xx` from a GN call on `PCB2`	A status code other than `" "`, `"GA"`, `"GK"`, or `"GB"` was received for a `GN` call on `PCB2` during reading of variable intersection data.	Please contact your Software AG technical support representative.

Number	Message	Cause	Action
ADL0620	At least 3 PCBs required for `MODE=STANDARD`	There must be a third (and in some cases a fourth) PCB when `DAZELORE` is running with `"MODE = STANDARD"`.	Check the `PSB` and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0621	`PCB2` doesn't reference a logical `DBD`	`PCB2` must be based on a logical `DBD`.	Check the `PSB` and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0622	LC segment in `PCB2` is not concatenated	The sensitive segment in `PCB2` for the logical child is not concatenated.	Check the `PSB` and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0623	Missing destination parent segments in `PCB2`	Not all parent segments of the destination parent have been specified in `PCB2`.	Check the `PSB` and correct the error. See the section *ADL Data Conversion*

Number	Message	Cause	Action
			Utilities in the *ADL Conversion* documentation for details.
ADL0624	Missing variable intersection data segments in `PCB2`	Not all variable intersection data segments have been specified in `PCB2`.	Check the `PSB` and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0625	Unexpected status code `'xx'` on a `CNCT` call on `PCB1`	A status code other than `" "` has been received on a `CNCT` call on `PCB1`.	Please contact your Software AG technical support representative.
ADL0626	Unexpected status code `'xx'` on an `ISRT` call on `PCB4`	A status code other than `" "` has been received on an `ISRT` call on `PCB4`.	Please contact your Software AG technical support representative.
ADL0627	`PCB3` does not reference a physical `DBD`	`PCB3` must be based on a physical `DBD`.	Check the `PSB` and correct the error. See the section *ADL Data Conversion Utilities>* in the *ADL Conversion* documentation for details.
ADL0628	`PCB3` should have path calls allowed	`PCB3` must be defined with "PROCOPT=..P.."	Check the `PSB` and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0629	Destination parent segment not found in `PCB3`	The destination parent segment could not be found in `PCB3`.	Check the `PSB` and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0630	Incorrect number of segments in `PCB4`	`PCB4` should only contain the logical child segment, all its parent segments and all variable intersection data segments.	Check the `PSB` and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0631	Logical child segment not found in `PCB4`	The logical child segment could not be found in `PCB4`.	Check the `PSB` and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0632	Logical child segment is not concatenated in `PCB4`	The sensitive segment in `PCB4` for the logical child is not concatenated.	Check the `PSB` and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0633	`PCB2` and `PCB4` are out of sync	The sensitive segments describing the variable intersection data segments are not in the same order in `PCB2` as in `PCB4`.	Check the `PSB` and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0634	Input parameter specified more than once	One of the input keyword parameters was specified more than once.	Check the input parameters and correct the error. See the section *ADL*

Number	Message	Cause	Action
			Data Conversion Utilities in the *ADL Conversion* documentation for details.
ADL0635	Numeric input not numeric	One of the numeric input parameters was assigned a non-numeric value.	Check the input parameters and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0636	Unexpected status code `xx` from a `GU` call on `PCB1`	A status code other than " " was received for a `GU` call on `PCB1` during an attempt at repositioning during a restart.	Please contact your Software AG technical support representative.
ADL0637	Unexpected status code `xx` on a `CHKP` call	A status code other than " " was received for a `CHKP` call.	Please contact your Software AG technical support representative.
ADL0638	Unexpected status code `xx` from a `GU` call on `PCB2`	A status code other than " " was received for a `GU` call on `PCB2` during an attempt at repositioning during a restart.	Please contact your Software AG technical support representative.
ADL0639	Unexpected status code `xx` from a `GU` call on `PCB4`	A status code other than " " was received for a `GU` call on `PCB4` during an attempt at repositioning during a restart.	Please contact your Software AG technical support representative.

Number	Message	Cause	Action
ADL0640	`NUMLC` specified without `INTER`	The `NUMLC` parameter was specified without the `INTER` parameter. This is not allowed.	Check the input parameters and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0641	TURBO requires a bidirectional logical child	The Turbo procedure can only be used for bidirectional logical relationships.	Use the Special procedure for uni-directional logical relationships. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0642	Non-real logical child for Special or Turbo procedure	The Special or Turbo procedure was used for a bidirectional logical relationship although the logical child segment specified was not the real logical child.	Check the input parameters and specify the paired logical child segment. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0643	Turbo with `RESTART` requires 8 bytes `KFB`	The key feedback area of `PCB1` must be at least 8 bytes long if the Turbo procedure is used with `RESTART=YES`.	Increase the key feedback area length, so that it is at least 8 bytes long. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0644	`NUMCP` or `INTER` requires `RESTART=YES`	The `NUMCP` and `INTER` parameters can only be specified for `RESTART=YES`.	Specify `RESTART=YES`. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.

Number	Message	Cause	Action
ADL0645	`MAXDPISN` is only allowed for `MODE=TURBO`	The `MAXDPISN` can only be specified for the Turbo procedure.	Check the input parameters and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0647	Not unique real path to LC for `MODE=TURBO`	The Turbo procedure can only be used, if all parents of the `LC` in the real path have a unique sequence field.	Use the Special procedure. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0648	Not unique logical path to `LC` for `MODE=TURBO`	The Turbo procedure can only be used, if all parents of the `LC` in the logical path have a unique sequence field.	Use the Special procedure. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.

Unload Utility (DAZUNDLI) Messages

Number	Message	Cause	Action
ADL0651	Unload `DBD` is neither a physical nor an index `DBD`	The `DBD` referenced by the input `PSB` must be a physical or index `DBD`.	Check the `PSB` and correct the error. See the section *ADL Data Conversion Utilities>* in the *ADL Conversion* documentation for details.
ADL0652	Unexpected segment type read	The GN on `PCB1` retrieved an unexpected segment type.	Please contact your Software AG technical support representative.
ADL0653	Unexpected status code `xx` from a GET call on `PCB1`	A status code other than " ", "GA", "GK","GE", or "GB" was received for a GN or GNP call on `PCB1`.	Please contact your Software AG technical support representative.
ADL0654	Unexpected status code `xx` from a LOAD call on `PCB2`	A status code other than " " was received for the LOAD call on `PCB2`.	Please contact your Software AG technical support representative.
ADL0655	Invalid control card	An invalid control card was given.	Check the control card and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0656	Invalid `MODE` parameter	The `MODE` parameter was incorrectly specified.	Check the `MODE` parameter and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0657	Maximum number of fields (64) exceeded	More than 64 field names have been specified.	Specify at most 64 field names.
ADL0658	Illegal or missing segment name	The SEGM= keyword must precede each field name	Check the SEGM parameter and correct the error. See the section *ADL Data*

Number	Message	Cause	Action
		specification and must be followed by a segment name up to 8 characters long.	*Conversion Utilities* in the *ADL Conversion* documentation for details .
ADL0659	Illegal or missing field name	The FIELD= keyword must follow each SEGM specification and must be followed by a field name up to 8 characters long.	Check the FIELD parameter and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0660	Invalid NUMROOT parameter	The NUMROOT parameter was incorrectly specified.	Check the NUMROOT parameter and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0661	Invalid NUMREC parameter	The NUMREC parameter was incorrectly specified.	Check the NUMREC parameter and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0662	Invalid START or END parameter	The START or END parameter was incorrectly specified.	Check the START and END parameters and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0663	Target of secondary index is not the root segment	The target of the secondary index, referenced by the input `PSB`, is not the root segment.	Check the `PSB` and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0664	Invalid `ROOTKEYS` parameter	The `ROOTKEYS` parameter was incorrectly specified.	Check the control cards and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0665	`ROOTKEYS` don't allow `` `START' `` or `` `END' `` specification	The `ROOTKEYS` parameter must not be specified with the `START` or `END` parameters.	Check the control cards and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0681	Unexpected status code `` `xx' `` from a GN call on `PCB1`	A status code other than " ", "GA", "GK", or "GB" was received for a GN call on `PCB1`.	Please contact your Software AG technical support representative.
ADL0682	Unexpected segment type read	The segment type read does not match the sensitive segments in `PCB1`.	Please contact your Software AG technical support representative.
ADL0683	Unexpected response code from write seq. file	Error while writing to the output sequential file.	Check the JCL/JCS and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0684	Unexpected response code from close seq. file	Error while closing the output sequential file.	Check the JCL/JCS and correct the error. See the section *ADL Data Conversion*

Number	Message	Cause	Action
			Utilities in the *ADL Conversion* documentation for details.
ADL0685	Unexpected response code from read seq. file	Error while reading the input sequential file.	Check the JCL/JCS and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0686	Unexpected status code `xx` from LOAD	A non-blank status code was received for a LOAD call to the ADL.	Please contact your Software AG technical support representative.

Batch Region Controller (DAZIFP) Messages

Number	Message	Cause	Action
ADL0701	Illegal or missing parameters	The mandatory parameters for `DAZIFP` were not specified correctly. There should be three positional parameters: `xxx,pgmname,psbname,` These may be followed by keyword parameters.	Check the input parameters and correct the error. See the section *Batch Installation and Operation* in the *ADL Interfaces* documentation for details.
ADL0702	Illegal or missing program name	The name of the application program was not specified correctly. The program name is the second positional parameter and may be up to 8 characters long.	Check the input parameter and correct the error. See the section *Batch Installation and Operation* in the *ADL Interfaces* documentation for details.
ADL0703	Illegal `PSB` name	The `PSB` name was not specified correctly. It is the third positional parameter and may be up to 8 characters long.	Check the input parameter and correct the error. See the section *Batch Installation and Operation* in the *ADL Interfaces* documentation for details.
ADL0704	Illegal `TRACE` parameter specification	The `TRACE` parameter was not specified correctly.	Check the `TRACE` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0705	Illegal start count	The part of the `TRACE` parameter defining the start count was not specified correctly.	Check the `TRACE` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.

Number	Message	Cause	Action
ADL0706	Parameter not numeric	One of the numeric parameters was not assigned a numeric value.	Check the input parameters and correct the error. See the section *Batch Installation and Operation* in the *ADL Interfaces* documentation for details.
ADL0707	Illegal DL/I call count	The part of the `TRACE` parameter defining the DL/I call count was not specified correctly.	Check the `TRACE` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0708	Illegal logical unit	The part of the `TRACE` parameter defining the logical unit was not specified correctly.	Check the `TRACE` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0709	Illegal `SQ` parameter specification	The `SQ` parameter was not specified correctly.	Check the `SQ` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0710	Illegal record length or block size	An illegal record length or block size was specified for the `SQ` or `FX` parameter.	Check the `SQ` and `FX` parameters and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0711	Illegal record format	An illegal record format was specified for the `SQ` parameter.	Check the `SQ` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0712	Illegal or missing mixed `MODE` parameters	The input parameters for mixed mode operation are missing or invalid.	Check the input parameters and correct the error. See the section *Batch Installation and Operation* in the *ADL Interfaces* documentation for details.
ADL0713	Illegal trace function	An illegal trace function was specified.	Check the `TRACE` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0714	Illegal `DBID` parameter specification	The `DBID` parameter was not specified correctly.	Check the `DBID` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0715	Missing or illegal DBID	The `DBID` parameter is missing or was not specified correctly.	Check the input parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0716	Illegal `FNR` parameter specification	The `FNR` parameter was not specified correctly.	Check the `FNR` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.

Number	Message	Cause	Action
ADL0717	Missing or illegal `FNR`	The `FNR` parameter is missing or was not specified correctly.	Check the input parameters and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0718	The `DAZPARM` module does not point to itself	The ADL nucleus (`DAZNUCU` in the case of a CBC utility run and `DAZNUCB` in the case of any other batch run) was loaded during initialization but found to be invalid. The nucleus was probably not linked properly.	Relink the ADL nucleus. See the section *z/OS Installation* or *z/VSE Installation* in the *ADL Installation* documentation for details.
ADL0719	Illegal routine name	An illegal routine name was specified for the `TRACE` parameter.	Check the `TRACE` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.

Number	Message	Cause	Action
ADL0720	Illegal routine count	An illegal routine count was specified for the `TRACE` parameter.	Check the `TRACE` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0721	Illegal `PLI` parameter specification	The `PLI` parameter was not specified correctly.	Check the `PLI` parameter and correct the error. See the section *Batch Installation and Operation* in the *ADL Interfaces* documentation for details.
ADL0722	Illegal `STACK` parameter specification	The `STACK` parameter was not specified correctly.	Check the `STACK` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0723	Illegal `EBUF` parameter specification	The `EBUF` parameter was not specified correctly.	Check the `EBUF` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0724	Illegal `DBD` parameter specification	The `DBD` parameter was not specified correctly.	Check the `DBD` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0725	Illegal `PSB` parameter specification	The `PSB` parameter was not specified correctly.	Check the `PSB` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0726	Illegal `ET` parameter specification	The `ET` parameter was not specified correctly.	Check the `ET` parameter and correct the error. See the section *ADL Parameter*

Number	Message	Cause	Action
			Module in the *ADL Installation* documentation for details.
ADL0727	Illegal `UTI` parameter specification	The `UTI` parameter was not specified correctly.	Check the `UTI` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0728	Incompatible versions	An internal error. The internal version numbers of the ADL modules do not match. The ADL supplied is invalid.	Please contact your Software AG technical support representative.
ADL0729	`DAZMIX` could not be loaded	The ADL module `DAZMIX` could not be loaded.	Check that the `DAZMIX` module is present in one of the load libraries.
ADL0730	Application program could not be loaded	The application program could not be loaded.	Check that the application program is present in one of the load libraries.
ADL0731	Nucleus could not be loaded	The ADL nucleus (`DAZNUCU` in the case of CBC utility runs, `DAZNUCP` in the case of ADL precompiler runs, and `DAZNUCB` in all other cases) could not be loaded.	Check that the ADL nucleus is present in one of the load libraries.
ADL0732	`PSB` name not found	The `PSB` specified as one of the input parameters could not be found in the ADL directory file.	Run the CBC utility for the `PSB`. See the section *ADL Conversion Utilities for DBDs and PSBs* in the *ADL Conversion* documentation for details.
ADL0733	Not all ECBs referenced by the `PSB` converted into ICBs	The `PSB` specified as one of the input parameters references a `DBD` which has probably not been run through the CBC utility successfully.	Check the `PSB` and the referenced `DBD`s, and complete the conversion. See the section *ADL Conversion Utilities for DBDs and PSBs* in the *ADL Conversion* documentation for details.
ADL0734	`PSB` for `DAZUNDLI` contains DL/I PCBs	The `PSB` specified as one of the input parameters for a `DAZUNDLI` run references at least one `DBD` which has been marked as non-converted.	Check the `PSB` and correct the error. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.
ADL0735	Illegal `PR` parameter specification	The `PR` parameter was not specified correctly.	Check the `PR` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0736	`GETMAIN` for `PSB` failed	The `GETMAIN` to get the work area for initializing the input `PSB` has failed.	Increase the partition/region size.
ADL0737	`DAZUNDLI PSB` must contain at least 2 PCBs	The `PSB` used for this `DAZUNDLI` run contains only 1 PCB.	Create a `PSB` with 2 PCBs. See the section *ADL Data Conversion Utilities* in the *ADL Conversion* documentation for details.

Number	Message	Cause	Action
ADL0738	Illegal `CPID` parameter specification	The `CPID` parameter was not specified correctly.	Check the `CPID` parameter and correct the error. See the section *Batch Installation and Operation* in the *ADL Interfaces* documentation for details.
ADL0739	Illegal checkpoint ID / time stamp	The checkpoint ID or time stamp was not specified correctly.	Check the `CPID` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.

Number	Message	Cause	Action
ADL0740	Illegal `IMSY` parameter specification	The `IMSY` parameter was not specified correctly.	Check the `IMSY` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0742	The table module `DAZPSB` could not be loaded	During execution of the `DAZSHINE` utility, or while running the `DAZIFP` initialization program in a message region (IMS/TP only), the table module `DAZPSB` could not be loaded.	Check that the table `DAZPSB` was created correctly and is in one of the load libraries. Confirm that the suffix specified for `DAZPSB` either by default or explicitly is correct. See the section *Generating the Runtime Control Tables* in the *ADL Interfaces* documentation for details.
ADL0743	Illegal `DBDSF` parameter specification	The `DBDSF` parameter was not specified correctly.	Check the `DBDSF` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0744	Illegal `PSBSF` parameter specification	The `PSBSF` parameter was not specified correctly.	Check the `PSBSF` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0745	Illegal `BUFSF` parameter specification	The `BUFSF` parameter was not specified correctly.	Check the `BUFSF` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0746	The table module `DAZDBD` could not be loaded	(IMS/TP only) During execution of the `DAZIFP` initialization program in a message region, the table module `DAZDBD` could not be loaded.	Ensure that the `DAZDBD` table was correctly created and is in one of the load libraries, and that the `DAZDBD` suffix (either default or explicit) is correct. See the section *Generating the Runtime Control Tables* in the *ADL Interfaces* documentation for more information.
ADL0747	The table module `DAZBUF` could not be loaded	(IMS/TP only) During execution of the `DAZIFP` initialization program in a message region, the table	Ensure that the `DAZBUF` table was correctly created and is in one of the load libraries, and that the `DAZBUF` suffix (either default or explicit) is correct. See the section

Number	Message	Cause	Action
		module `DAZBUF` could not be loaded.	*Generating the Runtime Control Tables* in the *ADL Interfaces* documentation for more information.
ADL0748	MPP program: ICBs could not be rolled in	The requested `PSB` ICBs did not fit into the largest `DAZBUF` buffer module slot.	Ensure that the buffer module was created correctly. See the section *Generating the Runtime Control Tables* in the *ADL Interfaces* documentation for more information.
ADL0749	Illegal `OPENRQ` parameter specification	The `OPENRQ` `ADARUN` parameter was not specified as either `YES` or `NO`.	Enter either `YES` when `OPEN` is required or `NO` when `OPEN` is not required. `OPENRQ=YES` must be specified if the Adabas `ADARUN` `OPENRQ` parameter was specified as (or defaulted to) `YES`.
ADL0750	Illegal `DUO` parameter specification	The `DUO` parameter was not specified correctly.	Check the `DUO` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0751	Illegal `RETRY` parameter specification	The `RETRY` parameter was not specified correctly.	Check the `RETRY` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0752	Illegal `LCS` parameter specification	The `LCS` (last-call save area) parameter was not specified correctly.	Check the `LCS` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0753	Illegal `CHKPMSG` parameter specification	The `CHKPMSG` (checkpoint message) parameter was not specified correctly.	Check the `CHKPMSG` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0754	Illegal `RBE` parameter specification	The RBE (record buffer extension) parameter was not specified correctly.	Check the `RBE` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0755	Illegal `MFT` parameter specification	The `MFT` (Multifetch Table) parameter was not specified correctly.	Check the `MFT` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0756	The `RBE` and `MFT` parameters are mutually exclusive	Only one of the `RBE` (record buffer extension) or `MFT` (Multifetch Table) parameter types must be specified.	Check the parameters and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0757	Illegal `LOAD` parameter specification	The `LOAD` parameter was not specified correctly.	Check the `LOAD` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.

Number	Message	Cause	Action
ADL0758	Illegal `LANG` parameter specification	The `LANG` (language) parameter was not specified correctly.	Check the `LANG` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0759	The ADL pre-load program `DAZMPL` could not be loaded	(IMS/TP only) During execution of the `DAZIFP` initialization program in a message region, the ADL pre-load program `DAZMPL` could not be loaded.	Ensure that the `DAZMPL` program is in one of the load libraries. See the section *IMS/TP Installation and Operation* in the *ADL Interfaces* documentation for details.
ADL0760	Illegal `FX` parameter specification	The `FX` parameter was not specified correctly.	Check the `FX` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL0761	Illegal `PLILE` parameter specification	The `PLILE` parameter was not specified correctly.	Check the `PLILE` parameter and correct the error. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.

DAZSHINE Utility Messages

Number	Message	Cause	Action
ADL0800	Invalid or missing run parameter for `DAZSHINE`	Self explanatory.	Check the control card input for `DAZSHINE`. See the section *Generating the Runtime Control Tables* in the *ADL Interfaces* documentation for details.
ADL0801	Invalid `MODE` parameter for `DAZSHINE`	Self explanatory.	Check the `MODE` parameter for `DAZSHINE`. See the section *Generating the Runtime Control Tables* in the *ADL Interfaces* documentation for details.
ADL0802	Invalid `RANGE` parameter for `DAZSHINE`	Self explanatory.	Check the `RANGE` parameter for `DAZSHINE`. See the section *Generating the Runtime Control Tables* in the *ADL Interfaces* documentation for details.
ADL0803	Start parameter for `RANGE` 0 or larger than 4096	Self explanatory.	Check the `RANGE` parameter for `DAZSHINE`. See the section *Generating the Runtime Control Tables* in the *ADL Interfaces* documentation for details.
ADL0804	Step parameter for `RANGE` 0 or larger than 1024	Self explanatory.	Check the `RANGE` parameter for `DAZSHINE`. See the section *Generating the Runtime Control Tables* in the *ADL Interfaces* documentation for details.
ADL0805	`PSB` table missing or empty	The ADL table `DAZPSB` could not be loaded or contains no `PSB` definition.	Check that the `PSB` table was created correctly and is contained in the load library. See the section *Generating the Runtime Control Tables* in the *ADL Interfaces* documentation for details.

Number	Message	Cause	Action
ADL0806	An internal error has occurred	Internal error.	Please contact your Software AG technical support representative.
ADL0807	Error during write to file `DAZOUT4`	An error occurred while writing to file `DAZOUT4`.	Check that the file `DAZOUT4` is correctly defined in the JCL/JCS. See the section *Generating the Runtime Control Tables* in the *ADL Interfaces* documentation for details.
ADL0808	`DAZSHINE` - Internal Error	Internal error.	Please contact your Software AG technical support representative.
ADL0809	No `PSB` found in directory	None of the `PSB`s specified in the `DAZPSB` table were found in the ADL directory.	Check if the `DBID` and `FNR` of the ADL directory file are specified. Convert at least one of the `PSB`s of the `DAZPSB` table.
ADL0810	`DAZSHINE` - Overflow in `DBID/FNR` Table	Internal error.	Please contact your Software AG technical support representative.

CICS Environment Messages

The following messages are written either to the operator console or to the screen of the ADL supplied CICS transactions. They can also be displayed when you maintain the ADL Interfaces with the ADL Online Services.

Number	Message	Cause	Action
ADL0901	Invalid environment encountered	An unsupported environment was encountered. This message may apply either to CICS or to a particular version of an operating system.	Please contact your Software AG technical support representative.
ADL0902	Requested function is not supported	Self-explanatory.	Enter a valid function.
ADL0903	Internal Error occurred	Internal error.	Please contact your Software AG technical support representative.
ADL0904	Unable to load ADL Nucleus	The ADL nucleus `DAZNUCC` could not be loaded.	Check that `DAZNUCC` is defined in the ADLCSD table and that it is present in one of the CICS load libraries. See the section *z/OS Installation* or *z/VSE Installation* in the *ADL Installation* documentation for details.
ADL0905	Invalid ADL nucleus for CICS	During initialization of ADL, the nucleus `DAZNUCC` was loaded but found to be	Relink `DAZNUCC`. See the section *z/OS Installation* or *z/VSE Installation* in the *ADL Installation* documentation for details.

Number	Message	Cause	Action
		invalid. It probably was not linked properly.	
ADL0906	ADL module `xxxxxxxx` is not compatible to CICS level	During initialization of the ADL, the ADL module `xxxxxxxx` was found to be incompatible to the CICS level.	Use only modules which reside in the ADL CICS load library corresponding to the CICS level in use. If the module is partof `DAZNUCC`, then relink the nucleus. See the section *z/OS Installation* or *z/VSE Installation* in the *ADL Installation* documentation for details.
ADL0907	Insufficient storage for `GETMAIN`	CICS `GETMAIN` failed during initialization of the `DAZCICS` work area.	Retry. If the error persists, then insufficient storage is available in the CICS partition/region. Check the possible reasons for this and take appropriate action.
ADL0908	Unable to get work area for `DAZCICS`	Internal error.	Please contact your Software AG technical support representative.
ADL0909	Unable to load ADL table `DAZPSB`	The ADL table `DAZPSB` could not be loaded.	Check that `DAZPSB` is defined in the ADLCSD table and that it is present in one of the CICS load libraries. See the section *Generating the Runtime Control Tables* in the *ADL Interfaces* documentation for details.
ADL0910	Unable to load ADL table `DAZDBD`	The ADL table `DAZDBD` could not be loaded.	Check that `DAZDBD` is defined in the ADLCSD table and that it is present in one of the CICS load libraries. See the section *Generating the Runtime Control Tables* in the *ADL Interfaces* documentation for details.
ADL0911	Unable to load ADL table `DAZBUF`	The ADL table `DAZBUF` could not be loaded.	Check that `DAZBUF` is defined in the ADLCSD table and that it is present in one of the CICS load libraries. See the section *Generating the Runtime Control Tables* in the *ADL Interfaces* documentation for details.
ADL0912	Unable to activate ADL task related user exit	The ADL task related user exit `DAZSYNC` could not be loaded or not be activated.	Check that `DAZSYNC` is defined in the ADLCSD table and that it is present in one of the CICS load libraries. See the section *Prerequisites for CICS Installation* in the *ADL Interfaces* documentation for details.
ADL0913	ADL is in doubt. Switch it off first.	An attempt was made to switch ADL on, while the status was 'in doubt'.	Switch ADL off. Before you switch it on again, determine why ADL was 'in doubt', and correct the error.
ADL0914	Error during `OPEN/CLOSE` destid `DAZD`	An error occurred during the open or close of the transient data extra partition destination `DAZD`.	Check the status of `DAZD` using the `CEMT` transaction. Check that `DAZD` is properly defined in the ADLCSD table. See the section *Prerequisites for CICS Installation* in the *ADL Interfaces* documentation for details.

Number	Message	Cause	Action
ADL0915	Error during `OPEN/CLOSE` destid `DAZR`	An error occurred during the open or close of the transient data extra partition destination `DAZR`.	Check the status of `DAZR` using the `CEMT` transaction. Check that `DAZR` is properly defined in the ADLCSD table. See the section *Prerequisites for CICS Installation* in the *ADL Interfaces* documentation for details.
ADL0916	Error during `OPEN/CLOSE` destid `DAZP`	An error occurred during the open or close of the transient data extra partition destination `DAZP`.	Check the status of `DAZP` using the `CEMT` transaction. Check that `DAZP` is properly defined in the ADLCSD table. See the section *Prerequisites for CICS Installation* in the *ADL Interfaces* documentation for details.
ADL0917	Unable to load Adabas	The Adabas load module could not be loaded.	Ensure that the Adabas module, as named with the `ADANAME` parameter, is properly defined in the ADLCSD table and is present in one of the CICS load libraries. See the section *ADL Parameter Module* in the *ADL Installation* documentation for more details on the `ADANAME` parameter.
ADL0919	Invalid `ADALNC` version encountered	The Adabas link module is a version which is not supported by the ADL Consistency Interface.	Check if the Adabas link module is a supported version, i.e. CICS command level version. Contact your Software AG technical support representative for further help.

Number	Message	Cause	Action
ADL0920	Incompatible versions of ADL modules	An internal error. The internal version numbers of ADL modules do not match.	The whole ADL system supplied is inconsistent. Please contact your Software AG technical support representative.
ADL0921	Switch on `ADL CALLDLI` Interface first	The function requested can only be performed when the `ADL CALLDLI` Interface has been activated.	Self-explanatory.
ADL0922	No `PSB`'s in table `DAZPSB`	No `PSB`s found in `DAZPSB`.	Create a `DAZPSB` module with at least one `PSB`. See the section *Generating the Runtime Control Tables* in the *ADL Interfaces* documentation for details.
ADL0923	End of `PSB` table reached	Self-explanatory.	None. This message is for information only.
ADL0924	Trace facility already switched on	Self-explanatory.	None. This message is for information only.
ADL0925	Invalid `TWA` supplied for `DAZCICS`	The transaction which calls `DAZCICS` does not supply a valid `TWA`.	Specify a `TWA` size of at least 24 bytes for each transaction calling `DAZCICS`.

Number	Message	Cause	Action
ADL0926	Initialization of Adabas D.C.I. failed	The initialization call of the Adabas direct call interface failed.	Check if the supplied Adabas supports the direct call interface. Please contact your Software AG technical support representative for further help.
ADL0927	Initialization of ADL user exit failed	The initialization call of the ADL user exit failed.	Please contact your Software AG technical support representative.
ADL0928	Adabas D.C.I. Call failed Adabas-Rsp: `xxx`	The initialization call of the Adabas direct call interface returned an unexpected response code.	Check if the supplied Adabas supports the direct call interface. Please contact your Software AG technical support representative for further help.
ADL0931	Adabas Bridge for DL/I - Nucleus loaded	The ADL nucleus `DAZNUCC` has been loaded.	None. This message is for information only.
ADL0932	ADL - Run without Consistency	The ADL user exit was not applied.	The ADL user exit must be loaded to run with the ADL Consistency Interface. Follow the steps described in the section *Activating the Consistency Interface* in the *ADL Interfaces* documentation.
ADL0933	ADL Consistency Interface is active	The ADL Consistency Interface has been activated.	None. This message is for information only.
ADL0934	ADL table `DAZACT` loaded	The ADL table module `DAZACT` has been loaded.	None. This message is for information only.
ADL0935	ADL table `DAZBUF` loaded	The ADL table module `DAZBUF` has been loaded.	None. This message is for information only.
ADL0936	ADL table `DAZPSB` loaded	The ADL table module `DAZPSB` has been loaded.	None. This message is for information only.
ADL0937	ADL table `DAZDBD` loaded	The ADL table module `DAZDBD` has been loaded.	None. This message is for information only.
ADL0938	ADL task related user exit is active	The ADL task related user exit has been activated.	None. This message is for information only.
ADL0939	Adabas Bridge for DL/I initialised	ADL has been initialized.	None. This message is for information only.

Number	Message	Cause	Action
ADL0940	Adabas Bridge for DL/I switched off	ADL has been deactivated.	None. This message is for information only.
ADL0941	ADL Trace facility switched on	The ADL trace facility has been activated.	None. This message is for information only.
ADL0942	ADL Trace facility switched off	The ADL trace facility has be deactivated.	None. This message is for information only.
ADL0943	ADL table modules dumped	The ADL table modules `DAZSYSDS`, `DAZPSB`, `DAZBUF`,	None. This message is for information only.

Number	Message	Cause	Action
		DAZDBD and the directory of the ADL zap module have been written to the dump file.	
ADL0944	ADL table modules and DAZNUCC deleted	The ADL table modules DAZSYSDS, DAZPSB, DAZBUF, DAZDBD and the ADL nucleus DAZNUCC have been released.	None. This message is for information only.
ADL0945	User exit routine XXXXXXXX could not be loaded	A user-supplied index maintenance exit routine could not be loaded during ADL start-up under CICS.	Check that the exit routine name is correctly specified in the DBD source, that the corresponding module is in the load library of CICS, and that the module is defined in the ADLCSD table.
ADL0946	PSB XXXXXXXX successfully initialized	The PSB indicated has been successfully initialized.	None. This message is for information only.
ADL0947	PSB XXXXXXXX - at least one DBD is not converted	During initialization of the PSB, at least one of the DBDs referenced was found to be not converted.	None. This message is for information only. See the section *ADL Conversion Utilities for DBDs and PSBs* in the *ADL Conversion* documentation for details.
ADL0948	PSB XXXXXXXX not found in ADL directory file	The PSB indicated could not be found in the ADL directory file.	Run the CBC utility for the PSB indicated. See the section *ADL Conversion Utilities for DBDs and PSBs* in the *ADL Conversion* documentation for details.
ADL0949	Error during write to destid DAZP	An error occurred while writing to the transient data extra partition destination DAZP.	Check the status of DAZP using the CEMT transaction. Check that DAZP is properly defined in the ADLCSD table.
ADL0950	Task xxxxxx abend DAZA. ADL Error no.: yyyyyy	A task with the task number xxxxxx was abended with abend code 'DAZA' by ADL. The value yyyyyy is the number of the error which originated the abend.	Follow the instructions given for the error number specified in yyyyyy in the error message. Note : This message appears only on the operator console. For specific errors, this message is preceded by the originating error message.

Precompiler Messages

Number	Message	Cause	Action
ADL1001	Syntax error: invalid trigger	The command trigger was not `EXEC DLI` or `EXECUTE DLI`.	Check the application program and correct the error.
ADL1002	Syntax error: invalid command	An invalid function was specified for the `EXEC` command.	Check the application program and correct the error.
ADL1003	Syntax error: invalid `PSB` option	The `PSB` option was not specified correctly for a scheduling command. Specify either "`PSB(psbname)`" or "`PSB(data-area)`".	Check the application program and correct the error.
ADL1004	Syntax error: keyword specified more than once	At least one keyword was specified more than once in a single command.	Check the application program and correct the error.
ADL1005	Syntax error: keywords mutually exclusive	At least one of the keywords specified in the command cannot be used with one or more of the other keywords specified.	Check the application program and correct the error.
ADL1006	Syntax error: keyword not compatible with command	At least one of the keywords specified is not allowed for the command in question.	Check the application program and correct the error.
ADL1007	Syntax error: invalid relational operator	An invalid relational operator has been specified in a `WHERE` clause. Valid operators are "`>`", "`<`", "`=`", "`:=`", "`<=`", "`>=`", "`=<`", "`=>`", "`  >`", "`  <`".	Check the application program and correct the error.
ADL1008	Syntax error: invalid Boolean operator	An invalid Boolean operator was specified in a `WHERE` clause. Valid operators are "`OR`" and "`AND`".	Check the application program and correct the error.
ADL1009	Syntax error: invalid keyword	An invalid keyword was specified in the command.	Check the application program and correct the error.
ADL1010	Syntax error: buffer overflow	The internal buffer used for translating the command is too small.	Increase the buffer size using the `PBUF` parameter. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL1011	Syntax error: unmatched parentheses	A single parenthesis was found.	Check the application program and correct the error.
ADL1012	Syntax error: `WHERE` not allowed with `KEYS`	The `WHERE` and `KEYS` clauses are mutually exclusive.	Check the application program and correct the error.
ADL1013	Syntax error: `KEYS` must be on first level	The `KEYS` clause may only be specified in the first `SEGMENT` option specified.	Check the application program and correct the error.

Number	Message	Cause	Action
ADL1014	Syntax error: CURRENT on last level	The CURRENT option may not be specified in the last SEGMENT option.	Check the application program and correct the error.
ADL1015	Syntax error: SETPARENT on last level	The SETPARENT option may not be specified on the last SEGMENT level.	Check the application program and correct the error.
ADL1016	Syntax error: Qualification statement on last level	The WHERE clause cannot be specified on the last SEGMENT level of an insert command.	Check the application program and correct the error.
ADL1017	Syntax error: invalid ID	An invalid ID has been specified in a checkpoint command.	Check the application program and correct the error.
ADL1018	Syntax error: invalid PCB specification	An invalid PCB option was specified for the command in question.	Check the application program and correct the error.
ADL1019	Syntax error: invalid alphanumeric string	An invalid alphanumeric string was encountered in a command option.	Check the application program and correct the error.

Number	Message	Cause	Action
ADL1020	Syntax error: invalid string	An invalid string has been encountered in a command option.	Check the application program and correct the error.
ADL1021	Syntax error: too many field lengths	The number of field lengths specified in the FIELDLENGTH option does not match the number of field names referenced in the corresponding WHERE clause.	Check the application program and correct the error.
ADL1022	Syntax error: missing FIELDLENGTH option	The FIELDLENGTH option is missing for this command.	Check the application program and correct the error.
ADL1023	Syntax error: missing SEGLENGTH	The SEGLENGTH option is missing for this command.	Check the application program and correct the error.
ADL1024	Syntax error: missing LENGTH	The LENGTH option is missing for this command.	Check the application program and correct the error.
ADL1025	Syntax error: missing ID for XRST/SYMCHKP	The ID option is missing for this command.	Check the application program and correct the error.
ADL1026	Syntax error: too many areas for XRST/SYMCHKP	More than 7 areas were specified in this command.	Check the application program and correct the error.
ADL1031	An internal error has occurred	Internal error.	Please contact your Software AG technical support representative.
ADL1032	Error during punch card	Internal error.	Please contact your Software AG technical support representative.

Error Messages from the Consistency Interface

If an error situation occurs, the ADL Consistency Interface returns the response code 216 in the user supplied Adabas control block. The application may then retrieve a more detailed error message. See the section *Error Handling of the Consistency Interface* above for more details.

Note that all errors listed below cause the ADL Consistency Interface to terminate the data base request without any action. There is only one exception, namely the error code 1200.

Number	Message	Cause	Action
ADL1200	Consistency - Return code zero	The ADL Consistency Interface has no return code set. This return code is set when an attempt is made to read an ADL Consistency Interface error message and no return code is yet set.	None. This is an informational message.
ADL1201	The original Adabas link module could not be loaded	(Batch only) During initialization of the ADL *Consistency Interface*, the original Adabas link module could not be loaded.	Ensure that the original Adabas link module, as named in the ADL link module substitute, is available in one of the load libraries. See the *ADL Interfaces* documentation for more details.
ADL1202	Non-supported Adabas function	An attempt was made to update an ADL file with an unsupported Adabas function.	Correct the program. See the *ADL Interfaces* documentation for more details.
ADL1206	Incompatible versions of ADL modules	Internal error.	Please contact your Software AG technical support representative.
ADL1207	Consistency - `GETMAIN` failed	(CICS environment only.) A CICS `GETMAIN` request from the ADL Consistency Interface failed.	If the failure was caused by insufficient storage available from CICS, increase the size of the CICS region/partition. Otherwise call Software AG technical support.
ADL1208	Overflow in internal format buffer stack	An overflow was detected in the internal format buffer stack used during analysis of an Adabas call.	Increase the value of the `FSTAC` parameter for the ADL parameter module. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details.
ADL1209	Missing delimiter in format buffer	In the user supplied format buffer of an Adabas call, a required delimiter is missing.	Correct the program.

Number	Message	Cause	Action
ADL1210	User format buffer references an undefined field	A field referenced in the user format buffer could not be located in the Adabas file description table.	Correct the program.
ADL1211	Invalid entry element in user format buffer	An invalid entry in the user format buffer was encountered.	Correct the program.
ADL1212	Invalid sequence of entries in format buffer	An invalid sequence of entries in the user format buffer was encountered.	Correct the program.
ADL1213	Record buffer too short	The record buffer length specified in the user supplied Adabas control block was too short.	Correct the program.
ADL1214	Error during convert zoned number	Internal error.	Please contact your Software AG technical support representative.
ADL1215	Non-standard field format specified	The ADL Consistency Interface requires a standard format specification in the user format/record buffer for most of the fields in ADL files, but at least one non-standard format has been encountered.	Correct the program.
ADL1216	Error in input card for the Adabas link module substitute in batch	An invalid input card was read from file `DAZIN2` (z/OS) or logical unit `SYSIPT` (z/VSE).	Correct the input card.
ADL1217	Format buffer references ADL internal field	An ADL internal field (one of `Z0` - `Z8`) was referenced in the user format buffer.	Correct the program.
ADL1218	Format buffer references secondary index field	A field generated for a DL/I secondary index was referenced in the user format buffer.	Correct the program.
ADL1219	Format buffer references to more than one DL/I segment	In the user format buffer, fields have been referenced which correspond to more than one DL/I segment.	Correct the program.

Number	Message	Cause	Action
ADL1220	Relation data for bi-directional logical child segment incomplete	In the user format buffer, fields have been referenced which correspond to a DL/I logical child segment. The relation data for this segment are incompletely specified.	Specify the partial concatenated key fields of all physical and logical parent segments.
ADL1221	Overflow in internal segment qualification statement buffer	Internal error.	Please contact your Software AG technical support representative.

Number	Message	Cause	Action
ADL1223	Failure during internal Adabas `L1` call	Internal error.	Please contact your Software AG technical support representative.
ADL1224	Attempt to delete a segment which has physical dependents	An attempt was made to delete an ADL record type corresponding to a DL/I parent segment which has still dependent segment occurrences.	Delete all dependent segments first.
ADL1227	Unexpected response code on internal Adabas 'L1' call	Internal error.	Please contact your Software AG technical support representative.
ADL1228	Attempt to alter sequence field	In the user record buffer, a field corresponding to the sequence field of a DL/I segment was altered.	Correct the program.
ADL1229	Non standard length specified for sequence field	In the user format buffer, a non-standard length was specified for a field corresponding to the sequence field of a DL/I segment.	Correct the program.
ADL1230	Partial concatenated key fields but no segment referenced	In the user format buffer, partial concatenated key fields have been referenced, but no field corresponding to DL/I segment data.	Correct the program.
ADL1231	Attempt to alter partial concatenated key fields	In the user record buffer, an attempt was made to alter the contents of one or more partial concatenated key fields.	Correct the program.
ADL1232	Non-standard length specified for a partial concatenated key field	In the user format buffer, a non-standard length for a partial concatenated key field was supplied.	Correct the program.
ADL1240	Overflow in Segment Description Table	An overflow has occurred in the segment description table.	Batch only: Increase the size of the segment description table using the SDT parameter. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details. CICS only: Check if the values for generating the `DAZDBD` table are still sufficient. See the section *Generating the Runtime Control Tables* in the *ADL Interfaces* documentation for details.
ADL1241	Overflow in the File Description Table	An overflow has occurred in the ADL file description table. This	Batch only: Increase the size of the ADL file description table using the

Number	Message	Cause	Action
		can happen if new Adabas files or fields have been added.	`FDT` parameter. See the section *ADL Parameter Module* in the *ADL Installation* documentation for details. CICS only: Check if the values for generating the `DAZDBD` table are still sufficient. See the section *Generating the Runtime Control Tables* in the *ADL Interfaces* documentation for details.
ADL1242	Error detected in `DBD ECB`	Internal error.	Please contact your Software AG technical support representative.
ADL1243	Internal Error	Internal error.	Please contact your Software AG technical support representative.
ADL1244	`DBD` in Consistency `PSB` is connected with non-converted `DBD`	A `DBD` which belongs to the internal Consistency `PSB` is logically connected with a non-converted `DBD`.	Use the ADL Online Services to select just those `DBDs` for the Consistency `PSB` which are only connected with converted `DBDs`.
ADL1251	Attempt to access data base whilst ADL was shutting down	CICS only: Self-explanatory.	Rerun your program when ADL is active.
ADL1252	PSB `'ADL$PSB'` not found in the table of PSBs, `DAZPSB`	CICS only: An invalid table of PSBs, `DAZPSB`, has been encountered.	Generate a correct table of PSBs. See the *ADL Interfaces* documentation for details.
ADL1253	No free entry in task id table found	CICS only: The ADL internal task ID table has been exhausted.	Increase the `TSKENT` parameter of the `DBDMAC` macro during creation of the `DAZDBD` table.See the *ADL Interfaces* documentation for details.
ADL1254	Failure during `GETMAIN` request	CICS only: Probably internal error.	Please contact your Software AG technical support representative.

Number	Message	Cause	Action
ADL1272	DL/I status code `'AC'`	ADL response code corresponding to the DL/I status code mentioned above.	Informational message. ADL terminates the data base request.
ADL1273	DL/I status code `'AD'`	ADL response code corresponding to the DL/I status code mentioned above.	Informational message. ADL terminates the data base request.
ADL1274	DL/I status code `'AH'`	ADL response code corresponding to the DL/I status code mentioned above.	Informational message. ADL terminates the data base request.
ADL1275	DL/I status code `'AJ'`	ADL response code corresponding to the DL/I status code mentioned above.	Informational message. ADL terminates the data base request.
ADL1276	DL/I status code `'AK'`	ADL response code corresponding to the DL/I status code mentioned above.	Informational message. ADL terminates the data base request.

Number	Message	Cause	Action
ADL1277	DL/I status code 'AM'	ADL response code corresponding to the DL/I status code mentioned above.	Informational message. ADL terminates the data base request.
ADL1278	DL/I status code 'AO'	ADL response code corresponding to the DL/I status code mentioned above.	Informational message. ADL terminates the data base request.
ADL1279	DL/I status code 'AU'	ADL response code corresponding to the DL/I status code mentioned above.	Informational message. ADL terminates the data base request.
ADL1280	DL/I status code 'BM'	ADL response code corresponding to the DL/I status code mentioned above.	Informational message. ADL terminates the data base request.
ADL1281	DL/I status code 'DA'	ADL response code corresponding to the DL/I status code mentioned above.	Informational message. ADL terminates the data base request.
ADL1282	DL/I status code 'DJ'	ADL response code corresponding to the DL/I status code mentioned above.	Informational message. ADL terminates the data base request.
ADL1283	DL/I status code 'DP'	ADL response code corresponding to the DL/I status code mentioned above.	Informational message. ADL terminates the data base request.
ADL1284	DL/I status code 'DX'	ADL response code corresponding to the DL/I status code mentioned above.	Informational message. ADL terminates the data base request.
ADL1285	DL/I status code 'GA'	ADL response code corresponding to the DL/I status code mentioned above.	Informational message. ADL terminates the data base request.
ADL1286	DL/I status code 'GB'	ADL response code corresponding to the DL/I status code mentioned above.	Informational message. ADL terminates the data base request.
ADL1287	DL/I status code 'GE'	ADL response code corresponding to the DL/I status code mentioned above. This is probably caused by an insert request with an incorrectly specified PCK or VCK field.	Informational message. ADL terminates the data base request.
ADL1288	DL/I status code 'GK'	ADL response code corresponding to the DL/I status code mentioned above.	Informational message. ADL terminates the data base request.
ADL1289	DL/I status code 'GP'	ADL response code corresponding to the DL/I status code mentioned above.	Informational message. ADL terminates the data base request.
ADL1290	DL/I status code 'II'	ADL response code corresponding to the DL/I status code mentioned above. This is probably caused by an insert request to an already existing dependent segment with a unique sequence field.	Informational message. ADL terminates the data base request.
ADL1291	DL/I status code 'IX'	ADL response code corresponding to the DL/I status code mentioned above. This is probably caused by an insert request with an incorrectly specified VCK field.	Informational message. ADL terminates the data base request.
ADL1292	DL/I status code 'KA'	ADL response code corresponding to the DL/I status code mentioned above.	Informational message. ADL terminates the data base request.
ADL1293	DL/I status code 'KB'	ADL response code corresponding to the DL/I status code mentioned above.	Informational message. ADL terminates the data base request.

Number	Message	Cause	Action
ADL1294	DL/I status code `'KC'`	ADL response code corresponding to the DL/I status code mentioned above.	Informational message. ADL terminates the data base request.
ADL1295	DL/I status code `'KD'`	ADL response code corresponding to the DL/I status code mentioned above.	Informational message. ADL terminates the data base request.
ADL1296	DL/I status code `'LC'`	ADL response code corresponding to the DL/I status code mentioned above.	Informational message. ADL terminates the data base request.
ADL1297	DL/I status code `'RX'`	ADL response code corresponding to the DL/I status code mentioned above.	Informational message. ADL terminates the data base request.
ADL1298	DL/I status code `'XH'`	ADL response code corresponding to the DL/I status code mentioned above.	Informational message. ADL terminates the data base request.

CICS Transaction ABEND Codes

Under CICS the following abnormal termination codes can occur:

Code	Message	Cause	Action
DAZ1	No free `LUB` found	No free local user block (`LUB`) was found for the current transaction in the `LUB` table in `DAZDBD`.	Retry. If the error persists, then increase the size of the LUB table by specifying the `LUBENT` parameter with the `DBDMAC` statement at the `DAZDBD` generation. See the section *Generating the Runtime Control Tables* in the *ADL Interfaces* documentation for details.
DAZ2	EXEC CICS call failed	ADL received an unexpected response from an EXEC CICS call.	Please contact your Software AG technical support representative.
DAZ3	Requested `PSB` not available	The `PSB` specified was not found in the `PSB` table `DAZPSB` during processing of a DL/I scheduling call.	Check `DAZPSB` and correct the error.
DAZ4	GETMAIN failed	A CICS `GETMAIN` for working storage for the ADL failed.	Retry. If the error persists, then insufficient storage space is available in the CICS partition/region. Check possible reasons for this and take appropriate action.
DAZ6	No final commit/backout before task end	The task ended, but no final commit or backout has been issued before.	Please contact your Software AG technical support representative.
DAZ7	Insufficient `TWASIZE` for task	An attempt was made to issue a scheduling call for a task defined in the `PCT` with a `TWASIZE` of less than 24 bytes.	Increase the `TWASIZE` specification in the `PCT`.

Code	Message	Cause	Action
DAZ8	ADL EXEC call to DL/I	A program using the HLPI and compiled using the ADL precompiler attempted to issue a call against an unconverted DBD.	Examine the PSB and convert the DBD in question.
DAZ9	Invalid PCB number	A call generated by the ADL precompiler specified a PCB number larger than the number of PCBs in the PSB for which a scheduling call was issued.	Check application and correct the error.
DAZA	Unrecoverable Error - Transaction Abended	ADL detected an unrecoverable internal error or received an unrecoverable response code from Adabas.	The ADL error code is contained in an "ADL....." error message sent to the user's terminal as well as in the message "ADL0950" sent to the operator console. Please refer to the explanation of the actual ADL error code.
DAZB	Shared database support was not available	A database request was received from a batch region. The request can not serviced, because no interface scheduling block was found for the mirror task. Shared database support requires that the original DL/I is still active in the CICS system.	Re-activate DL/I in the CICS system and make sure that the required PSB is defined for both DL/I and ADL. If DL/I is not available in your environment, you should investigate if you could run the batch job under direct control of ADL in "SDB" mode.
DAZD	Error during ADL function under CICS	An error occurred while invoking an ADL function (like 'Switch ADL on/off', 'Switch trace on/off') from an Assembler program under CICS.	Check if DAZCICS is defined correctly in the ADLCSD table. If the DAZCFCT program has been used, check if the &FUNC parameter is specified correctly.

5   Appendix A - Glossary of ADL Terms

This chapter covers the following topics:

ACB

1. Adabas Control Block (see *Adabas Command Reference* documentation).

2. Application Control Block (DL/I term). DL/I internal control blocks which combine the `PSB` and `DBD` information. DL/I ACBs are not used by ADL.

ADL

Adabas Bridge for DL/I.

ADL control file

Synonym for **ADL Directory file**.

ADL Directory file

An Adabas file used to store all `DBD`, `PSB` and system related information relevant to ADL. Also referred to as the **ADL control file**.

ADL file

Generic term for any Adabas file which results from the conversion of a DL/I data base.

ADL Online Services

A program for maintaining the ADL Interfaces under CICS and for retrieving information from the ADL Directory file.

ADL parameter module

A module containing all system relevant parameters for ADL (see ADL Parameter Module in the *ADL Installation* documentation).

ADL precompiler

The ADL supplied precompiler to translate `EXEC DL/I` commands in COBOL or PL/I programs into calls to ADL. Thus the **HLPI** becomes obsolete.

Call level

A program which communicates with the database system through `CBLTDLI`, `PLITDLI` or `ASMTDLI` call statements.

CALLDLI Interface

The ADL interface to intercept DL/I data base calls from command level or call level application programs.

CBC

Control block conversion.

CBC utility

The ADL supplied utility programs for the **control block conversion**.

Child segment

One of the segment types or occurrences directly depending on another segment type or occurrence. The child segment types are all those segment types referring to another segment type in the `PARENT=` parameter of the `SEGM` macro.

Command level

A program which communicates with the database system through "`EXEC DLI`" statements.

Concatenated key

Identifies, by the means of key fields, the position of a segment occurrence in the data base. The concatenated key of a segment occurrence is built up from all individual key fields of its parent segment occurrences (top - down) including its own key field.

Control block conversion

The process of transforming a **DBD** or **PSB** source into an entry in the **ADL Directory file**.

Consistency Interface

The ADL interface to intercept Adabas database calls.

DAZIFP

The ADL batch region controller (to replace the DL/I batch region controller).

DAZPARM

The **ADL parameter module**.

DAZZLER

A program to test DL/I calls in batch.

DBD

Data Base Definition. Defines the layout of a physical, logical or secondary index DL/I data base. The user defines a DL/I data base by calls to various assembler macros.

DBD source

The source code for the definition of a DL/I data base.

DBD control block

An entry in the **ADL Directory file** for one particular DL/I database.

DIB

DL/I interface block. An area used for communication between the data base system and a command level application.

DL/I

A generic term for DL/I DOS/VS and IMS/VS used throughout the entire ADL documentation.

ECB

External Control Block. A **DBD** or **PSB** control block as used by ADL in its external (address free) format. `ECB`s are stored in the **ADL Directory file**.

Exec level

Command level.

FDT

File Description Table.

1. Describes the layout of an Adabas file.

2. A table used internally by ADL to describe the layout of an ADL file.

Field

1. A sub-element of a DL/I segment.

2. An element of an Adabas file structure.

Group

Adabas term used to designate an Adabas field which is built up from lower level field definitions.

HLPI

High Level Programming Interface. Provides the link between command level programs and the DL/I data base system. This module becomes obsolete for command level programs which have been precompiled with the **ADL precompiler**.

ICB

Internal Control Block. A **DBD** or **PSB** control block as used by ADL in its internal format. During initialization of ADL, the relevant **ECB**s are converted into ICBs.

IMS

DL/I.

Internal pointer fields

An internal Adabas field, which is used by ADL to reflect the hierarchical structure of the data. Internal pointer fields are based on the **logical ID**s, **ISN**s, and **sequence field** values of the records in an ADL file.

I/O area

DL/I delivers segment data to this area (GET-calls) or receives segment data from the application (ISRT/REPL-calls) in this area. A common I/O area can be defined for all segments to be processed in one application program. In this case, the size of the I/O area must be identical to the size of the longest segment to be processed.

ISN

Internal Sequence Number. Each logical record within an Adabas file is assigned an ISN which serves as a logical identifier. ISNs are unique within a file. For ADL files, this ISN is assigned by ADL and cannot be changed by the user.

Language Interface

This interface provides language dependent entry points for `CALLDLI` statements for the most commonly used languages (`COBOL`, `PL/1`, `BAL`). The language interface routes the data base requests to the DL/I request handler.

Last-call save area

An area used internally by ADL to store the last Adabas call for each sensitive segment. This area can help to improve performance.

LCS

Last-call save area.

Logical ID

Every DBD in a logical relationship must be assigned to a unique logical ID. It is stored in the **ADL Directory file** and used to build up the **internal pointer fields**.

Logical pointers

In ADL files, logical pointers are Adabas fields which contain key values that provide the hierarchical relationship between records. Logical pointers are used by Natural applications.

Loop

Loop.

MFT

Multifetch Table.

Mixed mode

A DL/I application running under control of ADL, accessing ADL files concurrently with non-converted DL/I databases.

Multifetch

Adabas feature to reduce the communication overhead between the application program and the Adabas nucleus. Can be activated automatically by the ADARUN PREFETCH parameter.

Multifetch Table

A table which defines the number of records returned by the Adabas **Multifetch** feature for specific `PCB`/`SENSEG` combinations.

Normal mode

A DL/I application accessing ADL files only.

Parent segment

The direct parent segment type or occurrence of a segment type or occurrence. The parent segment type is defined in the `PARENT=` `parameter` of the SEGM macro in the **DBD**.

Partial Concatenated Key

The part of the **concatenated key** of a segment occurrence for one particular level in the hierarchy.

PCB

Program Communication Block.

1. A `PCB` definition in a **PSB** defines the view of a DL/I database for the application program.
2. The data area used for communication between the data base system and the application program. The user has to define a PCB mask for this purpose in the application program. The data areas corresponding to the PCB definitions in a **PSB** are provided by the data base system at the time when the `PSB` is initialized.

PCB mask

A user coded layout (Assembler `DSECT`, `COBOL` data definition etc.) of the PCB.

PCK

Partial Concatenated Key.

Physical pointer fields

Internal pointer fields.

Prefetch

Functional subset of **Multifetch**.

PSB

Program Specification Block. Defines the user views (**PCB**s) of a DL/I data base as available for an application program.

PSB control block

An entry in the **ADL Directory file** for one particular PSB.

RBE

Record buffer extension.

Record Buffer Extension

A list to increase the record buffer length for specific `PCB/SENSEG` combinations. This reduces the number of records returned by the Adabas Prefetch feature.

Response codes (Adabas)

An Adabas response to a request. The response codes are passed to the user in the **ACB**.

SDT

Segment Description Table. A table used internally by ADL to describe the layout of segments and their implementation in ADL files.

Segment

The element type of a DL/I data base structure. A segment is identified by its unique name in the database. Its position in the hierarchical database structure is identified by its level and its ordinal number. The term is sometimes also used for an occurrence of the segment type.

Sequence field

A field defined in a segment which determines the order of segment occurrences of a type. A sequence field may be defined as unique or non-unique. A root segment type must have a unique sequence field defined. A unique sequence field defined for a non-root segment type is not necessarily unique over the entire data base. It is however unique for all segment occurrences under the given parent segment occurrence.

SSA

Segment Search Argument. An area which specifies search criteria for one particular segment type in a DL/I data base call.

Status codes

A DL/I response to a request. The status codes are passed to the user in the **PCB**.

Twins

All segment occurrences of the same type under one specific **parent segment** occurrence.

UIB

User Interface Block. An area used for communication between the data base system and the application program under CICS.

VCK

Virtual Concatenated Key.

Virtual Concatenated Key (VCK)

Same as a **PCK**, but instead of corresponding to a physical child relationship it corresponds to a logical child relationship.