Britton Lee, Inc.

Britton Lee Host Software - BSD/ULTRIX/SUN UNIX

HOST SOFTWARE MESSAGE SUMMARY

(IDL Version)

Revision 3.5.1.1
(87/02/08)

(R3v5)

February 1988
Part Number 205-1432-004

This document supercedes all previous documents. This edition is intended for use with Britton Lee Host Software release number 3.5 and subsequent releases, until further notice.

The information contained within this document is subject to change without notice. Britton Lee assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under license and may only be used or copied by the terms of such license.

IDM, Intelligent Database Language, and IDL are trademarks of Britton Lee Inc.

TABLE OF CONTENTS

HOST SOFTWARE MESSAGE SUMMARY
(IDL Version)

Britton Lee, Incorporated

Version 3.5.1.1 (87/02/08)

R3v5

This document summarizes the messages that may be generated by the Britton Lee host software package.

This document is arranged by function. In most cases the order is also alphabetical to simplify searching.

1. Database Server Query Messages

This section describes messages that are generated by the database server.

1.1. User Error Messages

The following messages result from errors encountered during query execution. They may be caused by user error.

IDM.E1

Insufficient descriptors available.

Explanation: Insufficient number of descriptors available for the relations referenced by your query.

User Action: This is a temporary condition. Try running your query again, or wait until other users free database server resources.

IDM.E2

<objname> already exists.

Explanation: A database object named <objname> already exists. You are not allowed to overwrite this object, and you cannot have another object with the same name.

User Action: Choose another name for <objname> or destroy <objname> and rerun the command. All objects (i.e., relations, files, stored commands, etc.) must have unique names for each user.

IDM.E3

Bad order number <order number> for variable <var number>.

Explanation: The order number (specified by a range ... with dorder = ...) was not acceptable. Order numbers may be from 0 to 254 with 255 disallowed.

User Action: Choose a different order number.

IDM.E4

Quota exceeded for relation.

Explanation: You have attempted to update a relation. The update would increase sizae of the relation beyond the quota set for it by the "create" command.

User Action: Increase the quota for the relation being updated or delete some data from the relation and reexecute the command.

IDM.E5

<dbname> is in use.

Explanation: An attempt was made to destroy the database, relation, file view or stored command <dbname>. An object cannot be destroyed if someone is using it.

User Action: Wait until <dbname> is no longer in use, or ask its user to relinquish it.

IDM.E6

<object> not found.

Explanation: The user has attempted to access <object> on the database server. <object> was not found in this context.

User Action: Try again, but check for spelling errors, etc. Use the system relations to verify that <object> actually exists and is being used in the correct context. To do this for relations, stored commands, and other objects, run the query:

```
range of r is relation
retrieve (r.name, r.type)
where r.owner = userid
```

If no results are returned, the object does not exist. If results are returned, the "type" field will be one of the following:

U — User relation
S — System relation
T — Transaction log
F — File
V — View

C — stored Command
P — stored Program
To see a list of available databases, do the following:
open system
range of d is databases
retrieve (d.name)
If you or the DBA does not own the relation that you want to access, then you must suffix the relation name with a colon and your username, for example:
kinds:larry

IDM.E7
Too many attributes.
Explanation: An attempt was made to create a relation with too many attributes. The maximum allowable number of attributes is 250.
User Action: A relation that large would probably be inefficient even if it were allowed. Break up the relation into logical components, and create several smaller relations.

IDM.E9
Bad option <option>.
Explanation: The user has attempted to set (or unset) an unknown or reserved option, <option>.
User Action: Unset the option <option> with the command,
unset <option>
to avoid getting this error again.

IDM.E10
Non char arg to concatenate: <operand>.
Explanation: The user tried to use the "concat" operator on the operand <operand>. The "concat" operator may only be used on character arguments, and <operand> is not of type char.
User Action: Check for typos. Make sure all character strings are enclosed by quotes. If the argument was a variable, make sure that it has type "char".

IDM.E11
Non char arg to substring: <operand>.
Explanation: The user tried to use the "substring" operator on the operand <operand>. "Substring" may only be used on character arguments, and <operand> is not a character string.
User Action: Check for typos. Make sure all character strings are enclosed by quotation marks. If the argument was a variable, make sure that it is type "char".

IDM.E12
Bad type in conversion: <op>.
Explanation: An attempt was made to perform an illegal conversion of the operand <op>. The attempted conversion is not allowed for an operand of this type.
User Action: Check query for typos.

IDM.E13
Bad symbol in expression: <operand 1> <operation> <operand 2>.
Explanation: This usually signifies an internal program error. Information sent to the database server was badly formatted or invalid.
User Action: Contact your Britton Lee distribution support person.

IDM.E14

<arg1> is not an aggregate operator.

Explanation: This usually signifies an internal program error. Information sent to the database server was badly formatted or invalid.

User Action: Contact your Britton Lee distribution support person.

IDM.E15

Arithmetic aggregate, <func>, on non-numeric.

Explanation: The user tried to use the aggregate function <func> on non-numeric data. <func> only works on numeric data.

User Action: Check for typos and other errors. You are probably trying to use a data value that is a character string or some other type where arithmetic does not make sense.

IDM.E16

Arithmetic not supported on floats: <operand 1> <operation> <operand 2>.

Explanation: An attempt was made to perform the <operation> operation on an operand of type "f4" or "f8". Arithmetic operations are not supported on floats.

User Action: Arithmetic operations on floats can only be performed on the host. If you discover that you will need to be doing a large number of floating point operations on this value, consult your database administrator to have the value converted to type BCD or BCDFLOAT, so that the operation can be performed.

IDM.E17

Non-numeric in arithmetic op: <operand 1> <operation> <operand 2>.

Explanation: The user tried to perform an arithmetic operation on non-numeric data in the expression <operand 1> <operation> <operand 2>.

User Action: Check for typos and other errors. You're probably trying to use the wrong data.

IDM.E18

Not comparable: <operand 1> <comparison> <operand 2>.

Explanation: The user tried to perform the comparison <comparison> on <operand 1> and <operand 2>. <operand 1> and <operand 2> are of incompatible types and cannot be compared.

User Action: Check for typos and other errors. You're probably trying to use the wrong attributes.

IDM.E19

Result for attribute: <domname> has wrong type.

Explanation: An attempt was made to append or replace an invalid tuple to a relation. The tuple was invalid because the value specified for the attribute <domname> was of the wrong type.

User Action: Determine the actual type of the attribute, and correct the query.

IDM.E20

Duplicate attribute name: <domname>.

Explanation: Duplicate attribute names are not permitted. Two or more attributes named <domname> were used in the target list. This is illegal for commands such as create, create view, replace, and append.

User Action: Try again, with unique attribute names.

IDM.E21

No free pages in database.

Explanation: There was not enough room in the database to perform the requested operation.

User Action: Use the "extend database" command in IDL to enlarge the database, or

destroy some relations in the database.

IDM.E22

ONCE or ONCEU returned <numtups> values.

Explanation: The aggregates ONCE and ONCEU are constrained to return only one tuple value. Instead, <numtups> values were returned.

User Action: Modify the query so these aggregates select only one value.

IDM.E23

Missing parameter: <param>.

Explanation: The user has tried to execute a stored command without entering the parameter <param>. The parameter is required.

User Action: Re-run the command, making sure to enter the parameter. Check the "execute" command in the IDL Reference Manual.

IDM.E24

Duplicate parameter: <param>.

Explanation: The user has tried to execute a stored command and has entered the parameter <param> twice.

User Action: Re-run the command. Enter each parameter once and only once.

IDM.E25

Duplicate stored-command name: <cmd-name>.

Explanation: An attempt was made to define a stored command named <cmd-name>. An object with that name already exists in the database.

User Action: Select a different name, or destroy the object with the same name.

IDM.E26

Relation or file is unavailable.

Explanation: The relation or file is currently being created or destroyed.

User Action: Try to access the relation or file again. This is a temporary condition.

IDM.E27

Too many attributes in index key.

Explanation: An attempt was made to create an index using a key that would contain too many attributes. The maximum number of attributes allowed in a key is 15.

User Action: Use only the first fifteen keys when creating the index.

IDM.E28

No attributes specified in index.

Explanation: An attempt was made to create an index, but no attributes were specified.

User Action: It does not make sense to have an index with empty keys. Choose attribute(s) on which to index.

IDM.E29

Index exists.

Explanation: An attempt was made to create an index that already exists.

User Action: If you wish to create another index, use the "create index" command with the "recreate" option.

IDM.E30

Index does not exist.

Explanation: An attempt was made to destroy or recreate an index that does not exist.

User Action: You are trying to destroy or recreate an index that has never been created. Check your command for typos.

IDM.E31

Too many variables used.

Explanation: The number of range variables used in the query plus the number of temporary relations needed to process the query exceeds 16.

User Action: Rephrase the query, or break it up into several smaller queries.

IDM.E32

Query buffer full.

Explanation: The database server's internal buffer overflowed. The query was too large and/or complex to be processed.

User Action: Rephrase the query, or break it up into several smaller queries.

IDM.E33

Duplicate key found for unique index.

Explanation: A duplicate key was found when the index was specified to be unique. This error can occur when the index is created or when a tuple is appended to the relation.

User Action: Check your data for typos. You may want to replace the old tuple, or you may choose to not append this new tuple.

IDM.E34

More than fifteen elements in by-list of aggregate.

Explanation: The list of elements after the "by" in the aggregate clause was too long. The maximum number of elements in the "by" list is fifteen.

User Action: Try to simplify the query. You can use the "concat" operator to reduce the number of elements if you are using character strings.

IDM.E35

By-list is too wide in aggregate.

Explanation: The list of elements following the "by" in the aggregate clause had too many bytes. The whole list cannot contain more than 255 bytes.

User Action: Rephrase the query, or break it up into several smaller queries.

IDM.E36

<paramcount> is too many parameters.

Explanation: An attempt was made to execute a stored command using too many parameters. The command was originally defined with fewer parameters.

User Action: Determine the correct number of parameters for the stored command and then try again.

IDM.E37

Bad length for char, binary or bcd field: <len>.

Explanation: The length <len> was given for a char, binary, or bcd field. This is an illegal length for the operand.

User Action: Determine the limits for this type of operand and try again.

IDM.E38

Too many databases.

Explanation: An attempt was made to create a database. The create failed because the number of databases would have exceeded the database server's maximum of 50.

User Action: Remove any unused databases or purchase another database server.

IDM.E39

Pattern matching string (<string>) in target list.

Explanation: A pattern matching character (*, ?, or []) was used as a data value in a query. This is probably due to executing a stored command that includes one of these characters in an argument.

User Action: Remove the pattern matching character, or precede it by a backslash character to remove its special meaning.

IDM.E40

<filename> exists and is not a file.

Explanation: An attempt was made to perform a file operation on the database server. object <filename>. <filename> is not a file.

User Action: Either find the correct name for the file you want to access, or determine the operation you want to use on <filename>.

IDM.E41

<command> may not be used in a stored command.

Explanation: An attempt was made to define a stored command containing the <command> command. <command> is not allowed in a stored command.

User Action: Try to find a different way to accomplish your goal.

IDM.E42

Out of space on disk <diskname>.

Explanation: An operation was attempted which would have used more room than is available on the disk named <diskname>.

User Action: Clean up the disk, removing and/or archiving unneeded material and try again. Alternatively, you could use a different disk, if one is available, or buy another disk drive.

IDM.E43

Permission denied on <idmobject> <attname>.

Explanation: The user attempted to use <idmobject> (<attname>), but did not have permission for this operation on this object.

User Action: Ask the owner of <idmobject> for the required permission.

IDM.E44

Not owner of <object>.

Explanation: The user attempted an operation on <object>. Only the creator of <object> is allowed to perform that operation.

User Action: You must ask the owner of the object to perform this operation on <object>.

IDM.E45

Illegal command.

Explanation: The command you tried either could not be done in this context or could not be done by you. The command was illegal for one or more of the following reasons: (1) You must be in the "system" database to use this command. (2) You must be the DBA of the current database to use this command. (3) You cannot use this command in a transaction. (5) You cannot use this command outside of a transaction. ("end transaction" and "abort transaction" commands only)

User Action: Check the manual for special restrictions on this command. If you keep getting this message, try "end transaction", because you may be in a transaction by accident.

IDM.E46

No open database.

Explanation: You cannot access data or create database objects until you open a database.

User Action: Open the database that you want to use using the "open" command, e.g.:

```
open mydatabase go
```

IDM.E47

<object> is not a transaction log.

Explanation: The user tried to perform an operation on <object>. The operation attempted is allowed only for a transaction log, and <object> is not a transaction log.

User Action: You are using the wrong object (or the wrong operation). Try again with the correct components.

IDM.E49

Format mismatch in copyin - relation: <relname>, attribute: <attname>.

Explanation: The format specified in the data for the <attname> attribute of the relation <relname> did not match the format of the data in the database.

User Action: If you want the copy in command to create <relname>, first destroy <relname> and then rerun the copy in. Otherwise, correct the type of <attname> in the input and then try again.

IDM.E50

Loop in Outer Join.

Explanation: The query contains outer join operators which cause a loop in the ordering of the variables.

User Action: Look for self joins as in: x.a *= x.b or loops in the query as in: x.a *= y.a and y.b *= z.b and z.b *= x.b

IDM.E51

File not open with proper mode.

Explanation: Usually a programmer error. An attempt was made to perform an operation on a file that was not opened for that operation.

User Action: Open the file specifying the operation(s) that you want to perform.

IDM.E53

Database locked for dump or rollforward.

Explanation: An attempt was made to access a database while that database was "locked". Databases may not be accessed while a dump or rollforward is in progress.

User Action: Wait for a while and then try again.

IDM.E54

Bad 'with' option: <badopt>.

Explanation: The user entered a command with an illegal 'with' option. <badopt> is not a legal option for the command.

User Action: Check the IDL or SQL Reference Manual for legal options for the command. Check for typos.

IDM.E55

Deadlock detected (xact <transaction>, relid <id>): transaction aborted.

Explanation: The database server detected a "deadlock" between your transaction and transaction <transaction> while accessing the relation with internal id <id> — a situation in which two or more processes must wait forever for certain resources that they can never acquire. When such a situation is detected, one of the processes must be aborted. Your unfortunate process has been chosen as the victim.

User Action: Try again. If this happens again, wait awhile until the resources become available.

IDM.E56

Can't create or destroy index on system catalogue.

Explanation: An attempt was made to create or destroy an index on a system relation. Creation and destruction of indices on system relations is not permitted.

User Action: Check that you are referencing the correct relation.

IDM.E57

Can't destroy, extend or truncate system relation: <relname>.

Explanation: An attempt was made to modify the system relation <relname>. If you were to modify this relation, the server could not function properly; therefore,

this operation is not permitted.
User Action: Check that you are referencing the correct relation.

IDM.E58
Transaction log out of sequence.
Explanation: An attempt was made to rollforward a transaction log with an out-of-sequence log. Logs must be rolled forward in sequential order.
User Action: Find the correct log and use it. If you are not sure which log is the right one, try several. The database server will only use the correct log, so you won't hurt anything by trying.

IDM.E59
Transaction log for wrong database.
Explanation: An attempt was made to rollforward a transaction log with a log for a different database.
User Action: Find the correct log and use it. If you are not sure which log is the right one, try several. The database server will only use the correct log, so you won't hurt anything by trying.

IDM.E60
View not updatable.
Explanation: This view cannot be updated because the results of such an update could not be unambiguously resolved.
User Action: You will have to update the underlying objects.

IDM.E61
Tuple too wide.
Explanation: An append or replace command tried to add a tuple that was wider than the maximum tuple width of 2000 bytes.
User Action: You cannot store the tuple. It may be be possible to store the data using two or more relations that have a single key value. See your database administrator for assistance.

IDM.E62
Too many open files.
Explanation: An attempt was made to open a database server file. The number of open files would have exceeded the legal maximum of 20 files per database.
User Action: Close some files or use more than one dbin.

IDM.E63
Option is of wrong type: <optval>.
Explanation: An attempt was made to perform an operation with options. One of the options was assigned the value <optval>. <optval> is not of the correct data type for the option.
User Action: Determine the meaning and proper type for the option. You may be trying to use the wrong option. Check for typos.

IDM.E64
Option specifier is missing for: <optname>.
Explanation: An attempt was made to perform an operation with the <optname> option, but no value was given for the option.
User Action: Determine the meaning and the legal values for the <optname> option. You may be trying to use the wrong option.

IDM.E65
Stored command or program is too big.
Explanation: The user has tried to create a stored command or program whose size is larger than (approximately) 8K, or there are too many stored commands

already.

User Action: You must use more than one stored command. For example, break your command into two commands, "mycommand1" and "mycommand2".

IDM.E66

Index would be <count> bytes too big.

Explanation: The size of the index would exceed the maximum allowable size by <count> bytes.

User Action: You will have to develop a different access scheme that does not require such a large index.

IDM.E67

Parameter <paramname> (<paramval>) is of wrong type.

Explanation: An attempt was made to execute a stored command with the value <paramval> assigned to the parameter <paramname>. <paramval> is not a valid type for <paramname>.

User Action: Check query for typos. Determine correct parameter names and types.

IDM.E68

Pattern matching string: <pmstring> is malformed.

Explanation: An attempt was made to employ a bad pattern-matching string, <pmstring>.

User Action: Check the syntax of pattern-matching strings. Look for an occurrence of "[" without a matching "]", and similar errors.

IDM.E69

Attribute <attname> cannot be updated.

Explanation: This attribute cannot be updated because the results of such an update could not be unambiguously resolved. This problem can only occur when updating views.

User Action: You will have to update the underlying objects.

IDM.E70

<objname> is not a relation.

Explanation: <objname> must be a user relation for this operation to succeed. In fact, <objname> exists but is of the wrong type — a file, transaction, system relation, etc.

User Action: Determine which object you actually wish to use. If you are trying to copy a transaction log, use "idmdump". To check the type of the object, issue the query:

```
range of r is relation
retrieve (r.type) where r.name = <objname>
go
```

The type will be from among the following:

U — User relation (this should work)
S — System relation
T — Transaction log
F — File
V — View
C — stored Command
P — stored Program

IDM.E71

Name is too long (<badname>).

Explanation: The user tried to give an database server object a name "<badname>". A name may be no longer than twelve characters.

User Action: Try again, but use a shorter name.

IDM.E72

<object> has dependencies (<dependent-obj>).

Explanation: The user tried to destroy the object <object>. The object <dependent-obj> is dependent on <object> and must be destroyed first.

User Action: Destroy <dependent-obj>, then try again to destroy <object>. The "crossref" relation can be used to find all of the dependent objects. For example,

```
range of c is crossref
retrieve (rel_name(c.drelid))
where c.relid = rel_id(<object>)
go
```

will give you a list of all of the objects that depend on <object>.

IDM.E73

Bad protection mode.

Explanation: An attempt was made to set a protection mode for some database server object. Either the mode was unknown, or the mode was not appropriate for that object.

User Action: Determine the nature of the object and the appropriate protections for the object, and then try again.

IDM.E74

Numeric overflow: <bigval>.

Explanation: The value <bigval> is too large to fit into the specified target variable.

User Action: Either use a value within limits, or define a larger target. If you are certain that this is not an error, you can disable ALL overflow detection by using the command:

```
set "overflow" go
```

IDM.E75

Divide by zero: <where>.

Explanation: An attempt was made to divide by zero.

User Action: Check your data to insure that it is within range. If this seems acceptable, contact the person at your local site responsible for the program you are using. If you are certain that this is not an error you can ignore ALL divide by zero operations using the command:

```
set "divzero" go
```

IDM.E76

Insufficient space for numeric to character conversion: <bignum>.

Explanation: An attempt was made to convert the number <bignum> into a string. The string was not long enough to contain the number.

User Action: Specify a suitable string length.

IDM.E77

Temporarily cannot process open. Try again.

Explanation: The database server has run out of available database instantiations, and cannot open your database until another database has been closed.

User Action: Try again. If the same thing happens, ask someone else to free some resources.

IDM.E78

Bad value <badval> for option <option>.

Explanation: The user tried to set the option <option> to <badval>. <badval> is an illegal value for this option.

User Action: Determine what constitutes a legal value for the <option> option, and try again.

IDM.E79

Incorrect file for load database command.

Explanation: An attempt was made to load a database using a file with an incorrect format. A database can only be loaded from a file that was created by a "dump database" command.

User Action: You've used the wrong file. Find the correct file and try again.

IDM.E80

Insufficient space for load database <dbname>: need approx <pages-needed> pages.

Explanation: The space currently occupied by <dbname> is not sufficient to hold the earlier version being loaded. The new version must be at least as large as the version being loaded.

User Action: Use the IDL "extend database" command to enlarge the database, and then try again to "load".

IDM.E81

Command not allowed on system database.

Explanation: An attempt was made to dump or load the system database. This is not permitted.

User Action: Perform the operation on the correct database.

IDM.E82

Corrupt input file at <byte position>.

Explanation: An attempt was made to load a database or transaction log using a file with an incorrect format. A load can only be done from a file that was created by a corresponding "dump" command.

User Action: You've used the wrong file. Find the correct file and try again.

IDM.E83

Cannot dump to the same database.

Explanation: An attempt was made to dump a database or transaction log to a file in that same database. This is not permitted.

User Action: Choose a different database to dump to, or dump to host file or database server tape.

IDM.E84

Cannot load from the same database.

Explanation: An attempt was made to load a database or transaction log from a file in that same database. This is not permitted.

User Action: Choose a different database to load from, copy the file to that database, and then perform the load.

IDM.E85

Command needs one (and only one) relation of type transaction.

Explanation: A operation was attempted with an incorrect number of transaction logs. This can occur during an attempt to rollforward a non-transaction relation, or when attempting to audit from more than one transaction log.

User Action: Specify exactly one transaction log for this operation. This includes range variables.

IDM.E86

Incorrect use of "value" or "type" attribute.

Explanation: This error can only happen when accessing a transaction log, usually using the "audit" command in IDL. The "value" attribute can only be used in the target list, never in a qualification, join, or function. If the "value" of a REPLACE record is requested, the audit may not be restricted to exclude REP_BEGIN or REP_OLD. If the REP_OLD is included, REP_BEGIN must also be included. For an audit into command, the "value" field can be

retrieved only if "relid" is restricted to a single value.
User Action: Rephrase the query, observing the constraints listed above.

IDM.E87
Cannot return value field, relation no longer exists.
Explanation: This error can only occur in performing an "audit" command in IDL. The database server is unable to return a value, because the relation in question no longer exists.
User Action: Rephrase the "audit" command, without referring to the "value" attribute.

IDM.E88
Rounding occurred: <result>.
Explanation: The result of some arithmetic operation was too large to retain perfect accuracy. The database server had to round the number.
User Action: Determine whether <result> is close enough for your purposes. If not, you may have to increase the precision of the data using the "bcdflt" function. If the result data is already at maximum precision (31 digits), you will have to consider other algorithms.

IDM.E89
Numeric underflow: <result>.
Explanation: An arithmetic operation on floating-point numbers produced a result whose exponent was less than -1023. This number is too small for the database server to handle.
User Action: This is probably an arithmetic error. Check input to make sure that all values are within reasonable ranges. Sometimes rearranging a series of calculations will fix the problem.

IDM.E90
Tuple ignored during reconfigure command: type <type>, number <num>, value <val>.
Explanation: An invalid tuple was encountered in the "configure" relation. One or more of the attributes were invalid, and the tuple was ignored.
User Action: The reconfigure was completed except for this tuple. Correct the tuple and try again. The "configure" relation and its contents are described in System Administrator's Manual.

IDM.E91
Must restrict the query to one relid.
Explanation: An "audit into" command which saves the "value" field must be restricted to one "relid".
User Action: Simplify your command: use a separate "audit into" command for each relation you are interested in.

IDM.E93
Database <dbname> is too fragmented.
Explanation: <dbname> cannot be extended because it would then have more fragments than the allowable maximum.
User Action: This is a very rare error. It can come about when several databases have been repeatedly extended by small amounts. The condition can be fixed by dumping the database, destroying it, recreating it, and then loading it from the dump. When extending a database, it is better to do one large extend than a series of smaller extends.

IDM.E94
Cannot roll forward: relation <relname> is inconsistent.
Explanation: The rollforward failed because of an inconsistency in the relation <relname>. The database server could not recover to continue the rollforward.
User Action: None. This error is not recoverable. The data cannot be rolled forward. You

may want to reload the data from the last dump.

IDM.E95

Database <dbname> is not currently accessible.

Explanation: The database <dbname> was either being loaded or was locked by another user.

User Action: Wait until the database is available, and try again.

IDM.E96

Database <dbname> is on a disk which is not currently on line.

Explanation: The database server was unable to access the database <dbname> because one or more of the disks occupied by the database is currently turned off. The database server prefers to disallow all access rather than risk giving you access to incorrect data.

User Action: Contact your local support staff to have the disk turned on or fixed.

IDM.E97

Database <dbname> is open and cannot be locked.

Explanation: An attempt was made to lock the database <dbname>. <dbname> was open for dump or rollforward and could not be locked.

User Action: Wait until the database is no longer being used and then try again. Otherwise find out who has the database opened and request that it be closed.

IDM.E98

Disk <diskname> is not on line or cannot be used for this command.

Explanation: An attempt was made to perform an operation on the disk named <diskname>. That disk is not connected to the database server or is turned off.

User Action: Ask your local support staff to verify that the disk is on-line or use a different disk.

IDM.E99

Illegal source log or file for dump/load command: <objname>.

Explanation: An attempt was made to dump a transaction from an object that is not a transaction log, or to load a database or transaction log from an inappropriate source.

User Action: Find the correct object. You can only load from an object that was produced by a corresponding dump. You can only dump a database or a transaction log.

IDM.E100

Bad block <blockloc>: no free blocks to remap.

Explanation: The database server encountered a bad block when loading the database, and tried to remap it to another location. It was unable to find a location to map to.

User Action: Destroy the database, recreate it with a larger demand and load the database again.

IDM.E101

Bad header on block <blocknum>: impossible to remap.

Explanation: The database server encountered a bad block when loading the database. It is impossible to remap this block because the header is also bad.

User Action: The disk block cannot be recovered, so the best thing to do is to create a small dummy database on top of the bad block(s). This can be done by destroying the database that you are trying to load, and then creating the dummy database. You may then load the "real" database; it will be loaded to a different disk location. The dummy database will "hide" the bad blocks.

IDM.E102

System is going off line, please exit.

Explanation: The database server is about to halt operations. No new commands will be accepted.

User Action: Exit the current database server process immediately. Contact your local system staff for information about why the database server is being taken off line.

IDM.E103

Bad tape file number <filenum>.

Explanation: An error was encountered involving database server tape. Either file number <filenum> could not be found at all, or the access mode specified was not legal for the file. (You can only append or overwrite the file, and if a new name is requested you can only overwrite.)

User Action: Correct the command and try again.

IDM.E104

Bad tape transport number <drivenum>.

Explanation: An attempt was made to access database server tape using an invalid tape transport number, <drivenum>. The transport must be specified by an integer between 0 and 7, inclusive, and the transport must be connected.

User Action: Determine the correct number for the tape transport that you wish to access, and then try the command again. Contact your local systems staff if you need assistance.

IDM.E105

Bad tape name. Current tape name <tpname>.

Explanation: A database server tape operation was requested with an option set that required the database server to check the name of the tape. The given name did not match <tpname>, the name on the tape.

User Action: You may be trying to access the wrong tape. If so, obtain the correct tape. Otherwise, re-run the operation without specifying the name-checking option.

IDM.E106

Tape out of sequence. Current sequence number <your seqnum>.

Explanation: An attempt was made to access a tape that was not in the proper sequence. The correct sequence number is <correct seqnum>.

User Action: You are probably trying to use the wrong tape. Find and use the correct tape.

IDM.E107

Invalid tape header or trailer record.

Explanation: An attempt was made to access a tape that was in a bad format.

User Action: You are probably trying to use the wrong tape or a tape with an obsolete format. Find and use the correct tape.

IDM.E108

Tape transport in use.

Explanation: An attempt was made to use a tape transport that was already in use.

User Action: Select an unused tape transport, if available, or wait until the transport becomes available.

IDM.E109

End of tape encountered.

Explanation: An attempt was made to write beyond the end of the tape. The tape does not have enough space to hold the data.

User Action: Try again, using a reel with more space on it.

IDM.E110

Not enough page buffers currently.

Explanation: The database server did not have enough internal memory to write to tape. This cannot happen unless you have more than one tape drive but only 1/2 megabyte of memory.

User Action: Wait until more memory is available. You may have to do tape writing at times when the database server isn't so busy. Another alternative is to buy more memory.

IDM.E111

No tape controller.

Explanation: An attempt was made to access database server tape. There is no tape controller on this database server, so no tape functions are available.

User Action: Use database server files or host files instead, or buy a tape system.

IDM.E112

Tape unavailable: error number <errnum>.

Explanation: An error was reported by the tape controller of the database server. The number <errnum> indicates which of the following errors was encountered:

0 - tape not at load point; density/speed cannot be changed.
1 - tape is write-protected.
2 - tape transport is not on line.
3 - the previous rewind operation failed.

User Action: As appropriate for error <errnum>.

IDM.E113

Tape hard error: error number <errnum>.

Explanation: A hardware error was reported by the tape controller of the database server. This was caused by a hardware problem or by a bad tape. The number <errnum> indicates which of the following was encountered:

0 - hardware timeout.
1 - illegal command.
2 - tape controller parity error.
3 - tape formatter hard error.
4 - couldn't write filemark.
5 - data overrun.
6 - software timeout.

User Action: These are usually due to hardware errors, writing a bad tape, or reading a tape that has never been written. If you still have problems, contact your Britton Lee distribution support person.

IDM.E114

Tape caution: error number <errnum>.

Explanation: A warning was issued by the tape controller of the database server. The number <errnum> indicates which of the following errors was encountered:

0 - file mark was found on a read.
2 - tape was at load point when a rewind command was issued.
3 - tape was rewinding when a rewind command was issued.
4 - tape is already at desired speed/density.
5 - tape record shorter than expected.
6 - tape record longer than expected.

User Action: As appropriate for error <errnum>.

IDM.E115

Bad tape density <baddens>.

Explanation: An attempt was made to open a tape with a density of <baddens>. This is an invalid density specification. Valid specifications are "H" for high, "M" for medium, and "L" for low. The default is specified in the "configure" relation in the system database.

User Action: Determine the correct density of the tape, then try again.

IDM.E116

Too many aggregates.

Explanation: The database server is unable to handle more than 50 aggregates in a single query. That number was exceeded.

User Action: Try to simplify the query. You could break it down into several smaller, simpler queries.

IDM.E117

Too many reopens in this transaction. The limit is 7.

Explanation: An attempt was made to perform an eighth reopen command inside a transaction. This exceeds the allowable maximum.

User Action: Try to simplify the transaction.

IDM.E118

Children still active at the end of transaction.

Explanation: Usually a programmer error. All child database instantiations (produced by "reopen" commands) must be closed before the end of the transaction.

User Action: Correct the program as indicated.

IDM.E119

Parent already has updates - reopen denied.

Explanation: Usually a programmer error. A "reopen" is not allowed if the parent has already performed updates in the current transaction.

User Action: Correct the program as indicated.

IDM.E120

Database server file size exceeded.

Explanation: An attempt was made to read or write a database server file beyond the maximum file size. Files are limited to 2 to the 31st (2,147,483,648) bytes. This can be caused by a large offset on an idmread or idmwrite command, or by program error.

User Action: In idmread or idmwrite, use a smaller file offset. Otherwise, consult your local support staff.

IDM.E121

No more temporary relations available.

Explanation: Internally, the database server uses temporary relations. There is a limit of 16 temporaries, and a query caused this limit to be exceeded.

User Action: This error is likely to occur in a query with many aggregates and an "order by" clause and several "or" clauses. Simplify the query, perhaps breaking it up into several smaller queries.

IDM.E122

Extra value in insert: <value>.

Explanation: There were more values specified in an insert than there were columns in the table. This error should only be possible in SQL.

User Action: Coordinate the values to the insert.

IDM.E123

Insert - no value for column: <column>.

Explanation: There were not enough values specified in an insert to supply a value for column <column>. This error should only be possible in SQL.

User Action: Coordinate the values to the insert.

IDM.E124

Ambiguous column name: <column>.

Explanation: The column <column> occurs in more than one table mentioned in the query. You must specify the table name (or use table labels) to resolve the ambiguity. This error should only be possible in SQL.

User Action: Specify the name of the table that contains the <column> that you wish to access.

IDM.E125

Illegal index id <index id> for variable <var number>.

Explanation: Index number <index id> does not exist or does not apply to variable <var number>. You can not specify the use of this index.

User Action: Don't specify an index number, or choose an index number that is relevant.

IDM.E126

Temporarily out of memory. Please try again.

Explanation: All memory in the database server is in use. You will have to wait to run your query until the database server is less active. If this is a perennial problem, you may want to consider installing more memory on the database server.

User Action: Execute the query when the database server is less busy.

IDM.E127

Attempted update during database dump.

Explanation: Transactions cannot be updated during a database dump.

User Action: In IDL or SQL it is possible to perform a 'set "dumpwait"' command to postpone updates until the dump completes.

IDM.E192

Bad command/program <name>: query buffer currently too small.

Explanation: The stored command or program ("<name>") that you tried to execute was defined while the system was configured with a larger query buffer size.

User Action: It may be possible to redefine the command with the current buffer size. If not the system administrator must reconfigure the system with a larger query buffer to run this command.

IDM.E193

Cannot roll forward: relation <relname> has turned on logging since created.

Explanation: <relname> is missing a "create" record in the transaction log (eg. the relation was created without logging). Logging was turned on after the relation was created (with the "extend relation" command). Roll forward requires knowledge about the creation of <relname> to update the relation properly from the transaction log. Information will be lost about the relation or the database will become inconsistent.

User Action: Load and roll forward a later version of the transaction log.

IDM.E194

Constant or constant expression in outer join: <attr> <joinop> <attr>.

Explanation: One or both sides of the outer join operator: <joinop> did not contain any database variables.

User Action: Check that expressions containing <attr> and/or <attr> are not constants; they each must contain at least one reference to a field in a relation.

IDM.E195

Access to database server is denied.

Explanation: A valid password and/or host user name was not sent with the first open database command.

User Action: Send the correct password or host user name as required, or contact the system DBA to see that you have an entry in the system database login relation.

IDM.E196

Updaters active in database.

Explanation: There are users currently modifying the database.

User Action: If you specify the wait option to idmdump, the dump will be postponed until the updaters complete. Updates that were initiated after the dump command be postponed or aborted. This error message will still appear.

IDM.E197

Can't change density in the middle of a tape.

Explanation: An attempt was made to change the density while appending to the tape. If the tape drive is configured to NOREWIND mode, density changes are not allowed.

User Action: Run the command again without resetting the tape density. If the density must be changed, contact the system DBA.

IDM.E198

Corrupt index <indid> at byte offset <offset> on relid <relid> during load.

Explanation: Load database discovered an inconsistent index <indid> on relid <relid>.

User Action: The clustered index of this relation must be created before this relation is used. All nonclustered indices will then also need to be recreated. CKDB should be run on this database after the indices have been created.

IDM.E199

Not enough memory for dump bit map.

Explanation: The user tried to do an on-line dump on a database server which doesn't have enough memory for the on-line dump bit map.

User Action: Either install more memory in the database server or do read-only dumps.

IDM.E200

Trap, syserr or system call syserr encountered.

Explanation: While running, this process generated a trap, syserr or system call syserr. The system managed to clean up after the problem, but the last transaction was aborted.

User Action: Inform your system manager that you received this error. Then you can continue as you would if you received any other user error.

IDM.E201

Out of space in the "lock" table trying to lock relid <relid>.

Explanation: The "lock" table used by all processes running on the system has no available space for a lock which this process is trying to set.

User Action: Try running the query again when there are fewer processes running (other processes may have many locks), or ask the system administrator to configure the system to allow fewer page locks before promoting to relation locking.

IDM.E202

No space left in transaction log.

Explanation: While your update was executing, the transaction log ran out of space. Your transaction has been aborted because of this problem. The hard-allocated transaction log for the database is full, and another page cannot be allocated for it.

User Action: Inform your system manager that the transaction log is out of space. Run

"extend database" with the "logblocks" option and demands extends on the appropriate disk(s). The transaction log will probably need to be dumped before you can resubmit your update.

IDM.E203

Blocked locking relid <relid> on locks held by dead process <process number>.

Explanation: Another process, which has died due to a system fault, has locks which are blocking this process.

User Action: Try rerunning the query so that a less restrictive type of locking on relation <relid> is required (such as page rather than relation locking), or ask the system administrator to reboot the system to remove the locks held by process <process number>.

IDM.E204

No order number specified with index=<index id> plan for range variable <range var>.

Explanation: An order number must be given when using index plans (range of .... with dindex=.., dorder=..)

User Action: Redeclare the range variable and include an order number.

IDM.E205

Both files of a configured on-line dump can not go to host.

Explanation: You tried to dump both the database and the transaction log to host without specifying the dump type with the database server configured (L tuple in the "configure" relation) for on-line dumps.

User Action: Change the destination of the database or log file from the host to the database server, specify whether you want a read-only or on-line dump on the idmdump command line or remove the L tuple from the "configure" relation if appropriate.

IDM.E206

Bad dump source file: expected file of type "<type>", received file type "<yourtype>".

Explanation: While loading a database or transaction log, the database server detected that the file specified to load was not an appropriate file type. A file to be loaded must have been created by a dump.

Type	Description
0	Unknown file type
1	Database dump
2	Transaction log dump

Dumps using the "online" dump option will reverse the order the files will be written (database first, transaction log second) from that of the standard dump (log then database). When writing to tape note that the first file becomes the database, and the second the transaction log.

User Action: Try the load again using a different file, or, if online dump was used make sure that you are not using the log when you want the database.

IDM.E207

Load encountered <fixblkcnt> bad page(s), check database server console for details.

Explanation: Load database encountered data page(s) which do not have the page number of the next page indicated in the "blockalloc" relation in the dump. The console message(s) have the following variables (in decimal): Expected rel 3031, pg 20593; got rel 7, pg 20597; newdbpg 21133 in <dbname> Expected relid and page are from the "blockalloc" relation in the dump. The actual relid and page are the values on the page header. The new page in "<dbname>" is the page number in the database being loaded. This is the same as the expected page if the database is loaded into the same location it was dumped

from.

User Action: The new page reported on the console may have been corrupted prior to the dump database or missing due to tape hard error. Load database attempted to fix the corruption. CKDB must be run on both relations indicated in the error and the blockalloc relation and also dump the new page and verify the data.

IDM.E208

Feature not support in current release, token <token>.

Explanation: The user has attempted to use a feature which is not currently supported.

User Action: See your system administrator or salesperson to see if the feature indicated by token <token> is available in another release.

IDM.E209

Out of order data was found when using the "nosort" option.

Explanation: When the nosort option is used with create index, the data tuples must already be ordered by the index keys.

User Action: The relation is not presently ordered by the given index keys. This index may be created without the "nosort" option.

IDM.E210

Error <error> on page <page>: pageid <pageid> has bad page format at <value>.

Explanation: The verify pass of idmcopy in detected bad page formatting information or a bad tuple on pageid <pageid> (the first data page copied is page 0) at <value>. Error <error> on page <page>.

Type	Description
2	freeoff <value> is wrong
3	freeoff <value> extends into tuple table
4	page has tuple <value> which is >= nexttno
5	tuple number table does not match tuple at offset <value>
6	tuple at offset <value> has an illegal attribute table
7	tuple number table refers to missing tuple <value>
8	nexttno <value> is wrong

User Action: Check the page header or the attribute offset table of the offending tuple or the tuple offset table at the bottom of the page for incorrect information.

IDM.E211

A zero length file was specified to be written to database server tape.

Explanation: A zero length file cannot be written to tape.

User Action: Verify that the correct file was specified. If the correct file has a size of zero, it cannot be written to tape.

IDM.E212

Illegal operation requested for the tape drive.

Explanation: A request was made for an operation that cannot be done on a 1/4" tape drive. The usual requests of this type are appending to a tape, or dumping both a database and a transaction log to tape.

User Action: Correct the request so that the drive is able to perform the operation, or select a 1/2" tape drive at the "filename:" prompt if a drive of that type is available.

IDM.E213

Too many relids.

Explanation: The user attempted to create a new relation or define a new view or stored command and the database already has the maximum number of allowable

relids. The maximum number of relids is 32767.

User Action: Destroy an already existing relation, view or stored command first, then create the new relation, view or stored command.

1.2. Front End Errors

These errors are generally caused by errors in the front-end software as opposed to end-user errors. They will be generated as Abort-level severity messages.

IDM.E128

Bad tree node: <badnode>.

Explanation: This usually signifies an internal program error. Information sent to the database server was badly formatted or invalid.

User Action: Contact your Britton Lee distribution support person.

IDM.E129

Ran out of symbols while building tree.

Explanation: This signifies an internal program error. The information to the database server was badly formatted or invalid.

User Action: Contact your Britton Lee distribution support person.

IDM.E130

Result variable does not exist: <varname>.

Explanation: This usually signifies an internal program error. Information sent to the database server was badly formatted or invalid.

User Action: Contact your Britton Lee distribution support person.

IDM.E131

Target list is malformed: <target>.

Explanation: This usually signifies an internal program error. Information sent to the database server was badly formatted or invalid.

User Action: Contact your Britton Lee distribution support person.

IDM.E132

Bad node in qualification <node>.

Explanation: This usually signifies an internal program error. Information sent to the database server was badly formatted or invalid.

User Action: Contact your Britton Lee distribution support person.

IDM.E133

Range number for <rangenum> is not declared or greater than 15: <totalused>.

Explanation: This usually signifies an internal program error. Information sent to the database server was badly formatted or invalid.

User Action: Contact your Britton Lee distribution support person.

IDM.E134

Duplicate range number: <rnum>.

Explanation: This usually signifies an internal program error. Information sent to the database server was badly formatted or invalid.

User Action: Contact your Britton Lee distribution support person.

IDM.E135

Bad option bit on index command.

Explanation: This usually signifies an internal program error. Information sent to the database server was badly formatted or invalid.

User Action: Contact your Britton Lee distribution support person.

IDM.E136

Bad node in index command: <node>.

Explanation: This usually signifies an internal program error. Information sent to the database server was badly formatted or invalid.

User Action: Contact your Britton Lee distribution support person.

IDM.E137

Tree must be one variable.

Explanation: This usually signifies an internal program error. Information sent to the database server was badly formatted or invalid.

User Action: Contact your Britton Lee distribution support person.

IDM.E138

Domain <domain> in order clause is non-existent.

Explanation: This usually signifies an internal program error. Information sent to the database server was badly formatted or invalid.

User Action: Contact your Britton Lee distribution support person.

IDM.E139

Extra symbols in tree: <symbol>.

Explanation: This usually signifies an internal program error. Information sent to the database server was badly formatted or invalid.

User Action: Contact your Britton Lee distribution support person.

IDM.E140

Unknown command: <command>.

Explanation: This usually signifies an internal program error. Information sent to the database server was badly formatted or invalid.

User Action: Contact your Britton Lee distribution support person.

IDM.E141

Symbol <symbol> appeared where TYPE was expected.

Explanation: This usually signifies an internal program error. Information sent to the database server was badly formatted or invalid.

User Action: Contact your Britton Lee distribution support person.

IDM.E142

<type> is not a valid type symbol.

Explanation: This usually signifies an internal program error. Information sent to the database server was badly formatted or invalid.

User Action: Contact your Britton Lee distribution support person.

IDM.E143

Symbol <symbol> appeared where WITH was expected.

Explanation: This usually signifies an internal program error. Information sent to the database server was badly formatted or invalid.

User Action: Contact your Britton Lee distribution support person.

IDM.E145

No command was specified.

Explanation: This usually signifies an internal program error. Information sent to the database server was badly formatted or invalid.

User Action: Contact your Britton Lee distribution support person.

IDM.E146

No such dbin: <dbin>.

Explanation: This usually signifies an internal program error. Information sent to the database server was badly formatted or invalid.

User Action: Contact your Britton Lee distribution support person.

IDM.E147

Illegal file mode.

Explanation: This usually signifies an internal program error. Information sent to the database server was badly formatted or invalid.

User Action: Contact your Britton Lee distribution support person.

IDM.E148

Bad file number.

Explanation: This usually signifies an internal program error. Information sent to the database server was badly formatted or invalid.

User Action: Contact your Britton Lee distribution support person.

IDM.E149

Attribute names and formats do not agree in number.

Explanation: On a copy in command, there was a mismatch between the data copied in and the data stored in the relation. This usually results from copying data into a relation other than the one from which it was copied out.

User Action: Copy into the correct relation.

IDM.E150

Bad token in copyin: expected <token>, read <token>, at byte <location>.

Explanation: This usually signifies a trashed input data file. Information sent to the database server was badly formatted or invalid.

User Action: If copying in using the DBA utility "idmcopy", check that your data file is a data file made using idmcopy out or contains data as expected with the copy in command. This may indicate an internal error for other programs such as "idmfcopy".

IDM.E151

Extra bytes in command.

Explanation: This usually signifies an internal program error. Information sent to the database server was badly formatted or invalid.

User Action: Contact your Britton Lee distribution support person.

IDM.E152

Illegal BCD number <number>.

Explanation: This usually signifies an internal program error. Information sent to the database server was badly formatted or invalid.

User Action: Contact your Britton Lee distribution support person.

1.3. Done Bits

These messages are generated by bits in the DONE packet.

IDM.DONE.BADBCD

Bad BCD number rejected.

Explanation: A BCD number was sent from the host in a bad format.

User Action: Check your results carefully.

IDM.DONE.COUNT

<count> tuple<s> affected.

Explanation: A total of <count> tuple<s> were affected by the query (or in the case of create or extend database, <count> page<s> were allocated).

User Action: none required.

IDM.DONE.DIVIDE

Divide by zero.

Explanation: A divide by zero was detected during an arithmetic operation.

User Action: Check that the denominator does not evaluate to zero. Be sure that expressions or attributes are non-zero.

IDM.DONE.DUP

Duplicates deleted.

Explanation: You appended or replaced tuple(s) having key values for a unique index that already existed. The new duplicate tuple(s) were ignored in favor of the old data.

User Action: If the new data is desired, remove the tuple with the same value key then replace or append the data.

IDM.DONE.INTERRUPT

Command was interrupted.

Explanation: The command was interrupted during processing.

User Action: none required.

IDM.DONE.INXACT

In transaction.

Explanation: You are currently in a transaction.

User Action: The "end transaction" command will cause any updates to be committed, and "abort transaction" will cause updates to be backed out.

IDM.DONE.LOGOFF

Please log off.

Explanation: The database server has been set into "safe" mode. Further queries will be rejected. Please stop using the database server immediately.

User Action: Log off.

IDM.DONE.OVERFLOW

Overflow detected.

Explanation: The database server detected an arithmetic overflow during processing.

User Action: Check that your results are correct. It may be necessary to change the size of the result field.

IDM.DONE.ROUND

Rounding occurred on a BCDFLT.

Explanation: During processing of an arithmetic expression, a BCDFLT number was rounded.

User Action: Check that the results are correct.

IDM.DONE.TIMER

Query time: <time>/60 seconds

Explanation: The time to process the query, as returned by option 5 or option 11, was <time>/60 seconds.

User Action: none required.

IDM.DONE.UNDERFLOW

Underflow detected.

Explanation: The database server detected an arithmetic underflow during processing.

User Action: Check your results.

IDM.DONE.XABORT

Transaction aborted.

Explanation: The database server aborted the transaction. This is usually because the user requested an abort. All operations performed since the beginning of the transaction have been backed out (undone).

User Action: none required.

1.4. Measurement Messages

The following messages report query performance information. They are returned by the database server with MEASURE tokens.

IDM.M33

Response time was <ticks>/60 seconds.

Explanation: The wall clock time required to run the last command is reported in 60ths of a second. The time is measured from when the database server actually started processing the command until it finished the command. This time is equivalent to the sum of the cpu time and time spent waiting for various resources.

User Action: None required.

IDM.M34

Cpu use was <ticks>/60 seconds.

Explanation: The total CPU time required to complete the last command is reported in 60ths of a second. If the database server has a DAC, its time is included in the total.

User Action: None required.

IDM.M37

Input wait was <ticks>/60 seconds.

Explanation: The total time the database server spent waiting for the host to transmit data is reported in 60ths of a second. This time is usually zero except for commands which transmit a significant amount of data such as copyin, load database, load transaction or file write.

User Action: None required.

IDM.M38

Memory wait was <ticks>/60 seconds.

Explanation: If there is no memory available when a command arrives, it must wait until another command finishes. The total time spent waiting for memory before starting a command is reported in 60ths of a second.

User Action: None required.

IDM.M39

Cpu wait was <ticks>/60 seconds.

Explanation: The total time spent waiting for the CPU is reported in 60ths of a second. The CPU is timeshared.

User Action: None required.

IDM.M40

Disk wait was <ticks>/60 seconds.

Explanation: The total time spent waiting for disk I/O is reported in 60ths of a second.

User Action: None required.

IDM.M41

Tape wait was <ticks>/60 seconds.

Explanation: The total time spent waiting for database server tape I/O is reported in 60ths of a second.

User Action: None required.

IDM.M42

Output wait was <ticks>/60 seconds.

Explanation: If the database server command is generating output faster than the host is reading the data, the database server command will wait for a portion of the output to drain. The time spent waiting for the output to drain is measured in 60ths of a second.

User Action: None required.

IDM.M43

Blocked wait was <ticks>/60 seconds.

Explanation: A command wishes to acquire a lock on a portion of the database but another command has a conflicting lock. The total time spent waiting to acquire locks is reported in 60ths of a second.

User Action: None required.

IDM.M44

Accelerator use was <ticks>/60 seconds.

Explanation: If the database server has a DAC, this is the CPU time spent in the DAC. If there is no DAC, this is the CPU time spent simulating the DAC's functions measured in 60ths of a second.

User Action: None required.

IDM.M45

Output buffer wait was <ticks>/60 seconds.

Explanation: A command is trying to send data back to a host, but no buffer space is immediately available. The time spent waiting for buffer space to become free is reported in 60ths of a second.

User Action: None required.

IDM.M46

Number of disk cache hits was <hits>.

Explanation: This is the number of disk reads that were avoided because a disk page was already in cache memory. This number covers reads of both user data and system data.

User Action: None required.

IDM.M47

Number of disk reads was <reads>.

Explanation: This is the total number of disk reads required to process the last command.

User Action: None required.

IDM.M48

Number of soft tape errors was <errorcount>.

Explanation: If the last command used a database server tape drive, the total number of soft errors is reported. Soft errors are errors which were overcome by retrying the command. A large number indicates that either the tape transport needs servicing or the quality of the tape itself is poor and the tape should be replaced.

User Action: None required.

IDM.M49

Query space used: <query space used> bytes

Explanation: The last command used <query space used> bytes of query buffer space. This may be used to determine whether the database server should be reconfigured for a larger query buffer.

IDM.M60

Plan: qtype=<query type>, strategy=<proc method>, relation=<relation>, index=<index id>

Explanation: The decomposition plan used for this query is described. The arguments are:

qtype — ROOT or AOPxxx, where "xxx" is AVG, MAX, or one of the other aggregates.

strategy — the processing method: one variable, two variable, tuple substitution, or relation scan.

relation — the name of the relation being processed or "TEMP".
index — the id number of the index that was used to process this query, if any.

Mortals are not expected to understand this.

User Action: None required.

2. IDMLIB Messages

The following messages are generated by IDMLIB.

2.1. Asynchronous Messages

These messages may be generated asynchronously, i.e., not as the direct result of processing. Typically they result from a user action.

IDMLIB.ASYNC.INT

Keyboard interrupt.

Explanation: The user has interrupted the program.

User Action: None required.

IDMLIB.ASYNC.NOFP

Cannot do floating point on this machine.

Explanation: You have tried to perform an operation that would require the use of floating point ("real", as opposed to integer) arithmetic. The machine cannot support floating point.

User Action: Either eliminate the floating point values or attributes in your query or tell your management to buy floating point hardware.

IDMLIB.ASYNC.TERM

Software termination signal.

Explanation: A termination signal has been detected.

User Action: None required.

2.2. Conversions

The following messages may be generated by the conversion routines.

IDMLIB.CNVT.ATOBIN

"<character>" is not a valid binary character.

Explanation: Binary characters have values "0" to "9", "a" to "f", or "A" to "F".

User Action: Check the string which is being converted for characters which are not in the above ranges.

IDMLIB.CNVT.BADTYPE

Illegal type for conversion: <type>.

Explanation: The specified type is unrecognized by the conversion routine.

User Action: Check the documentation for the legal data types accepted.

IDMLIB.CNVT.CANT

Cannot convert type <intype> to <outtype>.

Explanation: The type conversion routine does not support conversion from type <intype> to type <outtype>.

User Action: You must change either the input type or the output type so that the conversion is possible.

IDMLIB.CNVT.GARBAGE

Garbage on "<string>" deleted converting to <targtype>.

Explanation: When converting the string "<string>" to type <targtype>, trailing characters that could not be interpreted as numeric data were deleted. The leading number was still converted. If there was nothing that could be interpreted as a number, zero was used.

User Action: Check the input string for typos or extra characters.

IDMLIB.CNVT.NODSC

Descriptor types are not supported.

Explanation: A program has tried to use a "descriptor type." This is reserved for systems that support many data types, such as packed decimal, etc. This system does not support any of these types.

User Action: Ask the programmer to correct his program.

IDMLIB.CNVT.OVERFLOW

Conversion overflow on "<string>".

Explanation: An attempt to convert "<string>" failed because there is not enough room to store the result. The maximum magnitude of the result is <maxvalue>. Conversion of unsigned hex and octal numbers is not supported. Integers will be left with a strange value; character strings will be truncated.

User Action: Check your results carefully to see which of them are incorrect or check the input for "<string>".

IDMLIB.JOB.CONTINUE

User process resumption.

Explanation: The user has resumed the program.

User Action: None required.

IDMLIB.JOB.SUSPEND

User process suspension.

Explanation: The user has suspended the program.

User Action: None required.

2.3. General Messages

IDMLIB.CANTFORK

Cannot execute system command: <reason>.

Explanation: It was not possible to create a subprocess to run a command because <reason>.

User Action: If the reason is not self explanatory, ask your system support staff for help.

IDMLIB.CLOCK.BAD

Bad date "<datestring>".

Explanation: The routine that parses dates was unable to parse the string "<datestring>", either because the syntax was bad or because the resulting date was not valid.

User Action: Try again, but check your format. If it still doesn't work, try using some other common date format. Make sure you are specifying a legitimate date — "2/29/83" will generate this error.

IDMLIB.MAPSYM.NOSYM

Symbol "<prefix><symname>" not found.

Explanation: The symbol "<prefix><symname>" is not present in the symbol database. The "<prefix>" indicates the context in which the symbol was used (e.g., as a trace flag) and was not part of your command. "<symname>" is the symbol you actually entered.

User Action: Correct the symbol name "<symname>".

IDMLIB.PEXTRACT.SYNTAX

Illegal syntax for parameter: "<params>".

Explanation: Probably a programmer error. The parameter string "<params>" was in a bad format.

User Action: Correct the parameter syntax. The correct syntax is a comma-separated list of "name(value)" pairs. Parameters are not required to have a value. For example, the following are legal parameter lists:

```
mode(r)
mode(w),rs(132),linebuffer
```

IDMLIB.RECOMPILE

This program must be recompiled.

Explanation: This program is trying to access a different version of IDMLIB than is currently installed on this system. The program must be recompiled from the source before it can be run.

User Action: Recompile and relink the program.

IDMLIB.SYSERR

Internal error: <message>.

Explanation: An internal error was detected in the current program.

User Action: Contact your Britton Lee distribution support person.

IDMLIB.TEMPNAME.NOFILES

Out of temporary file names.

Explanation: The system has tried to create a temporary file, but there were no more names available for such a file.

User Action: The program has probably opened more temporary files than it needs. Try to simplify the task.

IDMLIB.XALLOC.SIZE

Illegal memory allocation size <size>.

Explanation: Your program has requested <size> bytes of memory from the allocation module. Unfortunately, it is only possible to allocate positive quantities of memory.

User Action: Consult the programmer.

2.4. BCD Messages

These messages are be generated by the BCD (Binary Coded Decimal) subsystem. These routines do BCD arithmetic and conversion. For more information on BCD numbers, consult the System Programmer's Manual.

IDMLIB.BCD.TYPE

Bad BCD type (<type>).

Explanation: The program attempted a BCD operation on a non-BCD operand (of type "<type>"). The type must be BCD or BCDFLT. This is often a programmer error.

User Action: Check that the BCDNO variable has been initialized.

IDMLIB.BCD.LEN

Bad BCD length (<length>).

Explanation: The program attempted a BCD operation on an operand that has a length of <length>. This is an invalid length — the length must be between zero and 31. Normally a programmer error.

User Action: Check that the BCDNO variable has been initialized.

IDMLIB.BCD.ROUND

BCD rounding error occurred.

Explanation: A BCD number had to be rounded in processing, causing a loss of precision.

User Action: Check your results to insure that they are what you expected. No further action need be taken.

IDMLIB.BCD.OVERFLOW

BCD overflow occurred.

Explanation: A number was generated which was too large to be represented in the computer. The largest possible value was used in its place.

User Action: Check your results to insure that they are what you expected.

IDMLIB.BCD.UNDERFLOW

BCD underflow occurred.

Explanation: A number was generated which was too small to be represented in the computer. Zero was used in its place.

User Action: Check your results to insure that they are what you expected.

IDMLIB.BCD.DIVZERO

BCD division by zero occurred.

Explanation: An attempt was made to divide by zero.

User Action: Check your data to insure that it is within range. If this seems acceptable, contact the person at your local site responsible for the program you are using.

IDMLIB.BCD.ILLEGAL

Illegal BCD number.

Explanation: An illegal BCD number was passed to one of the BCD routines.

User Action: This is probably due to a programmer error. Check that a valid BCD number is being supplied.

2.5. Support Subsystem

These messages come from the back-end support subsystem of IDMLIB.

IDMLIB.IDM.AGG.FORBID
Aggregate not allowed in <qry-clause>.
Explanation: This error should not be possible in IDL.
User Action: Please contact your Britton Lee distribution support person.

IDMLIB.IDM.ALL.NOTONE
Too many or too few tables specified for "*".
Explanation: This error should not be possible in IDL.
User Action: Please contact your Britton Lee distribution support person.

IDMLIB.IDM.BADARG.RANGE
Bad argument <argnumber> in <func>: Must be >= 0 and <= 255.
Explanation: Argument number <argnumber> to the function <func> was invalid, because the argument was not between 0 and 255.
User Action: Correct the syntax as suggested.

IDMLIB.IDM.BADARG.STAR
Function <arg1>: "*" not allowed as argument.
Explanation: This error should not be possible in IDL. Please contact your Britton Lee distribution support person

IDMLIB.IDM.BADARG.SUBSTR
Bad argument <argnumber> in <func>: Cannot be 0.
Explanation: Argument number <argnumber> to the function <func> was 0.
User Action: Correct the syntax as suggested.

IDMLIB.IDM.BADARG.TYPE
Bad argument <argnumber> in <func>: is a bad type.
Explanation: Argument number <argnumber> to the function <func> was an invalid type.
User Action: Correct the syntax as suggested.

IDMLIB.IDM.BADCONS
Bad constant type "<type>", expected "<type>".
Explanation: "<type>" are the only allowed constant type in this context.
User Action: Use a "<type>".

IDMLIB.IDM.BADDIREC
Illegal direction specifier "<direction>".
Explanation: The "order by" direction "<direction>" is not valid.
User Action: Directions may be "ascending" or "descending". "a" and "d" are accepted as abbreviations.

IDMLIB.IDM.BADESC
Illegal ESCAPE character specification.
Explanation: This error should not be possible in IDL. Please contact your Britton Lee distribution support person

IDMLIB.IDM.BADMODE
Bad mode "<mode>".
Explanation: Mode "<mode>" is not a valid mode for database server tape. Legal modes are "r", "w", and "a".
User Action: Correct the mode parameter.

IDMLIB.IDM.BADORDER

Bad order-by clause <badord>.

Explanation: An attempt was made to order the query output by <badord>. <badord> was not preceded by a range variable, and it was not found in the target list.

User Action: Either precede <badord> by a range variable, or include it in the target list.

IDMLIB.IDM.BADSUB

Too many targets in subquery.

Explanation: This error should not be possible in IDL. Please contact your Britton Lee distribution support person

IDMLIB.IDM.BADTYPE

Bad type "<typespec>".

Explanation: <typespec> was specified as a data type in a "create" command. <typespec> is not a recognized data type in "create".

User Action: Check for spelling errors and typos. Determine which types are legal for this command, and try again.

IDMLIB.IDM.BADWITHOPT

Illegal with option "<optname>".

Explanation: The option "... with <optname>" has been specified; this option cannot be recognized.

User Action: Check the documentation for legal options in this context.

IDMLIB.IDM.CANTBY

Cannot use a by list in function <funcname>.

Explanation: An attempt was made to specify a "by" clause in the function <funcname>. "By" clauses are not allowed in function <funcname>.

User Action: A "by" clause probably doesn't make sense in this context. Find out what the function does, and how to use it, and then try again. It may also be that you need to use a different function.

IDMLIB.IDM.CANTFIX

Cannot use "fixed <funcname>".

Explanation: An attempt was made to specify the "fixed" option for the function or aggregate <funcname>. The "fixed" option is allowed only for the functions "string", "bcd", and "bcdflt".

User Action: Correct the syntax of the query.

IDMLIB.IDM.CANTUNIQUE

Cannot use "unique <funcname>".

Explanation: An attempt was made to specify the "unique" option for the function or aggregate <funcname>. The "unique" option is allowed only for the aggregates "count", "sum", and "avg".

User Action: Correct the syntax of the query.

IDMLIB.IDM.CANTWHERE

Cannot use "where" clause in function <funcname>.

Explanation: An attempt was made to use a "where" option in the function <funcname>. The "where" option is not allowed in any function.

User Action: Correct the syntax of the query.

IDMLIB.IDM.CONSTTOOLONG

<Type> constant longer than <maxlen> bytes.

Explanation: A <Type> constant exceeded the maximum allowable length of <maxlen> bytes.

User Action: Shorten the constant to at most <maxlen> bytes.

IDMLIB.IDM.CRACKARGV.TEMPLATE

Bad Template: <reason>.

Explanation: The template passed to crackargv() was incorrect for the following reason: <reason>

User Action: This is always a programmer error. Contact your system support staff.

IDMLIB.IDM.EXEC.PARAM

Parameter <argnum> to stored command "<cmdname>" is neither a constant nor a substitution.

Explanation: Arguments to stored commands must be constants (or substitutions). You have tried to pass an expression as argument number <argnum> to command "<cmdname>".

User Action: Change the argument to be a constant (or a substitution).

IDMLIB.IDM.EXEC.PROGID

Illegal program identifier.

Explanation: An attempt was made to invoke a stored program with an invalid program identifier. Only integers are valid.

User Action: Usually a programmer error. Find the correct identification of the stored program that you wish to run, and try again.

IDMLIB.IDM.FIELDSIZE

Field size too large in "<typespec>".

Explanation: The field size specified in the attribute type "<typespec>" is greater than 255. Variable length fields are limited to a size of 255 by the database server..

User Action: If you really need a field larger than 255 characters, you must break it up into more than one attribute.

IDMLIB.IDM.GETHUNPW

User name and password required from this host.

Explanation: The database server requires that a user name and password be supplied to access this database. This information was not available, and it was not possible to access the terminal to ask for it.

User Action: Provide a user name and password by setting the IDMHUNAME and IDMPASSWD parameters respectively. Consult your users' manual for more information.

IDMLIB.IDM.GRAMMAR.OVERFLOW

Internal grammar stack overflow.

Explanation: This error should only occur during the parsing of the "create" command which contains too many relation partitions.

User Action: Check that the you are not exceeding the legal number of partitions.

IDMLIB.IDM.ILLEGPARAM

Illegal use of stored command parameter "<param name>": command <command>.

Explanation: Parameters to stored commands are only legal within a define or define program command. Parameter "<param name>" is not legal inside the <command> command.

User Action: Remove the '$' from before the variable "<param name>".

IDMLIB.IDM.ILLEGPRTCT

Illegal protection mode "<badmode>".

Explanation: An attempt was made to set a protection mode on a database object. The mode given was <badmode>, which is not a legal protection mode. Valid modes are: "read", "write", "create", "create database", "create index", "execute", "read tape", "write tape", "all tape", and "all".

User Action: Try again, using a legal mode for the command.

IDMLIB.IDM.ITXCMD

No name specified to itxcmd().

Explanation: Itxcmd() has been called with no command name parameter.

User Action: This is a programmer error. Make sure the command name parameter is correctly passed.

IDMLIB.IDM.ITXSETP.NOTREE

No command tree specified to itxsetp();

Explanation: Itxsetp() has been called with no command tree parameter.

User Action: This is a programmer error. Make sure the command tree parameter is correctly passed.

IDMLIB.IDM.ITXSETP.BADTYPE

Illegal tree type <treetype> for itxsetp().

Explanation: The command tree specified to itxsetp() is of type <treetype>. Only EXEC or EXECP are valid command types.

User Action: This is a programmer error. Make sure that a valid command tree is specified to itxsetp(). The tree should be created using the itxcmd() or the itxprog() functions.

IDMLIB.IDM.LONGNODE

<nodetype> node too long; only <length> bytes sent.

Explanation: The contents of a <nodetype> node is longer than the database server allows. It has been truncated to a length of <length> bytes.

User Action: Any character strings longer than <nodetype> characters should be shortened or broken into more than one string.

IDMLIB.IDM.LONGTOKEN

Token too long: <token>.

Explanation: A word has been entered that is longer than can be handled. This word begins "<token>".

System Action: The first <maxlen> characters were used; the rest were thrown away.

User Action: Shorten the long word.

IDMLIB.IDM.MAPC.ESCAPE

Bad pattern matching string: <badstr>.

Explanation: The string <badstr> was given as a pattern matching string, but is malformed.

User Action: Check the syntax of the string. A full description of pattern matching can be found in the IDL Reference Manual.

IDMLIB.IDM.MATCHLST.NOMAT

Wrong number of names specified.

Explanation: This error should not be possible in IDL.

User Action: Please contact your Britton Lee distribution support person.

IDMLIB.IDM.MOREERRS

Error information has exceeded the <maxerrbytes> byte limit and has been discarded.

Explanation: Error information has been discarded because more information was returned than could be stored.

User Action: Contact your Britton Lee distribution support person.

IDMLIB.IDM.NOTABLE

No table specified for "<colname>".

Explanation: This error should not be possible in IDL.

User Action: Please contact your Britton Lee distribution support person.

IDMLIB.IDM.NOITAPEOP

User input may be required with database server tape.

Explanation: If this job requires more than one tape, all work will be lost without operator control to load tapes.

User Action: User response to queries is required when using database server tape. Run the job in the foreground or interactively.

IDMLIB.IDM.NOTCOMMAND

Illegal root node <node> of command tree.

Explanation: The root node of this command tree is type <node>. It must be of type iCOMMAND.

User Action: This is probably a programmer error. Check that the address of the tree is valid.

IDMLIB.IDM.NOTFUNC

"<name>" not an aggregate or function.

Explanation: The IDL parser found "<name>" where it was expecting to find an aggregate or a function. "<name>" is not a known aggregate or function. Frequently, this is caused by omitting the quotation marks enclosing a name.

User Action: Check for spelling errors, typos, or missing quotation marks and try again.

IDMLIB.IDM.NOTINT

type <constant type>: Incorrect type for substitution name.

Explanation: Substitution names must be integer or string values. Type "<constant type>" is not a valid type.

User Action: Quote the substitution name.

IDMLIB.IDM.NULLTREE

<func>: null command tree.

Explanation: This is a programmer error. "<func>" has been called on a null tree.

User Action: Make sure that the tree is not being accidentally trashed.

IDMLIB.IDM.NUMARGS.TOOFEW

Function <func>: Too few arguments (<goodcount> required).

Explanation: The function <func> was called with not enough arguments. <func> can only be called with <goodcount> arguments.

User Action: Try the query again, correcting the problem. Make sure you understand the meaning and use of the <func> function.

IDMLIB.IDM.NUMARGS.TOOMANY

Function <func>: Too many arguments (<goodcount> required).

Explanation: The function <func> was called with too many arguments. <func> can only be called with <goodcount> arguments.

User Action: Try the query again, correcting the problem. Make sure you understand the meaning and use of the <func> function.

IDMLIB.IDM.OBJECT.SYNTAX

Bad type <type> for object name.

Explanation: Object names (e.g., the name of a relation to destroy) may be simple names or quoted strings. You have used a constant of type <type> instead. For example, the commands

```
destroy myrelation
destroy "myrelation"
```

are both legal, but

```
destroy 17
```

is not legal, since the object name is neither a simple name or a quoted string.

User Action: Put quotation marks around the name if you want it to be used as a text string, e.g.:

destroy "17"
to destroy the relation, command, or file with the name "17".

IDMLIB.IDM.OBJGIVEN
An illegal object was given with <command>.
Explanation: In the protection command (<command>) an object was specified. Objects are not allowed for "create", "create database", "create index", "all tape", "read tape", or "write tape". Objects are required for "read", "write", "execute", and "all".
User Action: Remove the object from the command, or be sure to use the "to" keyword before the "user".

IDMLIB.IDM.OPT.ILLEGAL
Illegal option value (<optvalue>).
Explanation: The second argument to set must be an integer between 0 and 127.
User Action: Make sure that you are not putting quotation marks around the argument and the value is between 0 and 127.

IDMLIB.IDM.OPT.NOTSET
Option <optname> is not set.
Explanation: You have attempted to unset an option which has not been set.
User Action: None required.

IDMLIB.IDM.OPT.SET
Option "<optname>" already set.
Explanation: You have attempted to set an option which is already set.
User Action: None required.

IDMLIB.IDM.OPT.TAPE
Cannot set database server tape option <optvalue>.
Explanation: The database server tape option may not to be set in this context. If this is option is to be set in a user program, use the itaddopts or itapeopts interface instead.
User Action: Unset option <optvalue> and use the proper interface.

IDMLIB.IDM.OPT.TOOMANY
Too many options set already.
Explanation: The user has attempted to set too many options.
User Action: "unset" any options which are not currently in use.

IDMLIB.IDM.PERMDENY
No object was specified with command <command>
Explanation: In a protection command ("permit" or "deny") an object was not specified. Objects are not required for "create database" "create index", "read tape", "write tape", or "all tape"; objects are required for "read", "write", "execute", and "all".
User Action: Specify the object that the command is supposed to affect.

IDMLIB.IDM.RANGE.BADNO
Bad range variable number <rvno>.
Explanation: A range variable number was used that is not within range or does not refer to a currently defined range variable number. This is an internal system error.
User Action: Contact your Britton Lee distribution support person.

IDMLIB.IDM.RANGE.BADOPT
Bad range option "<option>".
Explanation: An attempt was made to set option "<option>", which is not applicable to a range statement. For example, you may have tried to perform the command

"range of x is R with demand = 100"; demand does not make sense in this context.

System Action: The option will be ignored.

User Action: Change the range statement in error and try again.

IDMLIB.IDM.RANGE.GRAB

Range variable <rangevar 1> replaces "range of <rangevar 2> is <idm-object>".

Explanation: Range variable <rangevar 2>, which was declared to range over relation <idm-object>, has been replaced with range variable <rangevar 1>. This can occur either because the user reused the same range variable name (but on a different relation) or because the range table was full and an old entry had to be taken to process this range.

User Action: Normally none required. If the range variable which was replaced was important, the original range statement will have to be reissued.

IDMLIB.IDM.RANGE.ILLEGOPTVAL

Range option "<option>" does not accept a value.

Explanation: Option "<option>" cannot accept a value associated with it.

System Action: The value will be ignored.

User Action: Remove the value from the range statement.

IDMLIB.IDM.RANGE.NEEDOPTVAL

Range option "<option>" requires a value.

Explanation: You have issued a range statement that includes the option "... with <option>". The "<option>" option requires a value.

System Action: The option will be ignored.

User Action: Add a value to the with clause. For example, change the range statement to "range ... with <option> = (value)".

IDMLIB.IDM.RANGE.NOTDECL

Range variable "<rangevar>" not declared.

Explanation: A reference was made to the range variable "<rangevar>". This range variable has not been declared, and so is not associated with any relation.

User Action: Declare the range variable. This is done in IDL by entering "range of <rangevar> is relname" where "relname" is the name of the relation you want to access.

IDMLIB.IDM.RANGE.TOOMANY

Too many range variables in single query; <used> used, <allowed> allowed.

Explanation: A single query required the use of <used> range variables. However, only <allowed> range variables may be used in a single query.

User Action: Check the query to see if you really need that many variables. If you do, you will have to find a way to break up the query into two or more smaller queries.

IDMLIB.IDM.SET.SYNTAX

Bad set constant type "<type>".

Explanation: An attempt was made to use a constant of type <type> in a "set" or "unset" statement. Integers and option names may be used in this context.

User Action: Use an acceptable data type.

IDMLIB.IDM.SQL.OBSOLETE

Obsolete syntax: "<violation>".

Explanation: This error should not be possible in IDL. Please contact your Britton Lee distribution support person

IDMLIB.IDM.SUB.NEEDVAL

Substitution value for "<subname>" has not been set.

Explanation: The placeholder "<subname>" must be assigned a value before "<subname>" can be used in a query.

User Action: This is a user error. Either remove the "<subname>" place holder from the query, or assign it a value.

IDMLIB.IDM.SUB.TYPE

Invalid type "<type>" in substitute of "<subname>".

Explanation: A value of type "<type>" was passed to be substituted for "<subname>" in the tree. This type was incompatible with the tree.

User Action: Use a different type of value.

IDMLIB.IDM.SUB.OBJTYPE

Invalid type "<type>" in substitute of object substitution name "<subname>".

Explanation: A value of type "<type>" was passed to be substituted for "<subname>" in the tree. Only strings are valid in this context.

User Action: Use a string value for the substitution of "<subname>".

IDMLIB.IDM.SUB.VAL

Value "<value>" out of range [<minvalue>, <maxvalue>] in substitute of "<subname>".

Explanation: When substituting for "<subname>", the value <value> was out of range. The value must be between <minvalue> and <maxvalue> to be valid.

User Action: Use a value that is in range.

IDMLIB.IDM.SYNTAX

Unrecognized syntax; last token read: "<token>".

Explanation: The computer could not understand the command you gave it.

User Action: Find the word "<token>" in your command. The error is probably immediately before or after that word. Check to see that what you typed makes sense in the IDL language. For a complete description of the IDL language, see the IDL Reference Manual.

IDMLIB.IDM.TAPE.NOOPER

No operator available for database server tape transport number.

Explanation: The database server requires a new transport number for each tape used.

User Action: Check that you are not running the program in the background or in batch mode.

IDMLIB.IDM.TAPE.OPABORT

Operator abort: <operator message>.

Explanation: The operator aborted your job. The reason given was: <operator message>.

User Action: Remedy the problem stated by the operator.

IDMLIB.IDM.TLOVFLOW

Target list overflowed internal list: maximum of <buffersize>.

Explanation: The list of targets overflowed while the target list was being converted from tree form.

User Action: This is an internal error. Contact your Britton Lee distribution support person.

IDMLIB.IDM.TRACE.SYNTAX

Illegal type <type> in trace statement

Explanation: You have specified a constant of type <type> in a trace statement. Integers and string constants are the only types allowed in this context.

User Action: Change the type of the constant.

IDMLIB.IDM.UTREE.BADVER

Unknown UTREE version code <treever>: current = <codever>

Explanation: Every UTREE (version of a query tree that can be saved across runs) has a version code. A UTREE with version number <treever> was passed to IDMLIB. This version of IDMLIB knows about version <codever>, so the UTREE could not be used.

User Action: Regenerate the UTREE using the correct version of IDMLIB. If you are using a precompiler, this may be done by recompiling.

IDMLIB.IDM.UTREE.TRASH

Trashed UTREE

Explanation: You have given IDMLIB something as a UTREE which appears to be trashed.

User Action: This is probably due to a program bug or a trashed file. Fix the program or repair the file.

IDMLIB.IDM.WITH

With option is not a constant.

Explanation: You have specified "... with (something) = nonconstant". The database server can only understand constant values.

User Action: Change the with option to use a constant. For example, the value 100 is a constant, but r.name is not a constant.

2.6. I/O Subsystem

These messages come from the I/O subsystem.

IDMLIB.IO.BADIFP

Invalid ifp: <reason>.

Explanation: The program tried to perform a file operation (ifread, ifwrite, ifcontrol, etc.) with an invalid ifp.

User Action: Usually a programmer error. Make sure that all operations use the ifp returned by ifopen.

IDMLIB.IO.BADIFVER

Incorrect ifp version: <yourver>. Current version: <currentver>.

Explanation: The program tried to perform a file operation (ifread, ifwrite, ifcontrol, etc.) with an ifp compiled under another version of IDMLIB.

User Action: Recompile all modules from source and relink to the library. Be sure to include the latest version of the idmlib include file (idmlib.h).

IDMLIB.IO.BADMODE

Illegal mode for <filetype> <filename>: "<modestring>".

Explanation: The program attempted to open the <filetype> <filename> with an illegal mode, "<modestring>".

User Action: Use one of the legal modes for <filetype>s.

IDMLIB.IO.BRSIZE

Buffer/record size mismatch on <filetype> "<filename>": bs(<bsize>), rs(<rsize>)

Explanation: The <filetype> "<filename>" has a buffer size of <bsize> but a record size of <rsize>. Records must fit into buffers without being split.

System Action: The buffer size is expanded to accommodate the record size.

User Action: Increase the value of the bs parameter.

IDMLIB.IO.BSIZE

size mismatch on <filetype> "<filename>", user <usersize>, file <filesize>

Explanation: The user has specified that the "bs" parameter is <usersize> for <filetype> "<filename>", but a status inquiry on the file indicates that the actual value for "bs" is <filesize>.

System Action: The user parameter will be used instead of the file parameter, but further errors may result.

User Action: Change the parameter to match the file parameter, or omit the specification of the parameter altogether.

IDMLIB.IO.CANTNAME

Illegal name "<filename>" on <filetype>.

Explanation: The file type <filetype> does not accept names. For example, it is not possible to use a name when opening a string or the scanner.

User Action: Fix the program.

IDMLIB.IO.CANTOPEN

Can't open <filetype> "<filename>": <errmsg>.

Explanation: The program was unable to open the <filetype> "<filename>" in the mode "<modestring>". The operating system gave the following reason: <errmsg>.

User Action: Rectify the situation indicated by the error message.

IDMLIB.IO.CRACK.BADTYPE

Illegal type in spec "<spec>".

Explanation: The file type specified in "<spec>" was not valid. Valid types are "hfile", "htape", "ifile", or "itape".

User Action: Use a valid file type specification.

IDMLIB.IO.CRACK.NAMETOOLONG

File name "<name>" too long.

Explanation: The file name "<name>" specified in a file specification was too long. The maximum length is <maxlen> characters.

User Action: Shorten the name.

IDMLIB.IO.ETHER.BADDT

Bad datastream type: got <got>, expected <expected>.

Explanation: A communications packet of type <got> was read in a context where <expected> as expected during a read from an ethernet connection to the database server. This is an internal error.

User Action: Report the problem to your Britton Lee IDM distribution ethernet support person.

IDMLIB.IO.ETHER.NOADDR

No name-address binding found for ETHERNET device <logicalname> in <bindings>.

Explanation: The logical name <logicalname> had no entry in the database <bindings>. With XNS, this name is specified by the system parameter XNSNAMES.

User Action: If using XNS, check that your setting of XNSNAMES specifies the file containing the name address bindings you want. If so, be sure that you have spelled the logical name correctly and that an entry is made in this file for the database server you are naming. If you are using TCP, check your spelling and contact your System Administrator if the problem continues.

IDMLIB.IO.ETHER.NOETHER

Ethernet not supported.

Explanation: You have attempted to connected to a database server named <idmname> using an ethernet-based protocol. However, your version of IDMLIB does not support ethernet protocols. This results from an internal error; report this to your Britton Lee distribution support person.

User Action: Report the problem to your Britton Lee distribution support person.

IDMLIB.IO.ETHER.NOSERVICE

Ethernet service "<servname>" not found.

Explanation: The service was not found in the system's database file.

User Action: Contact your system administrator to determine if this ethernet service has been defined.

IDMLIB.IO.HFILE.DELETE

Cannot delete host file "<filename>": <errmsg>.

Explanation: An attempt was made to delete the host file "<filename>". The deletion failed. The operating system reported "<errmsg>" as the reason for the failure.

User Action: This problem is related to the operating system. Find out why the file cannot be deleted and rectify.

IDMLIB.IO.HFILE.NULLRESET

Reset on host file "<filename>" ineffective.

Explanation: It is not effective to reset a file that is opened for append mode, since each write goes to the end of file.

User Action: Check your usage to see if this operation makes sense.

IDMLIB.IO.IDM.LIFELINE

Got lifeline socket for database server <host>.

Explanation: When attempting to open the database server named <host>, you got the reserved "lifeline socket" instead of a regular communications socket. This is caused when the "identify daemon" is not running.

User Action: Consult your local systems staff and ask them to start the identify daemon.

IDMLIB.IO.IDM.NODEVICE

Cannot open database server device "<device>": <reason>.

Explanation: The database server device "<device>" does not exist or could not be opened. This can occur if the device name is misspelled, the database server is currently unavailable or powered off, or if your system manager has disabled database server access.

User Action: Check your default database server device name or command line option for misspelling. If this seems correct, contact your local system administrator.

IDMLIB.IO.IDM.NODRIVER

Nonexistent Device Driver: legal values "<options>"

Explanation: The database server driver you have specified using the IDMDEV or IDMDRIVER parameter is unknown. Legal driver names are: <options>

User Action: Change IDMDEV or IDMDRIVER to indicate a valid driver. Consult your local system administrator if in doubt about legal values.

IDMLIB.IO.IDM.TIMEOUT

Dtabase server results timeout on device "<device>"

Explanation: The program sent a query to the database server, but failed to read the results before they were canceled by the database server. This is usually caused by starting up an interactive query and then leaving the terminal for an extended period of time. The timeout period is specified in the "configure" relation.

User Action: Rerun the query, getting the results promptly.

IDMLIB.IO.IFUNGETC

No room for ifungetc on <filetype> <filename>.

Explanation: The program was unable to perform an ifungetc operation on <filetype> "<filename>". This is usually a programmer error — it can occur if ifgetc() was never called, or if ifungetc() is called twice in a row.

User Action: Correct the program error.

IDMLIB.IO.IOERR

I/O error on <filetype> <filename>, record <recno>: <errmsg>.

Explanation: An input/output failure occurred on <filetype> "<filename>" on record number <recno>. The operating system reported "<errmsg>" as the reason for the failure.

User Action: Verify that "<filename>" is actually a <filetype>. Correct the situation, "<errmsg>", if it makes sense. Check for programmer error. Look up the message in the messages book for your operating system.

IDMLIB.IO.ISTAND.ALRDYOPN

Serial Port already open or serial driver used for parallel channel.

Explanation: Some other process is already using the standalone driver or IDMDRIVER is set to indicate the standalone driver while IDMDEV indicates the parallel driver.

User Action: Wait until the driver is free and try again after first checking that IDMDRIVER and IDMDEV agree as to which driver is to be used.

IDMLIB.IO.ISTAND.BADACK

An acknowledgement was garbaged and could not be corrected.

Explanation: A badly garbled acknowledgement byte was received by the serial standalone driver. A likely cause is an unreliable RS-232C line.

User Action: Check your RS-232C line for severe noise.

IDMLIB.IO.ISTAND.BADACKNO

ACKNO on data packet

Explanation: The database server sent an ACKNO to a data packet. This indicates a serious synchronization problem between the host and the database server.

User Action: Contact your Britton Lee distribution support person.

IDMLIB.IO.ISTAND.BADBAUD

Bad baud rate (<baudrate>) was selected.

Explanation: The baudrate (<baudrate>) you selected is incorrect for use with the standalone serial driver.

User Action: Try using a legal baud rate with your driver. Consult your Britton Lee distribution support person if you have problems determining a legal baud rate.

IDMLIB.IO.ISTAND.BADCAN

Cancel Failed

Explanation: The database server could not process a CANCEL or CANCELP request. This is probably cause by a problem with the database server or communication lines. It may be caused by the operating system's difficulty handling asynchronous events.

User Action: Check all communication lines and the status of the database server. Check that your operating system is handling asynchronous events correctly. Contact your Britton Lee distribution support person for assistance.

IDMLIB.IO.ISTAND.BADCLOSE

Close of Standalone Port failed: <system error>.

Explanation: The system reported <system error> in response to an attempt to close the serial port.

User Action: The port is probably messed up. Consult a systems guru.

IDMLIB.IO.ISTAND.BIGPKT

Packet size selected (<size>) is too large.

Explanation: The packet size (<size>) you selected for use with the standalone driver is too large to be used. A smaller packet size will be used automatically.

User Action: None. The driver does it all for you.

IDMLIB.IO.ISTAND.BADRD

Read error on standalone connection: <system error>.

Explanation: A read to the database server standalone connection failed. "<system error>" is the reason reported by the operating system.

User Action: Ask your system manager what <system error> means.

IDMLIB.IO.ISTAND.BADWRT

Write error on standalone connection: <system error>.

Explanation: A write to the database server standalone connection failed. "<system error>" is the reason reported by the operating system.

User Action: Ask your system manager what <system error> means.

IDMLIB.IO.ISTAND.ICEERR

Bad response (<error kind>) to database server communication request.

Explanation: The database server indicated that a communication request could not be processed because of a severe error. This is probably caused by a misuse of the database server driver. It could be caused by a noisy communications line.

User Action: Check that all requests to the database server are legal (i.e., no negative counts, etc.). If you detect no illegal requests, ask your Britton Lee distribution support person to help you decipher what <error kind> could mean in your application.

IDMLIB.IO.ISTAND.IDACKNO

An ACKNO was received while trying to identify the host to the database server.

Explanation: An ACKNO was received while the host was identifying itself to the database server. It should not be possible to receive an ACKNO in this circumstance.

User Action: Contact your Britton Lee distribution support person.

IDMLIB.IO.ISTAND.MAX

Retry Limit Exceeded

Explanation: The standalone driver was unable to get through to the database server. The database server may have crashed or the serial line may have been disconnected.

User Action: Check the status of the database server and the communication lines and try again.

IDMLIB.IO.ISTAND.MULTIUSE

Standalone serial port is in use as multi-user line.

Explanation: The serial port is being used as a multi-user port. It is not possible to use a port in single-user mode and multi-user mode at the same time.

User Action: Try using another free serial port or wait until the current port becomes free.

IDMLIB.IO.ISTAND.NOTOPN

Standalone device is not open prior to use.

Explanation: You tried to to read or write to the database server before opening the connection. You must open the connection before you can use it.

User Action: Check to see that your code opens the serial port prior to using it.

IDMLIB.IO.KEYED.BADFILE

File "<badfile>" not in keyed format.

Explanation: The program attempted to open the host file "<badfile>" for keyed access. "<badfile>" is not in the correct format.

User Action: Use the proper access methods on "<badfile>", or convert "<badfile>" to keyed format.

IDMLIB.IO.KEYED.NOKEY

Key required.

Explanation: The program called ifcontrol() to set the key for a read or write of a keyed file. However, no key was given.

User Action: This is usually a programmer error. Check the call to ifcontrol().

IDMLIB.IO.KEYED.DUPKEY

Duplicate key "<keystring>" found in "<filename>".

Explanation: The program attempted to set the key "<keystring>" for a subsequent write to the keyed file "<filename>". This key already exists in the file, and duplicate keys are not allowed.

User Action: If you want to add this data, you must either use a different key or must destroy the file and re-create it. There is no way to delete the old key.

IDMLIB.IO.KEYED.NOTFOUND

Key "<keystring>" not found in "<filename>".

Explanation: The program attempted to set the key "<keystring>" for a subsequent read from the keyed host file "<filename>". The key was not found.

User Action: There is no "<keystring>" key in the file.

IDMLIB.IO.LTAPE.ABORT

Labeled tape aborted.

Explanation: The operator aborted labeled-tape operations.

User Action: Contact your operations staff to determine why your tape operation could not be satisfied.

IDMLIB.IO.LTAPE.ACCESS

Cannot access volume "<volume-id>" — access code "<access-code>"

Explanation: The user has attempted to access volume "<volume-id>". The access code is "<access-code>". This implementation restricts access to protected volumes.

User Action: Use an unprotected volume.

IDMLIB.IO.LTAPE.BADMODE

Illegal mode "<mode>" for labeled tape.

Explanation: The program attempted to open the labeled tape with an illegal mode "<mode>".

User Action: Use one of the legal modes for labeled tape.

IDMLIB.IO.LTAPE.CANT

Cannot <operation> magnetic tape: <reason>.

Explanation: It was not possible to <operation> tape drive as required to access labeled tape because <reason>. This is probably caused by a hardware or operating system error. If the tape was being written, it may be possible that the tape is not readable by normal means.

User Action: Consult your systems staff.

IDMLIB.IO.LTAPE.CANTRESET

Cannot reset labeled tape.

Explanation: It is not possible to reset a labeled tape file because it could require mounting a previous volume.

User Action: If you really need to start the file at the beginning close the file and reopen it.

IDMLIB.IO.LTAPE.CANTREWRITE

Cannot rewrite labeled tape.

Explanation: It is not possible to rewrite a labeled tape file because it could require mounting a previous volume.

User Action: If you really need to rewrite the file at the beginning close the file and reopen it.

IDMLIB.IO.LTAPE.DENSITY

Illegal density <density> for labeled tape.

Explanation: An illegal tape density was passed in a "density" parameter of an open of a labeled tape.

User Action: Usually a programmer error. Correct the ifopen statement to use a legitimate density specification.

IDMLIB.IO.LTAPE.FILENO

Accessing host tape file number <fileno>.

Explanation: Host tape file number <fileno> (numbered from one) will be accessed.

User Action: None required. This message is informational only.

IDMLIB.IO.LTAPE.FILESET

Wrong fileset: wanted "<requested>", actual "<actual>".

Explanation: The open requested fileset "<requested>". Although the volume verified, the fileset read from the tape was "<actual>", which did not match. The open was rejected for this reason.

User Action: Change the fileset parameter in the open, or find the correct volume.

IDMLIB.IO.LTAPE.NEWFILE

Illegal mode for creating "newfile" control on file "<filename>".

Explanation: The program attempted to create a "newfile" on labeled tape with an illegal mode.

User Action: Use one of the legal modes for labeled tape. The "newfile" control can only be used when writing or appending to tape.

IDMLIB.IO.LTAPE.NOBACKSPACE

Cannot perform requested operation on labeled tape.

Explanation: This system cannot open the requested file on labeled tape because the operating system does not support the backspace operation.

User Action: The only operations possible are "read" and "overwrite". Please choose one of these operations. It is not possible to write multiple files on the tape.

IDMLIB.IO.LTAPE.NOFILE

Cannot open file "<tapefile>" on labeled tape.

Explanation: "<tapefile>" was not found on this tape.

User Action: Check the spelling of the filename, or verify that the file number is correct.

IDMLIB.IO.LTAPE.NOOPERATOR

No operator is available to change tapes.

Explanation: You are trying to use host tape, but there is no operator available to process tape change requests. If your job requires operator attention it will be aborted.

User Action: If you do not believe that your job will require operator attention, none required. If your job will require operator attention, you may wish to stop the job now and arrange for it to be run when an operator is available. On some systems (e.g., UNIX), this warning can occur if you have redirected standard input of the command.

IDMLIB.IO.LTAPE.NOTEXPIRED

Tape file is not expired.

Explanation: You have tried to overwrite a tape file that is marked as "saved" until a designated expiration date. You cannot write this file until it has expired.

User Action: Select another file name, or a different generation number for the same file.

IDMLIB.IO.LTAPE.PERM

Access denied; tape accessibility = "<access>".

Explanation: Access to the tape volume or file was denied because the file specified accessibility key "<access>" which is not compatible with your access.

User Action: Consult the creator of the tape for access block size.

IDMLIB.IO.LTAPE.SMALLBLOCK

Block size <blocksize> smaller than system minimum (<minblocksize>).

Explanation: Due to hardware limitations, it is not possible to have a tape blocksize smaller than <minblocksize> bytes. You have specified a size of <blocksize> bytes.

User Action: Use a larger blocksize.

IDMLIB.IO.LTAPE.UNAVAILABLE

Labeled tape is unavailable.

Explanation: Your system does not support labeled tape. This is usually caused by to lack of operating system support. For example, only some versions of UNIX provide the controls necessary to manipulate the tape drive.

User Action: Use another I/O medium.

IDMLIB.IO.MSG.CANTWRITE

Cannot write to message file <filename>.

Explanation: The program attempted a write operation to an IftMText type file. This is an illegal operation for this file type.

User Action: This is usually a programmer-error. The message file is a read-only file. Try writing to the message source files using buildmsgs to create a new message file.

IDMLIB.IO.NEEDNAME

Name required for <filetype>.

Explanation: File type <filetype> requires a name. For example, host files cannot be opened without an associated file name.

User Action: Fix the program to supply a name.

IDMLIB.IO.NOBASE

Base IFP required on <filetype> <filename>.

Explanation: Files of type "<filetype>" require an underlying file to use for the actual I/O. An ifopen call was executed that did not have the necessary base file.

User Action: The ifopen call should be corrected.

IDMLIB.IO.NOMODE

Mode required to open <filetype> <filename>.

Explanation: A <filetype> must be opened with a mode — "r" for read, "w" for write, "a" for append, or "u" for update. Not all modes are defined for each filetype.

User Action: This is usually a programmer-error. Change the "ifopen" call.

IDMLIB.IO.REOPEN.CANTOPEN

Can not reopen on device <filename>: <errmsg>.

Explanation: The program was unable to reopen on device <filename>. The operating system gave the following reason: <errmsg>.

User Action: Check the maximum number of reopen connections allowed.

IDMLIB.IO.REWRITE

Cannot rewrite <filetype> <filename>: <reason>.

Explanation: It was not possible to rewrite file "<filename>" of type "<filetype>" in response to an ifcontrol call because: <reason>.

User Action: This is usually caused by a program error or an installation problem. Contact your program support group.

IDMLIB.IO.ROWOF

Read on write-only file <filetype> "<filename>".

Explanation: An attempt was made to read from a file that was not enabled for reading.

User Action: Usually a programmer error. Open the file with the correct mode or do not attempt to read it.

IDMLIB.IO.RSIZE

size mismatch on <filetype> "<filename>", user <usersize>, file <filesize>

Explanation: The user has specified that the "rs" parameter is <usersize> for <filetype> "<filename>", but a status inquiry on the file indicates that the actual value for "rs" is <filesize>.

System Action: The user parameter will be used instead of the file parameter, but further errors may result.

User Action: Change the parameter to match the file parameter, or omit the specification of the parameter altogether.

IDMLIB.IO.SCAN.CANTWRITE

Cannot write to scan file <filename>.

Explanation: The program attempted a write operation to an IftScan type file. This is an illegal operation for this file type.

User Action: This is usually a programmer-error. Scanning is a read-only operation. Try writing to the underlying file.

IDMLIB.IO.SCAN.EOFINCOMMENT

End of file in comment from <filetype> "<filename>".

Explanation: An end of file was found in a comment while the <filetype> "<filename>" was being read. This is normally caused by a missing "close comment" indication. For example:

```
/* this comment is unterminated
```

User Action: Fix the comment.

IDMLIB.IO.SCAN.EOFINSTRING

End of file in string from <filetype> "<filename>".

Explanation: An end of file was found in a string when reading the <filetype> "<filename>". This is normally caused by a truncated file.

User Action: Fix the file.

IDMLIB.IO.SCAN.NLINSTRING

Unterminated string from <filetype> "<filename>".

Explanation: A string was found when reading <filetype> "<filename>" that is apparently unterminated. This is normally caused by a missing closing quotation mark. For example:

```
"this quote is unterminated
```

User Action: Fix the string. If you need to insert an end of line in the string, use the backslash character at the end of the line. For example:

```
"this string comprises \
    two lines"
```

IDMLIB.IO.SCAN.NOROOM

No room for token from <infiletype> "<infilename>".

Explanation: The scanner cannot fit a complete token into the space provided. Tokens must be read in their entirety.

User Action: Check the underlying file for an exceedingly long token, such as a string without its terminating quotation mark.

IDMLIB.IO.SHORTREAD

Discarding <lost bytes> bytes of data from <file type> "<file name>": short read.

Explanation: Discarding <lost bytes> bytes of unread data in <file type> "<file name>" caused by a read specifying fewer bytes than the record size.

User Action: Increase the record size by at least <lost bytes> bytes.

IDMLIB.IO.STRING.OVERFLOW

String overflow.

Explanation: An attempt was made to write to a file of type IftString. The write would have caused the buffer to overflow.

User Action: If this is a program error, allocate a larger buffer for the file.

IDMLIB.IO.TCP.BADHID

Attempt to change HID.

Explanation: The host id is a non-default value and the IDENTIFY packet attempted to change the host id.

User Action: Verify that the host ID in the system params has not been modified.

IDMLIB.IO.TCP.BADHUID

Bad HUID received by database server.

Explanation: There were not four bytes of data accompanying the HUID packet. User Action: Contact your Britton Lee IDM distribution ethernet support person.

IDMLIB.IO.TCP.BADIDENT

Badly formatted IDENTIFY received by the database server.

Explanation: Either there was not enough data in the IDENTIFY packet or the data did not correspond to the host type.

User Action: Contact your Britton Lee IDM distribution ethernet support person.

IDMLIB.IO.TCP.BADLNKCHILD

Invalid LNKCHILD command sent to database server.

Explanation: Either (a) the child id in the data is invalid, or, (b) an open database referred to by the child has not had a MAKECHILD command issued on it, or, (c) the sending parent has no open database, or, (d) the sending parent is itself a child.

User Action: Verify that the application is not violating any of the above conditions.

IDMLIB.IO.TCP.BADMAKECHILD

Invalid MAKECHILD command sent to the database server.

Explanation: The socket sending the MAKECHILD packet either has an open database or is already a child.

User Action: Verify that the socket sending the MAKECHILD qualifies as a child.

IDMLIB.IO.TCP.HUID

Wrong datatype returned in ACK to HUID.

Explanation: A datatype not equal to HUID was returned in response to the HUID command.

User Action: Contact your Britton Lee IDM distribution ethernet support person.

IDMLIB.IO.TCP.SENDIDENT

Data sent before IDENTIFY.

Explanation: The first data packet sent to the database server on this socket was not an IDENTIFY packet.

User Action: Verify the application opens a socket before sending data.

IDMLIB.IO.TCP.SETOWN

Fcntl for F_SETOWN failed on socket <socket>.

Explanation: The process failed to set the process id which is to receive notification of pending out-of-band input to its own process id. This is an internal error.

User Action: Report the problem to your Britton Lee IDM distribution ethernet support person.

IDMLIB.IO.TCP.READERR

Read call to driver returned zero bytes.

Explanation: A read request to the database server returned zero bytes which was not expected.

User Action: Try the command again and contact your Britton Lee IDM distribution ethernet support person.

IDMLIB.IO.TCP.UNKNOWNERR

Unknown error message received from the database server.

Explanation: The database server channel sent an unknown error code.

User Action: Contact your Britton Lee IDM distribution ethernet support person.

IDMLIB.IO.TERM.UNKNOWN

Terminal type "<termtype>" unknown.

Explanation: The terminal type "<termtype>" is not a known type. Special terminal controls cannot be performed on an unknown type.

User Action: Consult your local system administrator for the names of the accepted terminal types.

IDMLIB.IO.UOWOF
"Ungetc" on write-only file <filetype> "<filename>".
Explanation: An attempt was made to unget a byte on a file that was not enabled for reading.
User Action: Usually a programmer error. Open the file with the correct mode or do not attempt to unget bytes to it.

IDMLIB.IO.WLR
Wrong length record on <filetype> "<filename>", record <recno>: recsize <reclen>, asked <reqlen>.
Explanation: When accessing record <recno> of <filetype> "<filename>", the record size requested (<reqlen> bytes) exceeded the size available (<reclen> bytes).
System Action: The record has been truncated to <reclen> bytes.
User Action: Specify a larger maximum record length when opening the file, or avoid using large records.

IDMLIB.IO.WOROF
Write on read-only file <filetype> "<filename>".
Explanation: An attempt was made to write on a file that was not enabled for writing.
User Action: Usually a programmer error. Open the file with the correct mode or do not attempt to write it.

IDMLIB.IO.XNS.BAD_DSTYPE
Illegal data stream type.
Explanation: The data stream type received, "<bad dstype>", was not valid.
User Action: Contact your Britton Lee IDM distribution ethernet support person.

IDMLIB.IO.XNS.BADASYNC
Failure to establish catcher for ATTENTION packets.
Explanation: An attempt to establish a catcher for asynchronous network events failed. This is probably an operating systems failure.
User Action: Consult your Britton Lee IDM distribution ethernet support person.

IDMLIB.IO.XNS.BADCANH
Cancel Host Failed.
Explanation: Cancel host request failed. Most likely, you attempted to cancel all connections and you didn't use the lifeline (first created) socket.
User Action: Use the lifeline socket (if you are authorized) and try again. Possibly the identify daemon was created AFTER some other socket was created. The FIRST SOCKET opened is always considered the lifeline socket.

IDMLIB.IO.XNS.BADCHILD
Database server failed to establish a child on a reopen.
Explanation: The database server complained about a reopen request.
User Action: Consult your Britton Lee distribution support person.

IDMLIB.IO.XNS.BADHALFC
A HALFCANCEL attempt failed on the ethernet.
Explanation: The database server sent data along with a half-cancel message.
User Action: Contact your Britton Lee distribution support person.

IDMLIB.IO.XNS.BADIDY
Identify failed.
Explanation: Attempt to send identify packet failed. The operating system reported <error> as the reason.
User Action: The ethernet may be having physical problems. Check that the ethernet is operating properly, and ask your Britton Lee distribution support person for help.

IDMLIB.IO.XNS.BADIDYACK

Illegal attempt to issue Identify command.

Explanation: You attempted to issue an IDENTIFY command and you didn't use the lifeline socket.

User Action: Use the lifeline socket (if you are authorized) and try again. Possibly the identify daemon was created AFTER some other socket was created. The FIRST SOCKET opened is always considered the lifeline socket.

IDMLIB.IO.XNS.BADREAD

XNS read request failed.

Explanation: A read request to the database server on the XNS channel failed.

User Action: Determine if the socket you used was legal and still active. Also check the connectivity of the ethernet and see that the database server is still connected and powered on.

IDMLIB.IO.XNS.BADRD

XNS read request failed.

Explanation: A read request to the database server on the XNS channel failed.

User Action: Check the meaning of <systemmessage> in your system manual. Check that the socket you used was legal and still active. Try using the identify daemon (if used) to resynchronize with the database server. Check that XPKTSIZE agrees with the value in the "configure" relation. Check (with a refectometer if available) the connectivity of the ethernet. Check that database server is connected to the ethernet. Inspect the ethernet taps. Check the host ethernet board by sending to other sites (including itself).

IDMLIB.IO.XNS.BADTYPE

Non-existent type field received from ethernet channel.

Explanation: A type field in the XNS header which is not used by the database server ethernet channel was received from the database server.

User Action: Check that the ethernet connection is working reliably. Try enabling checksums on your XNS driver if possible. Contact your Britton Lee distribution support person.

IDMLIB.IO.XNS.BADWRT

XNS write request failed.

Explanation: A write request to the database server on the XNS channel failed.

User Action: Check the meaning of <systemmessage> in your system manual. See that XPKTSIZE agrees with the value in the "configure" relation. See that the socket you used was legal and still active. Try using the identify daemon (if used) to resynchronize with the database server. Check (with a refectometer if available) the connectivity of the ethernet. Check that database server is connected to the ethernet. Inspect the ethernet taps. Check the host ethernet board by sending to other sites (including itself).

IDMLIB.IO.XNS.BIGPKT

Packet size too big.

Explanation: Attempt to use unsupported packet size.

User Action: Set the packet size in the configuration relation to a smaller size, set XPKTSIZE to agree with this, and try again.

IDMLIB.IO.XNS.ERRDST

Database server protocol error.

Explanation: An error data stream type was received by the database server library. This indicates there is an error in the HOST <-> database server protocol.

User Action: Contact your Britton Lee IDM distribution ethernet support person.

IDMLIB.IO.XNS.ERRORPKT

The database server reported that a command was in error.

Explanation: Data in data field of a packet did not correspond to the type field.

User Action: Contact your Britton Lee distribution support person.

IDMLIB.IO.XNS.LIFEONLY

Only Lifeline socket is active.

Explanation: Attempt to signal identify daemon is useless.

User Action: None. All client connections are closed and the lifeline socket is working ok. The signal is used to wipe out existing client sockets with cancel host.

IDMLIB.IO.XNS.NOIDENTIFY

No identify command has been sent to the database server device "<device>".

Explanation: You can not run this program unless an identify command has been sent to the database server.

User Action: Make sure that the xns daemon is currently running for the database server device "<device>", that the database server is powered up and that your network is fully connected.

IDMLIB.IO.XNS.NO_MSG_QUEUE

No message queues available.

Explanation: The IDMLIB open routine could not get a system message queue to send data to the XNS server process. The maximum number of message queues has probably been exceeded.

User Action: Have your system administrator verify that all message queues are being used. The system administrator might consider increasing the number of message queues allowed by the system.

IDMLIB.IO.XNS.NO_SERVER

Your process is unable to access the XNS Server.

Explanation: The attempt to access the XNS server failed because the XNS server isn't processing user requests. This can be caused by three conditions. 1) The XNS server has not been started. 2) The XNS server has not been enabled to process user requests. 3) The XNS server could have terminated itself because of an internal error condition.

User Action: Have your system administrator verify that the XNS server has been properly initialized.

IDMLIB.IO.XNS.OPEN_ERROR

<errmsg>: cannot establish a database server session.

Explanation: The database server library received a failure message when attempting to establish a session with the database server.

User Action: Correct the situation "<errmsg>" if it makes sense. Check that the host address has a matching address in the "configure" relation on the database server. It is also possible that the database server is off-line. Check with your system administrator if required.

IDMLIB.IO.XNS.SERVER_ERROR

The XNS Server reports an error on the last message sent.

Explanation: The IDMLIB/XNS Server interface received an error from the XNS server for the last message sent.

User Action: Correct the situation "<errmsg>" if it makes sense. Consult your local system adminstrator.

3. IDMRUN Subsystem

These messages are generated by the IDMRUN (application level interface) subsystem of IDMLIB.

IDMRUN.BADIDMRUN

Null or non-IDMRUN structure passed to "<function>".

Explanation: This program attemped to pass a non-valid structure pointer to "<function>".

User Action: Check that the arguments are correct to "<function>".

IDMRUN.BINDTYPE

Illegal conversion from <fromtype> to <totype>.

Explanation: The program called "irbind" in an attempt to convert data of type <fromtype> to type <totype>. This conversion is not supported.

User Action: Determine the correct types of all data, and determine what conversions are legal for these types. Then try again. In some cases, you may need to convert data to a type that can be legally converted.

IDMRUN.DONECMDS

<func>: done with all commands

Explanation: The program has tried to continue processing commands after all results from all commands have been processed.

User Action: Create a new command using "iridl" or "irsql" or reexecute the current command using "irexec".

IDMRUN.GETFLD

Illegal get field request: <fieldnum>.

Explanation: A call to "irget" attempted to get a nonexistent option.

User Action: Probably a program error.

IDMRUN.GETTREE

Command tree <cmdnum> is nonexistent.

Explanation: In a call to irget, an attempt was made to refer to tree number <cmdnum>. This tree was not found.

User Action: Usually a programmer error. Contact the responsible programmer.

IDMRUN.MOREDATA

<function>: More data is pending.

Explanation: There is more data to be read from the database server before you can call <function>.

User Action: The program must call irfetch, irflush, or ircancel to read or discard the data.

IDMRUN.NEWTL

Received new AUDIT target-list.

Explanation: An "audit" command has been run. Unlike most other commands, "audit" can return more than one target-list. A program that reads data must make special arrangements to recieve all of the data from the "audit" command. If you recieve this message, the program has not made these arrangements, and data is being discarded.

User Action: Check your results carefully for accuracy. The results of any "audit" command are suspect. In particular, look for streams of repeated data.

IDMRUN.NOCMDS

<function>: No executable commands.

Explanation: The program has tried to call "<function>" using an IDMRUN structure currently containing no executable commands.

User Action: Normally results from not testing for failure from "iridl" or "irsql". Check and correct the program.

IDMRUN.NOTEXEC

<arg1>: Commands not executed.

Explanation: The program has tried to call "<arg1>" while commands were remaining to be processed in the IDMRUN structure.

User Action: Call "irnext" to execute the remaining commands.

IDMRUN.RECOMPILE

Called "<function>" with bad IDMRUN version. This program must be recompiled.

Explanation: This program is trying to access a different version of IDMRUN than is currently installed on this system. The program must be recompiled from the source before it can be run. Detected by "<function>".

User Action: Recompile and relink the program.

IDMRUN.SETFLD

Illegal set field (<field>) request.

Explanation: A call to "irset" attempted to set field "<field>", which does not exist.

User Action: Probably a programmer error. Contact the responsible programmer.

IDMRUN.TARGNUM

Bad target-list element number in <func>: <targnum>.

Explanation: An illegal target-list element number (<targnum>) was passed to the run-time routine <func>.

User Action: Correct the program, or inform the programmer responsible.

IDMRUN.USEIREXEC

<func>: command not executed with "irexec".

Explanation: The program called <func> before calling "irexec".

User Action: Use the routine "irexec" to execute the first command followed by the routine "irnext" to execute the remaining commands.

IDMRUN.USENEXTCMD

<func>: more commands pending.

Explanation: The program has tried to begin a new command without processing all of the previous commands.

User Action: Use the routine "irnext" to process the results of the remaining commands.

3.1. IDMLIB System Dependent Messages

These messages are specific to UNIX.

IDMLIB.SYSEDIT

Edit of "<filename>" failed: <reason>

Explanation: It was not possible to run the editor on "<filename>" because <reason>.

User Action: If the reason is not self explanatory, ask your system support staff for help.

IDMLIB.SYSSHELL

Shell failed: <reason>

Explanation: It was not possible to invoke a shell because <reason>.

User Action: If the reason is not self explanatory, ask your system support staff for help.

4. Interactive Query Processors

These messages may be generated by the Interactive Query Processor IDL.

IQP.BADCONTCHAR

Illegal continuation character

Explanation: You have tried to specify an illegal continuation character. Continuation characters must be single operator characters (i.e., not letters or digits) only.

User Action: Use a single character as the continuation character. Make sure it is a special character, e.g., "-" or "/".

IQP.BUILTIN.ARGCNT

Wrong number of arguments to <builtin> command; <mincount>-<maxcount> allowed.

Explanation: You have given the "<builtin>" command too many arguments. You must specify at least <mincount> argument(s), and at most <maxcount> argument(s).

User Action: Correct the arguments to the <builtin> command.

IQP.BUILTIN.ONOROFF

Illegal argument "<argument>" — "on" or "off" required

Explanation: The "<builtin>" command requires an argument that is either "on" or "off". You have given it "<argument>" instead.

User Action: Use "on" or "off" instead of "<argument>" in the "<builtin>" command.

IQP.CTRLCHAR

Illegal character "<character>" deleted.

Explanation: The non-printable control character "<character>" was found while reading your commands. It was deleted to prevent confusion.

User Action: Normally none required.

IQP.IGNOREINPUT

Input before the "exit" command has been ignored.

Explanation: The IDL statements input were not processed before the "exit" command. These commands will be ignored.

User Action: Add a "go" to the end of the last statement(s) before the "exit" command if you want them to be processed. Use "reset" if you want them to be ignored.

IQP.UNKNOWN

Unknown command "<command>".

Explanation: The command "<command>" is not known to the IDL program.

User Action: Use a command that is known. Known commands are:

%associate [on | off]
%continuation [<char>]
%display <text>
%edit [<filename>]
%experience <level>
%help
%input [<filename>]
%redo
%showranges
%substitute <name> <value>
%trace <tracepec>

Consult the IDL Reference Manual for details.

IQP.USAGE

Explanation: The "idl" command was used incorrectly.

User Action: Consult the manual for details and try again.

IQP.XCANCEL

Exiting inside a transaction — transaction canceled.

Explanation: Exiting idl from within a transaction causes all operations performed since the beginning of the transaction to be backed out (undone).

User Action: End the transaction before exiting the parser

5. DBA Utilities

These messages are generated by the DBA utility programs.

5.1. Dumptape

DUMPTAPE.CANTOPEN

Cannot open tape drive "<tapename>".

Explanation: The tape drive "<tapename>" cannot be opened.

User Action: Make sure that "<tapename>" is a valid tape device on your system and that the tape drive is on line.

5.2. Idmcklog

IDMCKLOAD.DONE.HOST

Load verification completed, with a total of <count>.

Explanation: The "idmckload" command was completed as requested. A total of <count> were loaded.

User Action: none required.

IDMCKLOAD.DONE.IFILE

Load verification completed from database server file.

Explanation: The "idmckload" command was completed as requested.

User Action: none required.

IDMCKLOAD.DONE.ITAPE

Load verification completed from database server tape.

Explanation: The "idmckload" command from database server tape was completed as requested.

User Action: none required.

IDMCKLOAD.VERIFYING

Verifying the <log to db> from <wherefrom>.

Explanation: An "idmckload" command is progressing, verifying a database from <wherefrom>.

User Action: none required.

IDMCKLOAD.PAGE

Sent <blocks> blocks.

Explanation: So far, <blocks> blocks have been copied from the dump image to the database server.

User Action: none required.

5.3. Idmcklog

IDMCKLOG.COMPLETE

Transaction log(s) have verified to contain all data pages.

Explanation: If the time stamps are correct (sequential), then these logs will load into the database server properly.

User Action: none required.

IDMCKLOG.INCOMPLETE

Log "<transfile>" is NOT a complete transaction log.

Explanation: The last page of the log is not there. The log will not load into the database without possibly hanging the database server communications channel.

User Action: Dump another log or find a log with the same start timestamp as this one.

IDMCKLOG.IOERR

Read of "<transfile>" failed

Explanation: The read of the file "<transfile>" returned negative.

User Action: Check the permissions on the file.

IDMCKLOG.NOARG

A transaction log specification must be given.

Explanation: A transaction log specification was not supplied to idmcklog. This argument is required.

User Action: List a log.

IDMCKLOG.NOTALOG

File "<filename>" is not a transaction log.

Explanation: A heuristic test shows that the first three bytes are not set to -1 as in a normal transaction log.

User Action: This is not the right type of object. Check your records to see if you created "<filename>" with idmdump.

IDMCKLOG.IDM.NOCHECK

Cannot currently check transaction logs on the database server ("<transfile>").

Explanation: Database server files and database server tape cannot be read from the host.

User Action: Dump the database server file to a host file and then check the log.

5.4. Idmconfig

IDMCONFIG.FULL

There are no more slots (max is <maximum>) for tuple type "<type>".

Explanation: Only <maximum> slots are available for configuration. This message is only possible when new configure information for the "E" (ethernet) or "B" (block mux) channels is being appended.

User Action: Try to reuse or remove unused entries.

IDMCONFIG.HUNAME.SET

A value (<username>) has been set for the IDMHUNAME parameter.

Explanation: This is not advised, as all users will appear as user name <username>.

User Action: Check that an environment variable has been set or that the system params file was incorrectly modified to set the IDMHUNAME parameter.

IDMCONFIG.ILLEGAL.NUMBER

A configure.number value greater than 31 was found for type "<type>".

Explanation: A number entry in the "configure" relation of the system database for type <type> is greater than the legal limit of 31 and should be removed interactively using IDL or SQL.

User Action: Remove tuple where type = "<type>" and number > 31.

IDMCONFIG.PASSWD.SET

A value (<password>) has been set for IDMPASSWD in the system params file.

Explanation: This is a very dangerous practice.

User Action: Remove the value from the system params file

IDMCONFIG.SELECT.NOSUCHITEM

There is no such item numbered "<itemnumber>". Select a valid item.

Explanation: The item selected was not found to exist in the "configure" relation.

User Action: Check the values listed and select a valid "Item" number

IDMCONFIG.TOOMANY.AFFECTED

Too many (<tuples affected>) tuples were affected.

Explanation: This should not happen. Contact your Britton Lee distribution support person

User Action: Check that the proper values are present in the "configure" relation for the configuration type that was just modified.

5.5. Idmcopy

IDMCOPY.IFILE
Cannot copy using database server files.
Explanation: The database server does not support copying to or from database server files.
User Action: Change the file specification type to host tape ("htape"), host file ("hfile") or database server tape ("itape").

IDMCOPY.IN.DONE
Copied from <filespec> into database <dbname>.
Explanation: Copied successfully.
User Action: None required.

IDMCOPY.IN.DB
Copied entire database from <filespec> into database <dbname>.
Explanation: Copied successfully.
User Action: None required.

IDMCOPY.NORESULTS
There were no results sent by the database server.
Explanation: The database server did not send any results when a copy command was sent.
User Action: Contact your Britton Lee distribution support person for assistance.

IDMCOPY.NOSOURCE
Missing host source file name.
Explanation: When copying in from host file or host tape, a name must be specified.
User Action: Use the command line option to set the name.

IDMCOPY.OUT.DONE
Copied "<relname>" from database <dbname> to <filespec>.
Explanation: Copied successfully.
User Action: None required.

IDMCOPY.OUT.DB
Copied entire database <dbname> to <filespec>.
Explanation: Copied successfully.
User Action: None required.

IDMCOPY.OUT.NORELATION
No user relations in the database to copy out.
Explanation: The database contains only system relations which are not copied out. No relations were copied out.
User Action: Create relations before copying them out.

IDMCOPY.PAGE
Dumped <blocks> blocks.
Explanation: This message is issued periodically through the dump to inform you of the situation.
User Action: None required.

IDMCOPY.TAPE.IN.DONE
Copied into database "<dbname>" from database server tape.
Explanation: Copied successfully.
User Action: None required.

IDMCOPY.TAPE.IN.DB
Copied entire database "<dbname>" from database server tape.
Explanation: Copied successfully.
User Action: None required.

IDMCOPY.TAPE.OUT.DONE
Copied from database "<dbname>" to database server tape.
Explanation: Copied successfully.
User Action: None required.

IDMCOPY.TAPE.OUT.DB
Copied entire database "<dbname>" to database server tape.
Explanation: Copied successfully.
User Action: None required.

5.6. Idmdate

IDMDATE.DATE

Set database server date to <date>.
Explanation: Date set successfully.
User Action: None required.

IDMDATE.DATETIME

Set time and date on the database server to <date-time>.
Explanation: Date and time set successfully.
User Action: None required.

IDMDATE.TIME

Set database server time to <time>.
Explanation: Time set successfully.
User Action: None required.

5.7. Idmdba

IDMDBA.BADCOMMAND

"<command>" is not a recognized command.
Explanation: Your command was not legal for this program.
User Action: Try a legal command.

IDMDBA.BADPWFILE

Error in parsing password file <pwname>: on line containing: <line>
Explanation: The file <pwname> is of in the correct syntax.
User Action: Have the system administrator correct the file.

IDMDBA.BADGRPFILE

Error in parsing group file <grpname> for on line containing: <line>
Explanation: The file <grpname> is of in the correct syntax.
User Action: Have the system administrator correct the file.

IDMDBA.EXISTS

User "<name>" already exists with the following values (<hostuid>).
Explanation: The user "<name>" exists and was not added.
User Action: Use "showuser <name>" to check that the user is correctly installed in the database already.

IDMDBA.NEEDHUID

No host user id given.
Explanation: A host user id could not be found and must be specified by the user.
User Action: Use the "host user id" flag to the command.

IDMDBA.NOCMD.DBS

No stored command 'dbs' found in database 'system'.
Explanation: This program will use the 'dbs' command if found. Only name and owner will be printed now, without 'dba'.
User Action: Load 'dbs' from the Stored Commands supplied by Britton Lee.

IDMDBA.NODBNAME

Must specify a database name to the "open" command.
Explanation: Pass the name of the database to open as the second argument to "open".
User Action: "open <databasename>" is the correct usage.

IDMDBA.NOTUNIQUE

Attempt to append user with a duplicate database server user id <idmuid> rejected
Explanation: Multiple users on the same database server user id are not allowed in this program. It causes several different problems: book keeping shows phantom items, the database server does not honor such a configuration, etc.
User Action: choose another database server user id, or let idmdba pick one for you. Use the group id to map users together or share one database server name.

IDMDBA.NOUSER

User <user> not found.
Explanation: The query on user <user> did not succeed since the user name was not in the "users" relation. It is possible that the user is in the "host_users" relation, but not "users".
User Action: None.

IDMDBA.MOREUSERS

<count> more 'users' with uid:<uid>: no deletion of "host_users" performed.
Explanation: It is not possible to determine which "host_users" tuple corresponds to users:id <uid> for user <user>. User Action: If you wish to remove access from '<user>', delete the appropriate "host_users" tuple interactively.

IDMDBA.PASSWD.OPEN

Checking file "<filename>" for host user and group id

Explanation: The password file <filename> is being read for information

User Action: none.

5.8. Idmdump

IDMDUMP.DUMPXACT.MODE

Illegal mode for transaction log dump: <mode value>

Explanation: Mode flags may only be used when dumping the transaction log and the database.

User Action: Either remove the flag or also dump the database.

IDMDUMP.DUMPXACT.WAIT

Ignoring request to wait for updaters on transaction log dump.

Explanation: It is possible to wait for updaters to commit only when dumping a database. You are dumping the transaction log.

User Action: Do not use this command line option or dump the database also.

IDMDUMP.HOST

Dumping the <database/transaction log> "<name>" to <whereto>.

Explanation: The <database/transaction log> "<name>" is about to be dumped to <whereto>. Additional information will be given as the dump proceeds.

User Action: Sit back and enjoy the feeling of comfort that comes from knowing that your database is being safely backed up.

IDMDUMP.HOST.DONE

Dumped <object> with a total of <blockcount>.

Explanation: The dump successfully completed. <blockcount> blocks were dumped to host file or host tape.

User Action: None Required.

IDMDUMP.IDM.DONE

Dumped <object> tothe database server.

Explanation: The dump successfully completed to database server file or tape.

User Action: None Required.

IDMDUMP.LOGNAME

Ignoring named log "<usersname>".

Explanation: The database server does not support the naming of log files while dumping the database.

User Action: Dump only the transaction log if you must rename the log file.

IDMDUMP.MIXEDDEST

Cannot mix database server tape and host file or tape destinations.

Explanation: Dumping the database to <db destination> and the log to <log destination> is not an allowed combination.

User Action: Change the destination of either the database or the transaction log so that you no longer have this invalid mixture. For example, do not use database server tape or only use database server tape (not the host file or tape).

IDMDUMP.MODE.CNVT

Bad mode value <user mode> for <field name> field.

Explanation: An error occured in converting the numeric "<field name>" field.

User Action: Check that there is a numeric value specified on the command line for the "online" mode option.

IDMDUMP.NOTUSED

Ignoring database server tape parameters: "<userspec>".

Explanation: When database server tape is used for both database and transaction dumps, all tape parameters must be specified with the "-t" flag. Those specified with the "-d" flag have been ignored.

User Action: Move these parameters from the "-d" flag to the "-t" command line flag.

IDMDUMP.PAGE

Dumped <blocks> blocks.

Explanation: This message is issued periodically through the dump to inform you of the dump's progress.

User Action: None required.

5.9. Idmfcopy

IDMFCOPY.BADBCDLEN

Bcd length <user's length> is too large.

Explanation: Lengths to the bcd format specify the number of BCD bytes, not the number of significant figures.

User Action: Use the formula: bytes = (digits + 3) / 2 to convert from the created length to the number of BCD bytes.

IDMFCOPY.BADCONS

The initialized field type <constant> cannot be converted to <destination>.

Explanation: Conversion cannot be performed from type <constant> to <destination>.

User Action: Try double quoting (") the constant to force it to a character type. Contact your Britton Lee distribution support person.

IDMFCOPY.BADFIELD

Initialized field and data field are different ("<initfield>" from data "<datafield>").

Explanation: A difference was found in comparing the initialized value with the input data. The two fields have different values.

User Action: Check the record in the reject file for possible misaligned data fields.

IDMFCOPY.BADFTYPE

The type of the file in "<filespec>" is not supported with idmfcopy.

Explanation: The only types which are supported are "htape" and "hfile".

User Action: Change the type following the '%' sign. For example, "name%ifile" must change to "name%hfile". Database server file and tape are not supported.

IDMFCOPY.BADRECTYPE

<rec type> is not a valid record type.

Explanation: It is not legal to use <rec type> in this context. Legal record types are "(*)", "to <recsep>" or "(N)" where N is a the number of bytes in the fixed length record.

User Action: Use a legal record type.

IDMFCOPY.BATCH

Starting a new batch with a total of <tupcount> tuples accepted.

Explanation: A total of <tupcount> tuples have been accepted so far by the database server. We are starting up a new batch.

User Action: none required.

IDMFCOPY.CANTCNVT

Field <field name>: Cannot convert type "<input type>" to "<output type>".

Explanation: Conversion from <input type> to <output type> is not yet supported.

User Action: See if the output type (<output type>) can be changed to a more generic data type.

IDMFCOPY.CANTSIGN

Unable to change the sign on the constant type "<constant type>".

Explanation: The initialized constant of type "<constant type>" cannot have it's sign changed.

User Action: Contact your Britton Lee distribution support person.

IDMFCOPY.CK

Checked formatting on <tuplecnt> tuple(s) with <errorcount> error(s).

Explanation: No tuples have actually been copied in. <tuplecnt> tuple(s) were checked for validity; <errorcount> of those tuple(s) had errors.

User Action: Use the reject file to collect the bad tuples. The tuples can then be checked and corrected.

IDMFCOPY.CKVALUE

Illegal character '<badchar>' in data "<value>" for field "<fieldname>", type <user-type>.

Explanation: While the data for field <fieldname> was being checked, an illegal character ('<badchar>') was encountered. This character is illegal for the type <user-type>.

User Action: Remove '<badchar>' from the data <value>.

IDMFCOPY.CNVT.OVERFLOW

Conversion overflow on "<string>".

Explanation: An attempt to convert "<string>" failed because of an overflow that would probably produce an incorrect result. The maximum value is <maxvalue>. Integers will be left with a strange value. Conversion to unsigned hex and octal numbers is not supported.

User Action: Check your results carefully to see which of them are incorrect.

IDMFCOPY.DELCONST

The delimiter <delimvalue> is less than 0 and greater than 255.

Explanation: The specified integer delimiter evaluated to an illegal range. Delimiters must be between 0 and 255.

User Action: Check the constant for typos.

IDMFCOPY.ERROR

Aborting the formatted copy after <errormax> errors.

Explanation: <errormax> errors occurred in the process of copying.

User Action: To change the error limit, use the command line option.

IDMFCOPY.FOVERF

Field "<fieldname>" overflowed by <howmuch> byte(s).

IDMFCOPY.ILLEGDECIMAL

Field <fieldname>, type <IDM type> cannot be converted to or from type <user type>.

Explanation: Only database server integer types (int1, int2, int4) are legal with type <user type>.

User Action: Use the fcopy type "text".

IDMFCOPY.ILLEGFLOAT

Field <fieldname>, type <IDM type> cannot be converted to or from type <user type>.

Explanation: Only database server float and bcd types (flt4, flt8, bcd, bcdflt) are legal with type <user type>.

User Action: Use the fcopy type "text".

IDMFCOPY.IN.DONE

Copied in <tuplecnt> tuple(s) with <errorcount> error(s).

Explanation: A total of <errorcount> tuples were copied from your source file to the database server.

User Action: none required

IDMFCOPY.IN.EMPTY

Empty data file to copy in.

Explanation: The specified data file is empty.

User Action: Check for typos in the data file specification, or put some data in your file.

IDMFCOPY.IN.EOF

Unexpected end of data found on copy in.

Explanation: The last record was too short, or idmfcopy has gotten out of sync.

User Action: If using the reject file, check to see where the error started. Check that each record in the file matches the format description.

IDMFCOPY.IN.FAIL

There was an error with this copy in. The data was not accepted.

Explanation: If copying in in batches, this batch was not committed, otherwise, the whole copy in failed.

User Action: Correct the reported errors that may have caused the the database server to reject this copy in.

IDMFCOPY.IN.OVERFLOW

Buffer overflow.

Explanation: This is an internal error.

User Action: Please contact your Britton Lee distribution support person.

IDMFCOPY.IN.RECTOOBIG

The end of the record (size <arg2>) was found searching for record terminator.

Explanation: The record buffer overflowed while reading in a record searching for a record terminator. The size of the record is <arg2>.

User Action: Try specifying a larger record size: "fname%ftype,rs(N)" where N is larger than <recordmax>. See ifscrack(3I) and introduction(1I) for a more detailed description of file specifications and record sizes.

IDMFCOPY.LONGTOK

Token is too long. Discarding extra characters.

Explanation: The token contained too many characters.

User Action: Shorten the name or constant.

IDMFCOPY.MAXDEL

Too many delimiters (<maximum>) have been specified.

Explanation: You have specified more than the allowed number of delimiters. <maximum> is more than is allowed.

User Action: Contact your Britton Lee distribution support person if it is required that you specify this many delimiters. If possible, change the delimiters in the data file to use a smaller set of delimiters.

IDMFCOPY.NOFORMAT

No format description was given by the user.

Explanation: The format of the data file must be specified for idmfcopy using a "-f" flag to name a file containing the format specification, or including a specification on the command line (be sure to quote the specification).

User Action: Include a "-f" flag or an on-line specification. If you want to do bulk data copy for backup purposes, use idmcopy instead. See also idmfcopy(1I).

IDMFCOPY.NOREL

Relation "<relname>" was not found in database "<dbname>".

Explanation: The relation "<relname>" is not in the database "<dbname>".

User Action: Check that this is an existing relation in "<dbname>".

IDMFCOPY.NOMOREDATA

Ran out of data before field "<fieldname>"

Explanation: Attempted to read past end of record in converting field "<fieldname>".

User Action: Try using a larger record size.

IDMFCOPY.NORECTERM

The specified record terminator '<recterminator>' was not found.

Explanation: '<recterminator>' was not found to terminate the record.

User Action: Check that this is the desired terminator. Check the data for the correct terminator.

IDMFCOPY.NORELNAME

No relation name was specified.

Explanation: The relation name was not given on either the command line or in the format specification.

User Action: To set the relation name, use the positional relation name on the command line, or specify using the "relation" command in the format specification file. Be sure to also give the database name if using the command line, as the relation name position is dependent on the this name.

IDMFCOPY.NOTARGET

"<field name>" is not a recognized field name for relation "<relation>".

Explanation: The relation "<relation>" does not contain a field named "<field name>".

User Action: Check the names of the fields in "<relation>".

IDMFCOPY.NOTDELIM

"<delimiter>" is not a recognized delimiter.

Explanation: "<delimiter>" is not one of the named delimiters.

User Action: Consult idmfcopy(1I) for a correct list.

IDMFCOPY.OUT.DONE

Copied out <tuplecnt> tuple(s) with <error count> error(s).

Explanation: A total of <error count> tuple(s) were copied from the database server to the destination file.

User Action: none required

IDMFCOPY.OUT.FAIL

<error count> errors or warnings with this copy out.

Explanation: The database server reported an error with this copy out.

User Action: Correct the error as reported in the error message from the database server.

IDMFCOPY.OUT.NOBATCH

Ignoring batch size <batchsize> for idmfcopy out.

Explanation: Batch copying is only used with idmfcopy in. This is a warning that idmfcopy does not batch data on copy out.

User Action: Remove the batch size flag from the command line.

IDMFCOPY.OUT.NOCHECK

Ignoring check data request for idmfcopy out.

Explanation: Conversion checking is only used with idmfcopy in. This is a warning that idmfcopy does not check data formatting on copy out.

User Action: Remove the check data flag from the command line.

IDMFCOPY.OUT.NOREJFILE

Ignoring request to write to reject file <rejectfname>.

Explanation: Reject files are only used with idmfcopy in. This is a warning that idmfcopy does not write to file <rejectfname> on idmfcopy out.

User Action: Remove the reject file flag from the command line.

IDMFCOPY.OUT.NOSKIP

Ignoring request to skip <skipnum> tuples during idmfcopy out.

Explanation: Tuple skipping is only used with idmfcopy in. This is a warning that idmfcopy does not skip data on copy out.

User Action: Remove the skipnum flag from the command line.

IDMFCOPY.RECBASED

"<filename>" is record based and not recommended with "record (*)".

Explanation: Fixed length records and field at a time processing produces varying results, most of which are wrong.

User Action: Change the record type in the format specification from "record (*)" to

"record". This will query the file for the size of the fixed length record to use.

IDMFCOPY.REJ

<record count> bad records were written to reject file "<rejectfile>"

Explanation: <record count> records had errors or warnings during copy in. Check the file "<rejectfile>" for a complete list of the bad records which were not copied in.

User Action: Determine the problem (compare with the exceptions raised during copy in) with the records. A different format specification may correct the problem. Rerun using "<rejectfile>" as the input data file and a new reject file to copy in the bad records. The "checkdata" flag to idmfcopy could be used to check (but not copy in) the records.

IDMFCOPY.ROVERF

The record overflowed by <howmuch> bytes.

Explanation: Your record did not have enough room to hold the converted data.

User Action: Try enlarging your record size using the "rs" parameter to the data file specification.

IDMFCOPY.SYNTAX

Unrecognized syntax; last token read: "<token>".

Explanation: Some of the input to the idmfcopy parser was not recognizable.

User Action: Check your input for the error. Look for an error before the input "<token>". For a complete description of the format specification language, see the documents for idmfcopy.

IDMFCOPY.USINGSTR

Command line specified format spec overriding format spec file.

Explanation: The format specification file is being ignored and the format specification on the command line is being used instead.

User Action: If this is not what you desire, remove the record description (format spec) from the end of the command line.

5.10. Idmidyd

IDMIDYD.TOOMANY

Too many database servers listed on command line, <maxhosts> maximum

Explanation: The idmidyd program can accept up to <maxhosts> database servers on the command line. This limit has been exceeded.

User Action: If you must identify this host to more than <maxhosts> database servers, you will need to run two daemons.

5.11. Idmload

IDMLOAD.DONE.HOST

Load completed, with a total of <count>.

Explanation: The "idmload" command was completed as requested. A total of <count> were loaded.

User Action: none required.

IDMLOAD.DONE.IFILE

Load completed from database server file.

Explanation: The "idmload" command was completed as requested.

User Action: none required.

IDMLOAD.DONE.ITAPE

Load completed from database server tape.

Explanation: The "idmload" command from database server tape was completed as requested.

User Action: none required.

IDMLOAD.LOADING

Loading the <log or database> "<filename>" from <wherefrom>.

Explanation: An "idmload" command is progressing, loading a <log or database> from <wherefrom>.

User Action: none required.

IDMLOAD.PAGE

Loaded <blocks> blocks.

Explanation: So far, <blocks> blocks have been copied from the dump image into the database.

User Action: none required.

IDMLOAD.SAMEDB

Working database and database to load must be different: <dbname>.

Explanation: The working database and the database that is to be loaded must be different, since the database server requires that no one be using a database while it is being loaded.

User Action: Pick a different name for the working database.

5.12. Idmpasswd

IDMPASSWD.BACKGND

Cannot run idmpasswd in background.

Explanation: The idmpasswd command must communicate with the user in order to read the old and new password. To do this the program must be run in foreground, that is, connected to the terminal.

User Action: Run idmpasswd in foreground.

IDMPASSWD.DONE

Password set.

Explanation: The new password has been set.

User Action: Use the new password instead of the old password for future database accesses.

IDMPASSWD.GETPASS

Could not read password.

Explanation: It was not possible to read one of the passwords.

User Action: Try again, making sure that the program is being run interactively.

IDMPASSWD.MISMATCH

The new password did not match.

Explanation: The new password is read twice so that a mistake or a bouncy keyboard will not lock you out of your database. The two versions of the password did not match, so the new password was not set.

User Action: Try again.

5.13. Idmread

IDMREAD.DONE.HOST

Read database server file <idmfile> in database "<dbname>" to <hostfile>; <bytes> bytes read.

Explanation: Read completed successfully.

User Action: None Required.

IDMREAD.DONE.ITAPE

Read database server file "<idmfile>" in database "<dbname>" to "<filespec>".

Explanation: Read completed successfully.

User Action: None Required.

IDMREAD.IFILE

Database server file to database server file copy is not yet supported.

Explanation: The database server does not support reads from database server file to database server file.

User Action: None Required.

IDMREAD.NEGOFFSET

Negative file offset.

Explanation: The user entered a negative file offset as one of the command-line arguments to IDMREAD. Only positive numbers are legal.

User Action: Use only a positive number. Negative numbers do not make sense in this context.

IDMREAD.NEGCOUNT

Negative byte count.

Explanation: The user entered a negative byte count as one of the command-line arguments to IDMREAD. Only positive numbers are legal.

User Action: Use only a positive number. Negative numbers do not make sense in this context.

5.14. Idmrollf

IDMROLLF.DONE

Rolled forward database "<dbname>" using the log "<logname>" from "<working dbname>" <to date>.

Explanation: Roll forward completed successfully.

User Action: None Required.

5.15. Idmwrite

IDMWRITE.DONE.HOST

Wrote <hostfile> to database server file "<idmfile>" in database "<dbname>"; <bytes> bytes written.

Explanation: Write completed successfully.

User Action: None Required.

IDMWRITE.DONE.ITAPE

Wrote database server file "<idmfile>" in database "<dbname>" from "<filespec>".

Explanation: Write completed successfully.

User Action: None Required.

IDMWRITE.IFILE

Database server file to database server file copy is not yet supported.

Explanation: The database server does not support write from database server file to database server file.

User Action: Try database server tape.

IDMWRITE.NEGOFFSET

Negative file offset.

Explanation: The user entered a negative file offset as one of the command-line arguments to IDMWRITE. Only positive numbers are legal.

User Action: Use only a positive number. Negative numbers do not make sense in this context.

IDMWRITE.NEGCOUNT

Negative byte count.

Explanation: The user entered a negative byte count as one of the command-line arguments to IDMWRITE. Only positive numbers are legal.

User Action: Use only a positive number. Negative numbers do not make sense in this context.

5.16. Idmxdbin

IDMXDBIN.DONE

Dbin <dbin> killed

Explanation: Dbin <dbin> was killed on the database server.

User Action: none.

IDMXDBIN.BADCOMMAND

The database server does not recognize the KILLDBIN command.

Explanation: The KILLDBIN command was added in RDBMS Software Release 3.5.

User Action: Check that your machine is running release 3.5 or newer RDBMS code or do not run this command.

5.17. Inittape

INITTAPE.DONE

Initialized tape volume "<volumeid>" for "<owner>".

Explanation: A tape has been initialized with a volume id of "<volumeid>". "<owner>" has been assigned as the owner.

User Action: none required.

INITTAPE.OPEN

Cannot open tape drive "<tapefile>".

Explanation: It was not possible to open "<tapefile>" to initialize the tape as requested.

User Action: Verify that the tape is correctly mounted and that the drive is on-line. At some installations you may require special permission to access the tape drive; check with your system administrator if your are not sure.

INITTAPE.WEOF.ERR

Write EOF error on labelled tape: "<reason>"

Explanation: A write error was reported by the system when initializing the tape. The reason given by the system was "<reason>".

User Action: Check that the write ring is in, the proper density has been selected and that the tape drive is online. If the problem persists, contact your System Administrator.

INITTAPE.WRITE.ERR

Write error on labelled tape: "<reason>"

Explanation: A write error was reported by the system when initializing the tape. The reason given by the system was "<reason>".

User Action: Check that the write ring is in, the proper density has been selected and that the tape drive is online. If the problem persists, contact your System Administrator.

6. Help Facility

HELP.ATTOP

At top of help tree.

Explanation: You are already at the top level menu, but you have asked to go even higher! There is nothing more general than what you are reading right now.

User Action: Pick one of the subtopics for perusal.

HELP.NONEXT

No next node defined.

Explanation: You have typed carriage return to get the default next subject, but there is no next subject defined for this particular topic.

User Action: You will have to guide yourself. Pick one of the subtopics, or use "%UP" or "%TOP" to go another topic.

HELP.UNKNOWN

Topic "<helptopic>" unknown.

Explanation: You have asked the help facility to tell you about something it knows nothing about.

System Action: The system will try successively less specific topics until it finds something it can tell you about.

User Action: Pick one of the subtopics for perusal.

7. RC Precompiler

These messages are generated by the Relational C (ric/rsc) Precompilers.

RC.ANYCAST.OHOH

Symbol of database server type "<idmsymbol>" cannot be cast into a C type.

Explanation: A symbol of database server type "<idmsymbol>" is being assigned to a C variable, but the precompiler knows of no way to cast this type into a C type.

User Action: Do not attempt to assign this symbol to a C variable.

RC.ANYFIELD.OHOH

Symbol of database server type "<idmsymbol>" cannot be assigned to a C variable.

Explanation: A symbol of database server type "<idmsymbol>" is being assigned to a C variable, but the precompiler knows of no way to convert it into a C type.

User Action: Do not attempt to assign this symbol to a C variable.

RC.ASSOC.SUBST

C expressions cannot be used for comment strings.

Explanation: In an "associate ... with string [, string]" command, one of the strings was a C-language variable or expression. This is not allowed.

User Action: Put the actual strings directly in the statement.

RC.BAD.END.ABORT

Syntax error in "<endORabort> transaction" statement.

Explanation: A statement started with "$ <endORabort>", but did not go "$ <endORabort> transaction ;".

User Action: Make the statement read "$ <endORabort> transaction ;".

RC.BAD.FILENAME

Filename "<filename>" has a suffix other than ".<suffix>".

Explanation: All precompiler source file names either must have the suffix ".<suffix>" or not have any suffix.

User Action: Rename file or fix typo.

RC.BAD.LOOP.STACK

Bad runtime nesting of $obtain loops detected at <where>.

Explanation: Internal variables that maintain the state of $obtain loops have become confused. This can be caused by doing a GOTO or a RETURN out of one of these loops. They may only be exited prematurely via the 'break' statement.

User Action: Investigate your code for a GOTO or a RETURN within a $obtain loop.

RC.BAD.TERMINATOR

This statement must be terminated by a ";".

Explanation: Only a 'retrieve' statement without an 'into' clause may be terminated with a curly brace.

User Action: Change this statement.

RC.BAD.TLIST

All retrieve targets must be C expressions.

Explanation: Usually caused by a missing dollar sign in front of a simple variable: i.e.
$retrieve (name = r.bufname) instead of
$retrieve ($name = r.bufname)

User Action: Fix the source file.

RC.BADCHARCODE

Invalid character: '<char>'

Explanation: Precompiler could not do anything with this character.

User Action: Find the character and remove it.

RC.BADSUB.TYPE
Illegal use of C expression in this command.
Explanation: User tried to use a C expression in a place where it is not allowed.
User Action: Eliminate the C expression.

RC.BUF.OVFL
Embedded query language command too long.
Explanation: Embedded commands are limited to 4000 characters. This error usually results from leaving out a command terminator. <command buffer>
User Action: Properly terminate all commands with either a semicolon or a left curly brace.

RC.CANCEL.SYNTAX
Bad syntax in 'cancel' statement.
Explanation: The cancel statement simply goes '$ cancel ;'. More was found after the word 'cancel'.
User Action: Fix the statement.

RC.CEXPR.EOF
End of file hit in C expression.
Explanation: User probably did not balance parentheses around a C expression.
User Action: Balance the parentheses, if this is the problem.

RC.CEXPR.FIRSTCHAR
Illegal first character in C expression.
Explanation: Valid C expression is either a simple identifier or a C expression enclosed by parentheses.
User Action: Correct the syntax of the expression.

RC.CEXPR.LENGTH
C expression is too long.
Explanation: C expressions must be less than 1000 characters. This error is often caused by failing to properly parenthesize expressions.
User Action: Check expression and fix.

RC.COM.OUTOFPLACE
Query language statement outside of function definition.
Explanation: There is a RIC statement outside of a function definition. Possibly caused by incorrect balancing of curly braces.
User Action: Check to see if a command is outside a function definition. If not, check curly brace matching.

RC.COM.SEMICOLON
Query Language commands must be separated by semicolons.
Explanation: Two embedded commands ran into one another.
e.g. $range of r is relation range of q is query;
User Action: Insert a semicolon and dollar sign to separate the two commands.

RC.COMMIT.ROLLBACK
The "<commitORrollback>" statement is only used interactively; it is not available in embedded sql, which uses "end transaction" or "abort transaction".
Explanation: A statement started with "$ <commitORrollback>". Although this could be valid interactive sql, embedded sql does not use this statement. Use "end transaction" or "abort transaction" instead.
User Action: Change the statement.

RC.CTOI.NOCANDO
Type of C expression ('<arg1>') has no database server equivalent.
Explanation: The precompiler only can accept a subset of the types possible in C. Check the "RIC User's Guide" for the allowed types.

User Action: Recode the expression, perhaps with a C type cast.

RC.DB.NOEXIST

Database <dbname> can not be opened.

Explanation: Trying to store programs on a database that does not exist.

User Action: Are you using the correct database server If so, create the database, or correct the typo.

RC.DORIGHTC.BADNEST

Bad nesting of command constructs; curly braces may be mismatched.

Explanation: All nested constucts must nest within the same scope level.

User Action: Check for mismatched left and right curly braces.

RC.DYRANGE.BADTYPE

C expression "<c expression>" is not a string.

Explanation: Name of relation must be in a string.

User Action: Make the expression a string.

RC.DYRANGE.DCL

Dynamic range variable must be declared globally.

Explanation: Dynamic range variables must be declared outside function definitions. The name of the range variable is in effect "global".

User Action: Move the declaration outside the function definition.

RC.DYRANGE.DECLNAME

Range variable name "<range var name>" not same as substitution name "<subst. name>".

Explanation: The "substitution name" of a dynamic range variable must be the same as a the range variable name.

User Action: Rename the substitution name.

RC.DYRANGE.NOTDECL

"<rangevar>" is not declared as a dynamic range variable.

Explanation: All dynamic range variables must be declared before use. For example: $range of r is :r;

User Action: Declare the range variable.

RC.DYRANGE.SET

Dynamic range variables can be set only in functions.

Explanation: There is a statement of the form: $range of r is $stringvar; outside a function. This is executable and must be within a function.

User Action: Move the range statement to an appropriate place.

RC.END.XACT

Unmatched "end transaction".

Explanation: Statement must pair up with a "begin transaction" at the same level of curly bracket nesting.

User Action: Fix problems with transaction-control statements or curly bracket "{...}" nesting.

RC.F77.BAD.RETRIEVE.NESTING

Bad nesting of retrieve loops detected at label <label>.

Explanation: Statement label <label> ends a retrieve loop, but there are other retrieve loops started within this loop that have not been ended yet. The statement label for the inner loop(s) is either missing or misplaced.

User Action: Fix the retrieve loops so they nest correctly.

RC.F77.CANT.DCL.YET

Can't declare variables of type <type> to the precompiler yet.

Explanation: The precompiler is currently unable to deal directly with <type> Fortran variables. Variables of this type may neither be used in Fortran expressions

embedded in IDL statements nor declared in $-flagged Fortran declarations.
User Action: Take the '$' off the declaration and assign the variable to or from another variable of a type the precompiler can deal with. Use this second variable in embedded IDL statements.

RC.F77.KEYWORD.SPLIT
RIF sentence-defining keyword is split over more than 1 source line.
Explanation: The word following the initial '$' that defines the type of IDL or RIF statement this is must be completely contained on the same line as the initial '$' sign.
User Action: Rewrite the statement, putting the entire keyword on the first line of the statement.

RC.F77.LONG.LINE
Line contains more than <max legal line length> characters.
Explanation: An input source-code line contains more than 72 characters. Characters 73 on are ignored.
User Action: If the extra characters are significant, put them on a continuation line.

RC.F77.NO.LOOP.END.CODE
Incorrect nesting for 'retrieve <label>' loop.
Explanation: A retrieve loop ending at statement label <label> is contained within a retrieve loop that ended before statement label <label> was encountered.
User Action: Fix the nesting of retrieve loops.

RC.F77.NO.SEGMENT
Missing '$ include <segment_type)' statement in this routine.
Explanation: All routines that contain statements flagged by initial '$' must also contain a
User Action: Add a '$ include <segment_type)' (or '$ include both') statement in the appropriate place.

RC.F77.NONNUMERIC.LABEL
Nonnumeric character in label field.
Explanation: A letter or a special character was found in the label field of this statement. This is often caused by forgetting the initial tab.
User Action: Add the initial tab or fix the label.

RC.F77.NOT.F77.NOR.DBL
This sentence is neither an IDL nor a Fortran 77 declaration.
Explanation: The precompiler cannot decipher this statement.
User Action: Fix the statement.

RC.F77.SYNTAX.ERR
Bad Fortran 77 syntax found near "<lexeme>".
Explanation: Precompiler can not parse this declaration or expression.
User Action: Fix the Fortran.

RC.F77.TOK.TOO.LONG
Token that begins "<token-head>" is too long.
Explanation: This Fortran77 "word" is longer than we can handle.
User Action: Figure out how to shorten this element.

RC.F77.TOO.MANY.GENLABS
Need to generate too many (><max>) labels for this routine.
Explanation: There is a maximum number of labels that the precompiler will generate. This maximum has been exceeded.
User Action: Either shorten or simplify this routine, or use the command-line option to increase the number of labels the precompiler can generate.

RC.F77.TOO.MANY.RETLABS

Too-deep nesting of retrieve loops detected at 'retrieve <label>'.

Explanation: There is a maximum depth to which retrieve loops may be nested. This maximum is exceeded here.

User Action: Simplify the nesting of retrieve loops in this routine.

RC.F77.TOO.MANY.PARAMS

Too many parameters — '<name>' must be discarded.

Explanation: There is an upper limit on the number of parameters you can inform the precompiler of. You have exceeded this limit.

User Action: Define less parameters in statements flagged by a '$'.

RC.F77.TOO.MANY.SUBS

Too many subscripts in an embedded array-element reference.

Explanation: In a Fortran77 array reference embedded in this database query, more subscripts were given than there were dimensions declared for the array.

User Action: Fix this reference.

RC.F77.UNBALANCED.EMBEDDED

Explanation: Parentheses must balance in all Fortran statements. This statement is lacking at least one ')'.

User Action: Fix the parentheses in this statement.

RC.F77.UNCLOSED.QUOTE

Quoted string is missing the terminating quotation mark.

Explanation: A quoted string began in this statement, but the end of the statement was found before a matching quotation mark ended the string.

User Action: Add a terminating quotation mark to the string.

RC.F77.UNDEFINED.PARAM

Syntactically <name> should be a parameter, but it hasn't been defined as one to the precompiler.

Explanation: Only integer literals or integer parameters can appear here. '<name>' is neither.

User Action: Define the parameter, replace with integer literal, or flag the current definition with a '$'.

RC.F77.UNMATCHED.RETLAB

A retrieve loop was to end at statement label <label>, but this statement label was never encountered.

Explanation: There was a 'retrieve <label>' statement in this routine, but no subsequent statement with label <label>.

User Action: Add a statement with label <label>.

RC.F77.VBL.UNDCL

Fortran variable '<name>' undeclared to the precompiler.

Explanation: The variable '<name>' was used in a Fortran expression embedded in an IDL statement, but this variable has never been declared in this routine in a Fortran declaration flagged with a '$'.

User Action: Either add a Fortran declaration for '<name>', or, if the declaration already exists, flag it with a '$' in column 1.

RC.F77.ZERO.DIVIDE

An attempt was made to divide by zero in a constant expression.

Explanation: In the evaluation of an expression made up of integer literals and parameters on this line, an attempt was made to divide by 0.

User Action: Redefine the expression.

RC.FETCH.NO.START

Fetch statement not contained within 'execute' block.

Explanation: Fetch statements must match up with 'retrieve' queries in stored commands, and thus only make sense within the body of a bracketed block following an 'execute' query.

User Action: Either place this statement within an 'execute' block or replace it with a 'retrieve' query.

RC.IDMCOM.NOSTORE

Can not store this command on the database server.

Explanation: Two possible problems: 1) Only the "append, retrieve, replace, and delete" commands may be stored on the database server; 2) All relations and attributes must exist at the time of storing the command.

User Action: Set up the database to conform to the usage, or compile this file without the "name" option.

RC.INTERNAL

System error in the precompiler routine "<routine name>", error=<position>.

Explanation: Previous errors may have caused this error to occur.

User Action: If no previous errors, inform your Britton Lee distribution support person.

RC.LEX.EOFSTRING

End of file encountered in string.

Explanation: The opening quotation of a string is not matched.

User Action: Close the string with a quotation.

RC.LEX.EOLINSTRING

End of line encountered in string.

Explanation: A string is not terminated on this line.

User Action: Use a '\' to break strings across lines.

RC.MACH.DORC.ACTION

System error in precompiler, invalid action. "<action number>"

Explanation: Might be caused by previous errors.

User action: If no previous errors, inform your Britton Lee distribution support person.

RC.MACH.ERRSTATE

System Error in precompiler.

Explanation: Invalid state in precompiler preprocessor will probably fix itself and continue.

User Action: Please inform your Britton Lee distribution support person.

RC.MACH.RC.EXTRA

Extra right curly brace '}' encountered.

Explanation: A right curly brace without a matching left curly has been found.

User Action: Find the mismatch and repair.

RC.NONESTCOM

No nesting of comments allowed.

Explanation: Comments may not be nested: /* /* nested */ */

User Action: Fix the comment.

RC.NO.ALL

The 'all' pseudo-attribute cannot be used in embedded IDL.

Explanation: This form of statement is legal in interactive IDL, but cannot be used in embedded IDL.

User Action: Replace the statement.

RC.NO.AUTOCOMMIT

The 'set autocommit' statement is not used with the precompiler.

Explanation: Although 'set autocommit [on|off]' may be used in interactive SQL, it may not be used as an embedded statement. Use 'begin transaction' instead of 'set autocommit on'. There is no equivalent for 'set autocommmit off'. See the RSC User's Guide section on Transactions.

User Action: Change or remove the statement.

RC.NO.REG.VARS

C variables used with IDL should not be declared 'register'.

Explanation: The C code generated by the precompiler will not compile if variables are declared 'register'.

User Action: Remove the word 'register' from the declaration.

RC.OBJECT.EXISTS

Object named "<object_name>" already exists — can't store commands under this name.

Explanation: An attempt was made to store commands under the name "<object_name>". An object of this name already exists which is not a stored command. This object will not be removed.

User Action: Destroy the object manually or choose another name to store commands under.

RC.ORDER.BY.C.VAR

Attempt to 'order by' a C expression.

Explanation: The object of an 'order by' clause is a C-language expression or variable. This is not allowed.

User Action: Fix or remove the 'order by' clause.

RC.PARSE.BADLEX

Bad C Syntax found near "<lexeme>".

Explanation: Precompiler can not parse this C declaration or expression.

User Action: Find the error in the statement and fix.

RC.PARSE.ID.NOFUNC

"<function>" is not declared as a function.

Explanation: Expression is not a function reference.

User Action: Probably need to declare an identifier as a function.

RC.PARSE.ID.UNDECL

Variable "<variablename>" is undeclared to the precompiler.

Explanation: All identifiers used in an expression referenced by the precompiler must be declared and the declaration must be prefaced by a dollar sign.

User Action: Declare the variable and/or put a dollar sign in front of the declaration.

RC.PNAME.NO.DBNAME

Stored-program-name argument given without database-name argument.

Explanation: If the precompiler is given a name under which to store programs on the database server, it must also be given the name of a database to store them in.

User Action: Give a database-name argument on the command line also.

RC.PREC.SYNTAX.ERR

Bad precompiler syntax found near "<lexeme>".

Explanation: Precompiler can not parse this statement.

User Action: Fix the precompiler statement.

RC.REL.NOEXIST

Relation "<relation name>" is not on the database server.

Explanation: Relation "<relation name>" is referenced but does not exist on the database server.

User Action: Either create "<relation name>" or precompile without the name flag.

RC.STORE.NOEXEC

Execute command not currently supported in stored programs.

Explanation: Not yet implemented. This message may also occur when an IDL command is mistyped.

User action: Get rid of the execute command.

RC.STRAY.C.EXPR

Host language expression at illegal place within embedded IDL statement.

Explanation: A variable name or expression with a prepended "$" occurred at an illegal spot within a database language statement. Host language expressions can only occur at special points within database language statements. User Action: Check the manual pages for RIC (1i) and fix the statement in question.

RC.STRUCTREF.NOTFIELD

"<field>" is not a fieldname within the structure.

Explanation: Attempt to use "<field>" as a structure reference where "<field>" is not declared as a fieldname for this structure.

User Action: Either a typo or incorrect reference. Fix the offending expression.

RC.STRUCTREF.NOTSTRUCT

Application of structure reference on a non-structure.

Explanation: The expression to the left of an arrow (->) or a period is not a valid structure reference.

User Action: Correct the offending expression.

RC.SUB.RANGE

Illegal use of a substitutable range variable.

Explanation: Probable system error.

User Action: Remove previous errors, if the error persists contact your Britton Lee distribution support person.

RC.SYM.BL.OVFL

Block level nesting too great; usually means missing '}'.

Explanation: Precompiler only supports 10 Block levels.

User Action: This is often caused by a missing '}'. If not, arrange declarations of variables to reduce the nesting depth of C scopes.

RC.SYM.BL.UNFL

System error, block level < 1.

Explanation: The precompiler has failed to determine the block level.

User Action: You might have errors above that caused this problem. If so, fix them and rerun.

RC.SYMTAB.OVERFLOW

Symbol table overflow.

Explanation: Number of symbols used by program exceeds symbol table limit. The default is 100 symbols.

User Action: Increase the size of the symbol table by the S option on the rc command line.

RC.SYMTAB.TOO.SMALL

Option specifies too-small symbol table; reset to minimum <minimum symtab size>.

Explanation: A command-line argument was specified to reset the size of the precompiler symbol table. The specified argument was <<minimum symtab size>, which is the minimum size within which anything can be done.

User Action: Specify a larger value, or specify nothing and thus allow the default to be used.

RC.TRANS.SYNTAX

Illegal syntax for "begin transaction".

Explanation: See "RIC User's Guide" for allowable syntax.

User Action: Fix the syntax.

RC.TYPE.APPLY.INVALID

Invalid application of C operator.

Explanation: The operator can not be used with expressions of this type.

User Action: Recode the expression or redeclare the types of the operands.

RC.TYPE.APPLY.NOTYPE

Invalid operator on expression of this type.

Explanation: The operator is used incorrectly in this context.

User Action: Check the types of all operands in the expression and correct the error.

RC.TYPE.BINOP

Types in binary C expression are incompatible.

Explanation: Can not perform arithmetic with these operands.

User Action: Check type declarations of the operands, and correct.

RC.TYPE.PTRORARRAY

Illegal pointer or array type for rc to store into or evaluate.

Explanation: Can only store into pointers or arrays of type char.

User Action: Correct the expression.

RC.UNEXPECTED.EOF

Unexpected end of file.

Explanation: The input file ended in the middle of a statement.

User Action: Correct the final statement of the program.

RC.WITHIN.FETCH

$-flagged query '<query verb>' found within $obtain loop.

Explanation: The query '$ ...' was found within a block controlled by an $obtain statement. No queries are allowed within these blocks.

User Action: Rethink and recode this block.

RC.WITHIN.START

$-flagged query '<query verb>' found within 'execute' block.

Explanation: The query '$ ...' was found within a block controlled by a $execute statement. No queries except $obtain... are allowed within these blocks.

User Action: Rethink and recode this block.

RC.XACT.NESTING

Bad transaction nesting in the above routine.

Explanation: There was an imbalance between the number of "begin transaction" and "end transaction" statements in the above routine. Each "begin transaction" must be matched by a corresponding "end transaction" in the same routine.

User Action: Add, subtract, or rearrange transaction-bracketing statements. Also check for missing or extra curly braces "{...}" (if you're programming in C) or missing '$' on transaction-bracketing statements.

RC.XACT.TOO.DEEP

Too many unmatched "begin transaction" statements (><maximum depth>).

Explanation: "Begin transaction" statements can only be nested <maximum depth> deep within a routine.

User action: Simplify the transaction nesting.

IDMLIB.RCR.CLEAR.REOPEN

System error, can not find the parent of a reopen transaction.

Explanation: Attempting to find parent to end the transaction.

User Action: Contact your Britton Lee distribution support person.

IDMLIB.RCR.DEADLOCKED.SELECT

Deadlocked 'retrieve' outside extended transaction not fixed by system.

Explanation: User should never see this message.

User Action: Contact your Britton Lee distribution support person.

IDMLIB.RCR.INIT.ALREADY

System already initialized, database=<dbname>, program=<progname>

Explanation: Program is reexecuting INITRIC macro.

User Action: Look at flow of program, or look for multiple instances of INITRIC in program.

IDMLIB.RCR.INIT.NODB

No database specified.

Explanation: You need to specify a database, either at precompile time via command line option or by using RCDBNAME within the program.

User Action: Either recompile with "database" flag or use RCDBNAME (see the RIC User's Guide).

IDMLIB.RCR.NOINIT

System not initialized for database <dbname>, program <progname>

Explanation: User has tried to execute an RIC command before the runtime system was initialized. Usually caused by forgetting to use the INITRIC macro.

User Action: Add INITRIC() macro to program, rerun ric and recompile.

IDMLIB.RCR.REOPEN.NEST

Exceeded limit of reopen nesting specified in program.

Explanation: Nesting of retrieves in this reopen transaction is greater than the number specified on the "begin nest <n> transaction".

User Action: Either reduce the level of retrieve nesting, or increase the number specified on the "begin nest <n> transaction" line.

IDMLIB.RCR.STR.TRUNC

The following string is greater than 255 characters: <string>

Expalanation: The database server can only accept character values up to 255 characters long.

User Action: Use shorter character data.

IDMLIB.RCR.TRANS.NESTING

Illegal transaction nesting.

Explanation: Attempting to end a transaction when not in a transaction. Could be caused by using goto or longjmp() out of transaction.

User Action: Fix code.

IDMLIB.RCR.XABORT.USER

Exception to indicate user requested abort transaction.

Explanation: This exception should be caught by runtime code to actually perform the "abort transaction"

User Action: None.

IDMLIB.RCR.XCLEAN

RC system error, rcxclean() called but current rci is not a transaction.

Explanation: rcxclean() is called when a transaction is backed out, but program not currently in transaction.

User Action: Contact your Britton Lee distribution support person.

8. Operator Communication Messages

These messages are generated for operator communications.

8.1. Host Tape

IDMLIB.OPER.HTAPE.EOV

At end of volume.

Explanation: The tape volume is full. Another tape mount request will be forthcoming.

User Action: None yet.

IDMLIB.OPER.HTAPE.ERR.INVALID

Invalid format tape.

Explanation: This tape is not in valid format. It will have to be initialized with inittape(8I) before it can be used. Another tape mount request will be forthcoming.

User Action: None yet.

IDMLIB.OPER.HTAPE.ERR.NODRIVE

Cannot open "<drivename>": <error>.

Explanation: It was not possible to open the tape drive named "<drivename>". The system reported "<error>" as the reason for the failure.

User Action: Correct the failure with the tape drive and continue.

IDMLIB.OPER.HTAPE.ERR.NOHDR1

No HDR1 label on tape.

Explanation: This volume is not properly initialized. Another tape mount request will be forthcoming.

User Action: None yet.

IDMLIB.OPER.HTAPE.ERR.NOTONLINE

Cannot open tape unit <unit>: <errcode>.

Explanation: Tape unit <unit> could not be opened. The system reported <errcode> as the reason.

User Action: None yet.

IDMLIB.OPER.HTAPE.ERR.WRONGTAPE

Incorrect tape.

Explanation: This is not the tape that was requested. Another tape mount request will be forthcoming.

User Action: None yet.

IDMLIB.OPER.HTAPE.ERR.WRONGVOL

Incorrect volume: "<needed>" required, "<actual>" mounted

Explanation: The operator has mounted volume "<actual>", although volume "<needed>" was specified.

User Action: The correct volume should be mounted.

IDMLIB.OPER.HTAPE.ERR.WRONGVOLUME

Wrong volume

Explanation: The incorrect volume has been mounted. Another tape mount request will be forthcoming.

User Action: None yet.

IDMLIB.OPER.HTAPE.FILENOTFOUND

File not found on this volume of host tape.

Explanation: The required file was not found on this volume. Another tape mount request will be forthcoming.

User Action: None yet.

IDMLIB.OPER.HTAPE.MOUNT

Please mount volume <volume> on host tape unit <unit>.

Explanation: The user has issued a tape mount request for volume <volume> on tape unit <unit>.

User Action: None yet.

IDMLIB.OPER.HTAPE.NEXTVOLUME

Ready for next volume.

Explanation: This volume has been exceeded. Another tape mount request will be forthcoming.

User Action: None yet.

8.2. Database Server Tape

IDMLIB.OPER.ITAPE.MOUNT

Please mount volume "<volume>", reel 0 on tape unit <unit>.

Explanation: The volume "<volume>" has been requested for database server tape unit <unit>.

User Action: Mount volume "<volume>".

IDMLIB.OPER.ITAPE.NEXT

Please mount database server tape volume number <volume number> and enter database server tape unit number.

Explanation: The next volume of a multivolume database server tape operation must be mounted.

User Action: Mount the tape on any unit, and enter the unit number.

IDMLIB.OPER.ITAPE.SYNTAX

Invalid syntax — use numeric volume number or "$abort" to abort.

Explanation: You have not correctly entered a tape unit number. You may enter a tape unit number (possibly followed by an exclamation point to ignore volume header checking) or "$abort <reason>" to abort this job. The reason will be reported to the user.

User Action: Reenter the unit number.