Oracle7 Server Messages

This chapter lists the messages issued by the Oracle Server. Each message code has the format, ORA-*nnnnn*, where *nnnnn* is an integer.

Oracle Server messages specific to a particular operating system are *not* listed in this chapter, although the message range is listed. For information about messages specific to your operating system, see "Related Publications" in the Preface.

00000-00099: Oracle Server

This section lists the messages generated by the Oracle Server.

ORA-00000

ORA-00000   **normal, successful completion**

Cause: An operation has completed normally, having met no exceptions.

Action: No action required.

ORA-00001

ORA-00001	**unique constraint *table.column* violated**

Cause:	An update or insert statement attempted to insert a duplicate key.
Action:	Either remove the unique restriction or do not insert the key.
	For Trusted Oracle7 configured in DBMS MAC mode, you may see this message if a duplicate entry exists at a different level.

ORA-00017

ORA-00017   **session requested to set trace event**

Cause: The current session was requested to set a trace event by another session.

Action: This is used internally; no action required.

ORA-00018

ORA-00018   **maximum number of sessions exceeded**

Cause: An operation requested a resource that was unavailable. The maximum number of sessions is specified by the initialization parameter SESSIONS. When this maximum is reached, no more requests are processed.

Action: Try the operation again in a few minutes. If this message occurs often, shut down Oracle, increase the SESSIONS parameter in the initialization parameter file, and restart Oracle.

ORA-00019

ORA-00019   **maximum number of session licenses exceeded**

Cause: All licenses are in use.

Action: Call customer support before increasing the number of session licenses.

ORA-00020

ORA-00020   **maximum number of processes *num* exceeded**

Cause: An operation requested a resource that was unavailable. The maximum number of processes is specified by the initialization parameter PROCESSES. When this maximum is reached, no more requests are processed.

Action: Try the operation again in a few minutes. If this message occurs often, shut down Oracle, increase the PROCESSES parameter in the initialization parameter file, and restart Oracle.

ORA-00021

ORA-00021   session attached to some other process; cannot switch session

Cause: The user session is currently being used by someone else.

Action: Do not execute calls in a user session concurrently.

ORA-00022

ORA-00022 **invalid session id; access denied**

Cause: Either the session specified does not exist, or the caller does not have the privilege to access it.

Action: Specify a valid session ID. This requires ownership of the session ID or the CHANGE_USER privilege.

ORA-00023

ORA-00023 **session references process' private memory; cannot detach session**

Cause: An attempt was made to detach the current session, which contains references to the process' private memory.

Action: A session may contain references to process memory (PGA) if it has an open network connection, a very large context area, or operating system privileges. To allow the detach, it may be necessary to close the session's database links or cursors. Detaching a session with operating system privileges is always disallowed.

ORA-00024

ORA-00024 **logins from more than one process not allowed in single-process mode**

Cause: An attempt was made to log in more than once from different processes after Oracle had been started in single-process mode.

Action: When Oracle is running in single-process mode, log out of the previous process before trying to log in from another one.

ORA-00025

ORA-00025 **failed to allocate** *num*

Cause: An allocation failed because Oracle ran out of memory.

Action: Restart Oracle with a larger SGA heap.

ORA-00026

ORA-00026 **missing or invalid session id**

Cause: The session ID string specified in the ALTER SYSTEM KILL SESSION command was invalid, or no string was specified.

Action: Retry the command with a valid session ID.

ORA-00027

ORA-00027 **cannot kill current session**

Cause: An attempt was made to kill the current session with the ALTER SYSTEM KILL SESSION command.

Action: If it is necessary to kill the current session, do so from another session.

ORA-00028

ORA-00028   **your session has been killed**

Cause: A privileged user killed the session and it is no longer logged in tothe database.

Action: Contact the database administrator. The administrator may be attempting to perform an operation that requires users to be logged out. When the database administrator announces that the database is available, log in and resume work.

ORA-00029

ORA-00029   **session is not a user session**

Cause: The session ID specified in an ALTER SYSTEM KILL SESSION command was not a user session. For example, the session may be recursively started.

Action: Retry the command with a valid session ID.

ORA-00030

ORA-00030   user session ID does not exist

Cause: The user session ID no longer exists, probably because the session was logged out.

Action: Use a valid session ID.

ORA-00031

ORA-00031   session marked for kill

Cause: The session specified in the ALTER SYSTEM KILL SESSION command cannot be killed immediately because the session is involved in a non-interruptible operation. For example, rolling back a transaction or being blocked by a network operation. The session has been marked to be killed as soon as possible after the current operation is done.

Action: No action required. The session will be killed when the operation is completed. Further executions of the ALTER SYSTEM KILL SESSION command may cause the session to be killed sooner.

ORA-00033

ORA-00033   **current session has empty migration password**

Cause: An attempt was made to detach or clone the current session, and it has an empty migration password. This is not allowed.

Action: Create the session with a non-empty migration password.

ORA-00034

ORA-00034   cannot COMMIT or ROLLBACK in current PL/SQL session

Cause: An attempt was made to ROLLBACK from a PL/SQL object (procedure, function, package) in a session where COMMIT and ROLLBACK are disabled by an ALTER SESSION DISABLE COMMIT IN PROCEDURE statement.

Action: Enable COMMITs from PL/SQL in this session or do not attempt a COMMIT or ROLLBACK when disabled.

ORA-00035

ORA-00035 **LICENSE_MAX_USERS cannot be less than current number of users**

Cause: Specified value for LICENSE_MAX_USERS is less than current numberof users.

Action: Check the license limit and drop extra users or purchase more licenses.

ORA-00050

ORA-00050 O/S error occurred while obtaining an enqueue. See O/S error

Cause: Could not obtain the operating system resources necessary for an Oracle enqueue. This most often occurs when the user resource quota on the operating system is too low.

Action: Look up the operating system error in the operating-system-specific Oracle documentation and perform the needed action.

ORA-00051

ORA-00051   **time-out occurred while waiting for resource**

Cause: This message is usually caused by an instance that has terminated abnormally.

Action: Restart any non-recovered instances.

ORA-00052

ORA-00052 **maximum number of enqueue resources *num* exceeded**

Cause: An operation requested a resource that was unavailable. The maximum number of enqueue resources is specified by the initialization parameter ENQUEUE_RESOURCES. When this maximum is reached, no more requests are processed.

Action: Try the operation again in a few minutes. If this message occurs often, shut down Oracle, increase the ENQUEUE_RESOURCES parameter in the initialization parameter file, and restart Oracle.

ORA-00053

ORA-00053   **maximum number of enqueues exceeded**

Cause: An operation requested a resource that was unavailable and the enqueue list for the resource had reached its maximum length. If a request is made for a resource that is unavailable, the request is enqueued to wait for the resource. The number of requests that may be queued for a given resource is specified by the initialization parameter ENQUEUE_RESOURCES. When this maximum is reached, no more requests may be added to the enqueue list.

Action: Try the operation again in a few minutes. If this message occurs often, shut down Oracle, increase the ENQUEUE_RESOURCES parameter in the initialization parameter file, and restart Oracle.

ORA-00054

ORA-00054   **resource busy and acquire with NOWAIT specified**

Cause: The NOWAIT keyword forced a return to the command prompt because a resource was unavailable for a LOCK TABLE or SELECT FOR UPDATE command.

Action: Try the command after a few minutes or enter the command without the NOWAIT keyword.

ORA-00055

ORA-00055   **maximum number of DML locks exceeded**

Cause: An operation requested a resource that was unavailable. The maximum number of DML locks is specified by the DML_LOCKS parameter in the initialization parameter file. When this maximum is reached, no more requests are processed.

Action: Try the operation again in a few minutes. If this message occurs often, shut down Oracle, increase the DML_LOCKS parameter in the initialization parameter file, and restart Oracle.

ORA-00056

ORA-00056   **DDL lock on object "*str.name*" already held in an incompatible mode**

Cause: The attempted lock is incompatible with the DDL lock already held on the object. This usually occurs when attempting to drop a table that has parse locks.

Action: Before attempting to drop a table, check that it has no parse locks. Wait a few minutes before retrying the operation.

ORA-00057

ORA-00057   **maximum number of temporary table locks exceeded**

Cause: The number of temporary tables equals or exceeds the number of temporary table locks. Temporary tables are often created by large sorts.

Action: Increase the TEMPORARY_TABLE_LOCKS parameter and restart Oracle.

ORA-00058

ORA-00058   DB_BLOCK_SIZE must be *num* to mount this database not *num*

Cause: The DB_BLOCK_SIZE parameter value in the initialization parameter fileused to start a database does not match the value used when that databasewas created.

Potential reasons for this mismatch are
- mounting the wrong database
- using the wrong initialization parameter file
- the DB_BLOCK_SIZE value was changed

Action: For one of the above causes, either
- mount the correct database
- use the correct initialization parameter file
- correct the DB_BLOCK_SIZE parameter value

ORA-00059

ORA-00059   **maximum number of DB_FILES exceeded**

Cause: An unavailable resource was requested. The maximum number of datafiles is specified by the DB_FILES parameter in the initialization parameter file. When this maximum is reached, no more requests are processed.

Action: Try again when the resource is freed. If this message occurs often, Oracle must be shut down and restarted after increasing the DB_FILES parameter in the initialization parameter file. If the DB_FILES parameter cannot be changed because it is already set to the MAXDATAFILES parameter value, set at database creation, you must create a new control file.

ORA-00060

ORA-00060   deadlock detected while waiting for resource

Cause: Your session and another session are waiting for a resource locked by the other. This condition is known as a deadlock. To resolve the deadlock, one or more statements were rolled back for the other session to continue work.

Action: Either
- Enter a ROLLBACK statement and re-execute all statements since the last commit.
- Wait until the lock is released, possibly a few minutes, and re-execute the rolled back statements.

ORA-00061

ORA-00061   another instance has a different DML_LOCKS setting

Cause: The shared instance being started is using DML locks and the running instances are not, or vice versa.

Action: Either
- Do not use DML_LOCKS by setting DML_LOCKS to zero at all instances.
- Use DML_LOCKS by setting DML_LOCKS to a positive integer at all instances.

ORA-00062

ORA-00062 **DML full-table lock cannot be acquired; DML_LOCKS is 0**

Cause: The instance was started with the initialization parameter DML_LOCKS set to zero, but the statement being executed needs a full-table lock (S, X, or SRX).

Action: Set DML_LOCKS to a non-zero value, restart the instance, and re-enter the statement.

ORA-00063

ORA-00063 LOG_FILES initialization parameter exceeded

Cause: The value of the LOG_FILES initialization parameter was exceeded.

Action: Increase the value of LOG_FILES and warm start Oracle. The value needs to be as large as the highest number log that currently exists rather than just the count of the logs that exist. If this parameter cannot be increased because it would exceed the MAXLOGFILES set at database creation, you must create a new control file first.

ORA-00064

ORA-00064 **object is too large to allocate on this O/S *num, num***

Cause: The initialization parameter DB_BLOCK_SIZE is set to a value that calls for more contiguous space than can be allocated on the operating system being used.

Action: Reduce the value of DB_BLOCK_SIZE so that the requested contiguous space is within the capacity of the operating system.

ORA-00065

ORA-00065   **initialization of FIXED_DATE failed**

Cause: An attempt was made to enter a fixed date string that was not in the proper format.

Action: Enter the date string using the format YYYY-MM-DD:HH24:MI:SS.

ORA-00066

ORA-00066   **LOG_FILES is _num_ but needs to be _num_ to be compatible**

Cause: The maximum number of redo log files supported by this instance is not the same as for the other instances. All instances must be able to open all the files any instance can open.

Action: Check that the LOG_FILES initialization parameter is the same for all instances.

ORA-00067

ORA-00067 **invalid value *num* for parameter *num*, must be at least *num***

Cause: The value for the initialization parameter is invalid.

Action: Choose a value as indicated by the message. Change the value of the LOG_FILES parameter to be compatible.

ORA-00068

ORA-00068   **invalid value *num* for parameter *num*, must be between *num* and *num***

Cause: The value for the initialization parameter is invalid.

Action: Choose a value as indicated by the message. Change the value of the LOG_FILES parameter to be compatible.

ORA-00069

ORA-00069 **cannot acquire lock -- table locks disabled for** *name*

Cause: A command was issued that tried to lock the table indicated in the message. Examples of commands that can lock tables are LOCK TABLE, ALTER TABLE ... ADD (...), and so on.

Action: Use the ALTER TABLE ... ENABLE TABLE LOCK command, and retry the command.

ORA-00097

ORA-00097   **use of Oracle SQL feature not in SQL92 *compliance* level**

Cause: You tried to use an Oracle SQL feature that is not compliant with the SQL92 standard.

Action: Do not use the feature or use the ALTER SESSION SET FLAGGER command to set the appropriate level of SQL92 compliance.

00100-00150: Multi-threaded Server

This section lists message generated by the multi-threaded server.

ORA-00100

ORA-00100   **idle public server terminating**

Cause: There are too many idle shared servers waiting on the common dispatcher request queue.

Action: No action required.

ORA-00101

ORA-00101   **invalid specification for initialization parameter MTS_DISPATCHERS**

Cause: The syntax for the MTS_DISPATCHERS parameter is incorrect.

Action: Enter the MTS_DISPATCHERS parameter into the initialization parameter file with the correct syntax and then shut down and restart the instance.

ORA-00102

ORA-00102   **network protocol *str* cannot be used by the dispatchers**

Cause: The network specified in MTS_DISPATCHERS initialization parameter does not have the functionality required by the dispatchers.

Action: Refer to your Oracle7 operating system-specific documentation for network protocols supported by the dispatchers.

ORA-00103

ORA-00103   **invalid network protocol; reserved for use by dispatchers**

Cause: The network specified in SQL*Net CONNECT string is reserved for use by the dispatchers.

Action: Specify other network protocols in the CONNECT string.

ORA-00104

ORA-00104   **deadlock detected; all public servers blocked waiting for resource**

Cause: This message occurs when a client locks a resource and the maximum number of shared servers are taken by other clients who are requesting a locked resource. The original client is unable to get a shared server and cannot release the lock on the resource.

Action: The system automatically starts up new servers to break the deadlock until the number of servers reaches the value specified in MTS_MAX_SERVERS. If this problem occurs frequently, ensure more shared servers will be available at the next restart of the instance by increasing the value of the initialization parameter MTS_SERVERS or MTS_MAX_SERVERS.

ORA-00105

ORA-00105   dispatching mechanism not configured to support network protocol

Cause: The ALTER SYSTEM SET MTS_DISPATCHERS command was used to alter a set of dispatchers for a network protocol that is not specified in the initialization parameter file.

Action: Do either of the following:
- Include the network protocol in the MTS_DISPATCHERS initialization parameter and restart the system.
- Use the ALTER SYSTEM SET MTS_DISPATCHERS command with a network protocol that exists in the initialization parameter file.

ORA-00106

ORA-00106   **cannot startup/shutdown database when connected to a dispatcher**

Cause: An attempt was made to start up or shut down an instance while connected to a shared server process via a dispatcher.

Action: Reconnect to the database using a dedicated server.

ORA-00107

ORA-00107   **failed to connect to network listener process**

Cause: Usually caused by the fact that the network listener process has not been started.

Action: Check for the following:
- The network configuration file is not set up correctly.
- The client side address is not specified correctly.
- The listener initialization parameter file is not set up correctly.

ORA-00108

ORA-00108   **failed to set up dispatcher to accept connection asynchronously**

Cause: The network protocol used by the dispatcher does not support asynchronous operations.

Action: Contact customer support.

ORA-00111

ORA-00111 **Warning: maximum number of servers is only *num*; adding *num* servers**

Cause: An attempt was made to start more shared server processes than the number allowed by the initialization parameter MTS_MAX_SERVERS. The actual number of shared servers started is specified in the message.

Action: If more shared server processes are required, increase the initialization parameter MTS_MAX_SERVERS and restart the instance.

ORA-00112

ORA-00112   **only created up to *num* maximum specified dispatchers**

Cause: An attempt was made to start up more dispatchers than the maximum number specified by the initialization parameter MTS_MAX_DISPATCHERS.

Action: If more dispatchers are required, increase MTS_MAX_DISPATCHERS, restart the instance, and add dispatchers using the ALTER SYSTEM MTS_DISPATCHERS command.

ORA-00113

ORA-00113   **protocol name *string* is too long**

Cause: A protocol name specified in the MTS_DISPATCHERS initialization parameter is too long.

Action: Use a valid protocol name for the MTS_DISPATCHERS initialization parameter.

ORA-00114

ORA-00114   **missing value for initialization parameter MTS_SERVICE**

Cause: There is no value for the MTS_SERVICE initialization parameter, nor for the DB_NAME parameter.

Action: Add an MTS_SERVICE or DB_NAME definition to the initialization parameter file. By default, MTS_SERVICE is the value of DB_NAME unless MTS_SERVICE is explicitly specified.

ORA-00115

ORA-00115   **connection refused; dispatcher connection table is full**

Cause: A connection request was refused by a dispatcher because the dispatcher cannot support any more connections.

Action: Connect to a different dispatcher or use a dedicated server.

ORA-00116

ORA-00116   **MTS_SERVICE name is too long**

Cause: The service name specified in the MTS_SERVICE initialization parameter istoo long.

Action: Use a shorter name for the MTS_SERVICE value (maximum is 255 characters).

ORA-00117

ORA-00117   **value out of range for the MTS_SERVERS system parameter**

Cause: The value of the MTS_SERVERS parameter is incorrect.

Action: The MTS_SERVERS parameter must be from 0 to MTS_MAX_SERVERS.

ORA-00118

ORA-00118 **value out of range for MTS_DISPATCHERS system parameter**

Cause: The value of the MTS_DISPATCHERS parameter is incorrect.

Action: The sum of all MTS_DISPATCHERS values must be from 0 to MTS_MAX_DISPATCHERS.

ORA-00119

ORA-00119   **invalid specification for system parameter MTS_LISTENER_ADDRESS**

Cause: The syntax for the MTS_LISTENER_ADDRESS is incorrect.

Action: Refer to the *Oracle7 Server Administrator's Guide* for the correct syntax.

ORA-00120

ORA-00120   **dispatching mechanism not enabled or installed**

Cause: Either the dispatching mechanism is not enabled or the Oracle executable image is not installed with the dispatching mechanism.

Action: Refer to the *Oracle7 Server Administrator's Guide* and your Oracle7 operating system-specific documentation for procedures for starting and configuring the dispatcher processes.

ORA-00121

ORA-00121 **MTS_SERVERS specified without MTS_DISPATCHERS**

Cause: A definition for MTS_SERVERS was specified in the INIT.ORA file, but no definition for MTS_DISPATCHERS was specified. If no value is specified for MTS_DISPATCHERS, shared servers will not be used.

Action: Add a definition for MTS_DISPATCHERS in the INIT.ORA file. If you do not intend to use shared servers, remove the MTS_SERVERS definition.

ORA-00122

ORA-00122   **cannot initialize network configuration**

Cause: Oracle could not initialize SQL*Net Version 2.

Action: See accompanying messages.

00150-00159: Oracle*XA

ORA-00150

ORA-00150 **duplicate transaction ID**

Cause: Attempted to start a new transaction with an ID already in use by an existing transaction.

Action: Check that your application uses a valid transaction ID.

ORA-00151

ORA-00151   **invalid transaction ID**

Cause: The specified transaction ID does not correspond to an existing valid transaction.

Action: Check that your application uses a valid transaction ID.

ORA-00152

ORA-00152   **current session does not match requested session**

Cause: The current session is not the same as the session that was passed into a UPIXADO() call.

Action: Check that your application is coded correctly.

00200-00249: Control Files

ORA-00200

ORA-00200   **cannot create control file *name***

Cause: The control file cannot be created, usually because of insufficient disk storage or filename conflicts.

Action: Check that there is sufficient disk space and no conflicts in filenames and try to create the control file again.

ORA-00201

ORA-00201   **control file version *num* incompatible with Oracle version *num***

Cause: The control file was created by a different version of Oracle.

Action: Either
- Restart with version of Oracle used to create the control file.
- Use the CREATE CONTROLFILE command to create a new control file that is compatible with this release.

ORA-00202

ORA-00202   **control file:** *name*

Cause: This message reports the name of the file involved in other messages.

Action: See the associated messages for a description of the problem.

ORA-00203

ORA-00203   **using the wrong control files**

Cause: The mount ID in the control file is not the same as the mount ID in the control file used by the first instance to mount this database. The control files are for the same database but are not the same files. The most likely cause is that one instance is using a backup of the control file. If you want to use a backed up control file, you can recover using the USING BACKUPCONTROLFILE option.

Action: Check that all instances are using the correct version of the control file.

ORA-00204

ORA-00204 **error in reading control file** *name* **block** *num*, **blocks** *num*

Cause: A disk read-failure occurred while attempting to read the specified control file. The block location of the failure is given.

Action: Check that the disk is online. If it is not, bring it online and shut down and restart Oracle. If the disk is online, then look for operating system reasons for Oracle's inability to read the disk or control file. Refer to the *Oracle7 Server Administrator's Guide* for information about recovering from the loss of a control file. See also your operating system-specific Oracle documentation.

ORA-00205

ORA-00205 error in identifying control file *name*

Cause: The system could not find a control file of the specified name and size.

Action: Either
- Check that the proper control filename is referenced in the CONTROL_FILES initialization parameter in the initialization parameter file and try again.
- When using mirrored control files, that is, more than one control file is referenced in the initialization parameter file, remove the control filename listed in the message from the initialization parameter file and restart the instance. If the message does not recur, remove the problem control file from the initialization parameter file and create another copy of the control file with a new filename in the initialization parameter file.

ORA-00206

ORA-00206   **error in writing control file** *name* **block** *num***,   blocks** *num*

Cause: A disk write-failure occurred while attempting to write to the specified control file. The block location of the failure is given.

Action: Check that the disk is online. If it is not, bring it online and shut down and restart Oracle. If the disk is online, then look for operating system reasons for Oracle's inability to write to the disk or control file. See the *Oracle7 Server Administrator's Guide* for information on recovering from the loss of a control file.See also your operating system-specific Oracle documentation.

ORA-00207

ORA-00207   **control files are not for the same database**

Cause: The database ID in the control file is not the same as the database ID in the control file used by the first instance to mount this database. The most likely cause is either that one of the mounts used the wrong control file or there are two databases with the same name.

Action: Check that the control file is for the correct database and not an old version. When using multiplexed control files, that is, more than one control file is referenced in the initialization parameter file, remove the control filename listed in the message from the initialization parameter file and restart the instance. If the message does not recur, remove the problem control file from the initialization parameter file and create another copy of the control file using a new filename in the initialization parameter file.

ORA-00208

ORA-00208   **number of control file names exceeds limit of *num***

Cause: An attempt was made to use more control files than Oracle supports. The limit is given in the message.

Action: Shut down Oracle, reduce the number of control filenames specified in the CONTROL_FILES parameter in the initialization parameter file, and restart Oracle. Delete unused files.

ORA-00209

ORA-00209   **block size *num* exceeds limit of *num* bytes**

Cause: A block size larger than that allowed by the operating system was specified.

Action: Reduce the block size to the amount specified in the message and try again.

ORA-00210

ORA-00210 **cannot open control file** *name*

Cause: The system was unable to open a control file.

Action: Check that the control file exists, that the storage device is online, and that the file is not locked by some other program and try again. Also, check to see that the operating system limit on the number of open files per process has not been exceeded.

When using multiplexed control files, that is, more than one control file is referenced in the initialization parameter file, remove the parameter from the initialization parameter file referencing the control filename indicated in the message and restart the instance. If the message does not recur, remove the problem control file from the initialization parameter file and create another copy of the control file using a new filename in the initialization parameter file.

ORA-00211

ORA-00211   **control file *name* does not match previous control files**

Cause: The specified control file is from another database.

Action: Locate and specify the correct control file for this database, then retry the operation. When using multiplexed control files, that is, more than one control file is referenced in the initialization parameter file, remove the control filename listed in the message from the initialization parameter file and restart the instance. If the message does not recur, remove the problem control file from the initialization parameter file, create another copy of the control file, and include the new filename in the initialization parameter file.

ORA-00212

ORA-00212 **block size *num* below minimum required size of *num* bytes**

Cause: The specified block size is too small. Additional space is needed for system overhead.

Action: Specify a larger block size and retry the operation.

ORA-00213

ORA-00213   **cannot reuse control file *name*; old file size *num*, *num* required**

Cause: To reuse a control file, it must be the same size as the previous one used.

Action: In the CREATE DATABASE statement, do not specify REUSE.

ORA-00214

ORA-00214   **control file** *name* **version** *num* **inconsistent with file** *name* **version** *num*

Cause: An inconsistent set of control files, datafiles, and redo log files was used.

Action: Use a consistent set of control files, datafiles, and redo log files. That is, all the files must be for the same database and from the same time period.

ORA-00215

ORA-00215   **must be at least one control file**

Cause: No control file was specified, or the control file specified does not exist.

Action: Specify at least one valid control file and retry the operation.

ORA-00216

ORA-00216   **unable to determine physical block size for control file** *name*

Cause: An error occurred while determining the physical block size of the specified control file.

Action: Check the accompanying message stack for more detailed information.

ORA-00217

ORA-00217 **control file *name* physical block size *num* inconsistent with *num***

Cause: The physical block size of the operating system is inconsistent with the block size of the control file.

Action: The system will not operate with invalid control files. Either restore a valid control file or recreate the database. When using multiplexed control files, that is, more than one control file is referenced in the initialization parameter file, remove the control filename listed in the message from the initialization parameter file and restart the instance. If the message does not recur, remove the problem control file from the initialization parameter file and create another copy of the control file with a new filename in the initialization parameter file.

ORA-00218

ORA-00218 **control file *name* was created with physical size *num* now is *num***

Cause: The control file size specified in the file's header differs from the physical block size of the control file. This usually means that the control file has been corrupted.

Action: Recover the control file as described in the *Oracle7 Server Administrator's Guide*

ORA-00219

ORA-00219   **required control file size *num* larger than maximum *num***

Cause: The CREATE DATABASE statement specified a combination of initialization parameters that results in the control file exceeding the internal maximum size.

Action: You must recreate the database. In the CREATE DATABASE statement, use a value less than the maximum permissible value for clauses such as MAXDATAFILES and MAXLOGFILES as described in the *Oracle7 Server Administrator's Guide*

ORA-00220

ORA-00220   **control file *name* not mounted by first instance**

Cause: The specified control file has a different mount ID than the other control files that are being mounted. This means that the first instance to mount the database did not use this control file.

Action: Find and use the correct control file.

ORA-00221

ORA-00221 **error on write to control file**

Cause: An error occurred when writing to one or more of the control files.

Action: See accompanying messages.

ORA-00222

ORA-00222   **control file may not be queried using dispatcher**

Cause: Attempted to use client process connected to dispatcher. This is not allowed for queries to control file fixed tables due to the memory requirements.

Action: Connect directly to the instance, then execute the query.

ORA-00223

ORA-00223   **conversion datafile is invalid or incorrect version**

Cause: A Version 6 to Oracle7 conversion datafile contains invalid data or was created with an old version of the migration utility.

Action: Regenerate the conversion file with the correct version of the migration utility or use the correct file for converting the database.

ORA-00224

ORA-00224   the specified file is not a control file

Cause: The file specified in the initialization parameter file is not a control file.

Action: Edit the initialization parameter file and specify the correct control filename.

00250-00299: Archiving and Recovery

This section lists the messages generated when archiving or recovery of the database is in progress.

ORA-00250

ORA-00250   **archiver not started**

Cause: An attempt was made to stop automatic archiving, but the archiver process was not running.

Action: No action required.

ORA-00252

ORA-00252	**log *name* of thread *num* is empty, cannot archive**
Cause:	The specified redo log was not used since it was introduced to the database. It is also possible that the instance died during a log switch, and the log was left empty.
Action:	Empty logs do not need to be archived. Do not attempt to archive the redo log file.

ORA-00253

ORA-00253   **limit of *num* exceeded by length *num* of archive string *name***

Cause: The specified archive string is too long. The limit is given in the message.

Action: Use a shorter string, making sure not to exceed the given limit and try again.

ORA-00254

ORA-00254   **error in archive control string *archive_log_location***

Cause: The specified archive log location is invalid in the archive command or the LOG_ARCHIVE_DEST initialization parameter.

Action: Specify a valid location name.

ORA-00255

ORA-00255 **error archiving log** *name* **of thread** *num***, sequence** *num*

Cause: An error occurred during archiving.

Action: Check the accompanying message stack for more detailed information.

If the online log is corrupted then the log can be cleared using the UNARCHIVED option. This will make any existing backups useless for recovery to any time after the log was created, but does allow redo to be generated.

ORA-00256

ORA-00256   **error occurred in translating archive text string *str***

Cause: An error occurred while translating the archive control string.

Action: Check the accompanying message stack for more detailed information.

ORA-00257

ORA-00257 archiver is stuck. CONNECT INTERNAL only, until freed

Cause: The ARCH process received an error while trying to archive a redo log file. If the problem is not resolved soon, the database will stop executing transactions. The most likely cause of this message is that the destination device is out of space to store the redo log file.

Action: Check the archiver trace file for a detailed description of the problem. Also, verify that the device specified in the initialization parameter ARCHIVE_LOG_DEST is set up properly for archiving.

ORA-00258

ORA-00258	**manual archiving in NOARCHIVELOG mode must identify log**

Cause: An attempt was made to archive a redo log file manually without specifying the sequence number, group number, or filename while the database was in NOARCHIVELOG mode.

Action: The name, group number, or thread and sequence number of redo log files must be specified to archive redo log files manually while the database is in NOARCHIVELOG mode.

ORA-00259

ORA-00259 **log** *name* **of open thread** *num* **is the current log, cannot archive**

Cause: An attempt was made to archive the current log of an open thread. This is not allowed because the redo log file may still be in use for generation of redo entries.

Action: Force a log switch in the instance where the thread is open. If no instances are open, open the database so instance recovery can recover the thread.

ORA-00260

ORA-00260   cannot find online log sequence *num* for thread *num*

Cause: The log sequence number specified in the ARCHIVE statement does not match any of the online logs for the specified thread. This can result from any of the following situations:

- This log may have been reused for another sequence number.
- This log may have been dropped.
- The sequence number may be greater than the current log sequence number.
- The thread may not have any logs.

Action: Check the ARCHIVE statement; then specify a valid log sequence number.

ORA-00261

ORA-00261 **log *name* of thread *num* is being archived or modified, cannot archive**

Cause: Either the log is being archived by another process or an administrative command is executing and modifying the log. Clearing, adding a member, dropping a member, renaming a member, and dropping the log are operations that modify a log.

Action: Wait for the current operation to complete and try again.

ORA-00262

ORA-00262 **current log *name* of closed thread *num* cannot switch**

Cause: The log cannot be cleared or manually archived because it is the current log of a closed thread, and it is not possible to switch logs so another log is current. All other logs for the thread need to be archived or cleared and cannot be reused.

Action: Archive another log in the same thread first or complete the clearing. See attached errors for the reason the switch cannot be completed.

ORA-00263

ORA-00263   **there are no logs that need archiving for thread** *num*

Cause: An attempt was made to archive the unarchived logs in this thread manually, but no logs need archiving.

Action: No action required.

ORA-00264

ORA-00264   no recovery required

Cause: An attempt was made to perform media recovery on files that do not need any type of recovery.

Action: Do not attempt to perform media recovery on the selected files. Check to see that the filenames were entered properly. If not, retry the command with the proper filenames.

ORA-00265

ORA-00265 **instance recovery required, cannot set ARCHIVELOG mode**

Cause: The database either crashed or was shut down with the ABORT option. Media recovery cannot be enabled because the online logs may not be sufficient to recover the datafiles.

Action: Open the database and then enter the SHUTDOWN command with the NORMAL or IMMEDIATE options.

ORA-00266

ORA-00266　**name of archived log file needed**

Cause:　　　During media recovery, the name of an archived redo log file was requested, but no name was entered.

Action:　　　Mount the correct redo log file and enter its name when it is requested.

ORA-00267

ORA-00267   **name of archived log file not needed**

Cause: During media recovery, the name of an archived redo log file was entered, but no name was requested.

Action: Continue media recovery, but do not enter a new log name.

ORA-00268

ORA-00268 **specified log file does not exist** *name*

Cause: The given redo log file does not exist.

Action: Check the spelling and capitalization of the filename and retry the command.

ORA-00269

ORA-00269 **specified log file is part of thread** *num* **not** *num*

Cause: The given redo log file is not part of the given thread.

Action: Check that the thread of the redo log file matches the thread on the command line. If not, use a redo log file from the appropriate thread. Retry the command after correcting the error.

ORA-00270

ORA-00270 **error creating archive log**

Cause: An error was encountered when either creating or opening the destination file for archiving.

Action: Check that the archive destination is valid and that there is sufficient space on the destination device.

ORA-00271

ORA-00271   **there are no logs that need archiving**

Cause: An attempt was made to archive the unarchived redo log files manually, but there are no files that need to be archived.

Action: No action required.

ORA-00272

ORA-00272   error writing archive log

Cause: An I/O error occurred while archiving a redo log file.

Action: Check that the output device is still available and correct any device errors that may have occurred. Also, make certain that sufficient space for archiving is available on the output device.

ORA-00273

ORA-00273   **media recovery of direct load data that was not logged**

Cause: A media recovery session encountered a table that was loaded by the direct loader without logging any redo information. Some or all of the blocks in this table are now marked as corrupt.

Action: The table must be dropped or truncated so that the corrupted blocks can be reused. If a more recent backup of the file is available, try to recover this file to eliminate this error.

ORA-00274

ORA-00274 **illegal recovery option** *str*

Cause: An illegal option was specified for a recovery command.

Action: Correct the syntax and retry the command.

ORA-00275

ORA-00275   media recovery has already been started

Cause: An attempt was made to start a second media recovery operation in the same session.

Action: Complete or cancel the first media recovery session or start another session to perform media recovery.

ORA-00276

ORA-00276 **CHANGE keyword specified but no change given**

Cause: The CHANGE keyword was specified on the command line, but no change number was given.

Action: Retry the command using a valid change number after the CHANGE keyword.

ORA-00277

ORA-00277   **illegal option to the UNTIL recovery flag** *str*

Cause: Only CANCEL, CHANGE and TIME can be used with the UNTIL keyword.

Action: Correct the syntax.

ORA-00278

ORA-00278	**log file *name* no longer needed for this recovery**

Cause: The specified redo log file is no longer needed for the current recovery.

Action: No action required. The archived redo log file may be removed from its current location to conserve disk space, if needed. However, the redo log file may still be required for another recovery session in the future.

ORA-00279

ORA-00279   **change *num* generated at *name* needed for thread *num***

Cause: The requested log is required to proceed with recovery.

Action: Please specify the requested log in the command or cancel recovery.

ORA-00280

ORA-00280   change *num* for thread *num* is in sequence *num*

Cause: This message helps to locate the redo log file with the specified change number requested by other messages.

Action: Use the information provided in this message to specify the required archived redo log files for other errors.

ORA-00281

ORA-00281	**media recovery may not be performed using dispatcher**

Cause:	An attempt was made to use a dispatcher process for media recovery. Memory requirements disallow this recovery method.
Action:	Connect to the instance via a dedicated server process to perform media recovery.

ORA-00282

ORA-00282   **UPI *name* call not supported, use ALTER DATABASE RECOVER**

Cause: The given UPI call is no longer supported.

Action: Use the ALTER DATABASE RECOVER command for all recovery actions.

ORA-00283

ORA-00283   **recovery session canceled due to errors**

Cause: An error during recovery was determined to be fatal enough to end the current recovery session.

Action: More specific messages will accompany this message. Refer to the other messages for the appropriate action.

ORA-00284

ORA-00284   **recovery session still in progress**

Cause: An error during recovery was determined to be minor enough to allow the current recovery session to continue.

Action: More specific messages will accompany this message. Refer to other messages for the appropriate action.

ORA-00285

ORA-00285   **TIME not given as a string constant**

Cause: UNTIL TIME was not followed by a string constant for the time.

Action: Enter the time enclosed in single quotation marks.

ORA-00286

ORA-00286 **no members available, or no member contains valid data**

Cause: None of the members of a redo log file group are available, or the available members do not contain complete data.

Action: If a member is temporarily off line, attempt to make it available. Check that the correct filenames are being used, especially if the redo log file is being accessed from a remote location.

ORA-00287

ORA-00287   **specified change number *num* not found in thread *num***

Cause: The given change number does not appear in any of the online redo logs for the given thread.

Action: Check the statement to make certain a valid change number is given. Perhaps try to use the NEXT option for archiving logs.

ORA-00288

ORA-00288   **to continue recovery type ALTER DATABASE RECOVER CONTINUE**

Cause: During media recovery, redo information from a new log is not required but the continuation command is necessary to do a checkpoint and report errors.

Action: Type ALTER DATABASE RECOVER CONTINUE and recovery will resume.

ORA-00289

ORA-00289   **suggestion:** *filename*

Cause: This message reports the next redo log filename that is needed, according to the initialization parameters LOG_ARCHIVE_DEST and LOG_ARCHIVE_FORMAT. This message assumes that LOG_ARCHIVE_DEST and LOG_ARCHIVE_FORMAT are the same now as when the required redo log file was archived.

Action: Consider using this filename for the next log needed for recovery.

ORA-00290

ORA-00290   **operating system archiving error occurred. See error below**

Cause: While attempting to archive to a redo log file, the server encountered an unexpected operating system error.

Action: Correct the operating system error given in the messages and retry the operation. See also your operating system-specific Oracle documentation.

ORA-00291

ORA-00291 **numeric value required for PARALLEL option**

Cause: A recovery command was specified incorrectly. The PARALLEL option must be followed by a numeric argument that specifies the degree of parallelism.

Action: Re-enter the command with a numeric argument specifying the degree of parallelism desired.

ORA-00292

ORA-00292   **parallel recovery feature not installed**

Cause: A parallel recovery was requested when the parallel recovery option is not installed.

Action: Delete the PARALLEL clause from the RECOVER command. Also, delete the RECOVERY_PARALLELISM parameter in the initialization file.

ORA-00293

ORA-00293   **control file out of sync with redo log**

Cause: The redo log and control file are out of sync because a non-current control file was specified when the instance was started.

Action: Retry the Recover command using the current control file or retry the Recover command using the USING BACKUP CONTROL FILE clause.

00300-00379: Redo Log Files

This section lists messages generated when the Redo Log Files are being accessed.

ORA-00300

ORA-00300   **illegal redo log block size *num* specified - exceeds limit of *num***

Cause: The specified block size of the redo log file is greater than the maximum block size for the operating system.

Action: Create the redo log on a device with a smaller block size.

ORA-00301

ORA-00301 **error in adding log file** *name* **- file cannot be created**

Cause: The creation of the redo log file failed.

Action: Check that there is enough storage space on the device, that the name of the file is valid, and that the device is online; then try again.

Also, it is possible REUSE was specified on the command line and a file of the incorrect size exists. Either do not specify REUSE or use a file of the correct size.

ORA-00302

ORA-00302	**limit of *num* logs exceeded**
Cause:	The maximum number of redo log files has been exceeded. There is a limit, set at database creation, on the number of redo log files (typically 16).
Action:	Use the CREATE CONTROLFILE command with a larger value for MAXLOGFILES.

ORA-00304

ORA-00304   **requested INSTANCE_NUMBER is busy**

Cause: An instance tried to start by using a value of the initialization parameter INSTANCE_NUMBER that is already in use.

Action: Specify another value for INSTANCE_NUMBER in the initialization parameter file or wait for recovery to finish for that instance number. Then restart the instance.

ORA-00305

ORA-00305   **log *name* of thread *num* inconsistent; belongs to another database**

Cause: The database ID in the redo log file does not match the database ID in the control file. This redo log file is not from the current database.

Action: Specify the correct redo log file, then retry the operation.

ORA-00306

ORA-00306	limit of *num* log writer instances in this database

Cause: Starting this instance would exceed the maximum number of instances allowed for this database. This message occurs only when attempting to start another instance in Parallel Server mode.

Action: It is not possible to start more than the given number of instances. The maximum is the lower of the operating system-specific maximum or the MAXINSTANCES option specified in the CREATE DATABASE statement. See also your operating system-specific Oracle documentation.

ORA-00307

ORA-00307   **requested INSTANCE_NUMBER out of range, maximum is** *num*

Cause: The initialization parameter INSTANCE_NUMBER specified a number that was out of range.

Action: Change INSTANCE_NUMBER to a valid range and restart the instance. The minimum value is one and the maximum value is the lower of the operating system-specific maximum or the MAXINSTANCES option specified in the CREATE DATABASE statement. See also your operating system-specific Oracle documentation.

ORA-00308

ORA-00308   **cannot open archived log** *name*

Cause: The system cannot access a required archived redo log file.

Action: Check that the off line log exists, the storage device is online, and the archived file is in the correct location. Then attempt to continue recovery or restart the recovery session.

ORA-00309

ORA-00309   **log belongs to wrong database**

Cause: The system cannot access the archived redo log because it belongs to another database.

Action: Specify the correct redo log file, then retry the operation.

ORA-00310

ORA-00310 **archived log contains sequence *num*; sequence *num* required**

Cause: The archived log is out of sequence, probably because it is corrupted or the wrong redo log filename was specified during recovery.

Action: Specify the correct redo log file; then retry the operation.

ORA-00311

ORA-00311   **cannot read header from archived log**

Cause: An error occurred when attempting to read the file header from the archived redo log file.

Action: Other messages will accompany this message. See the associated messages for the appropriate action to take.

ORA-00312

ORA-00312   **online log *name* thread *num*: *str***

Cause: This message reports the filename for details of another message.

Action: Other messages will accompany this message. See the associated messages for the appropriate action to take.

ORA-00313

ORA-00313 **open failed for members of log group *name* of thread *name***

Cause: The online log cannot be opened. The file may not be in the expected location.

Action: Specify the correct redo log file or make log available, if necessary.

ORA-00314

ORA-00314 **log *name* of thread *num*, expected sequence # *num* does not match *num***

Cause: The online log is out of sequence, probably because it is corrupted or an old version.

Action: For more information on recovery procedures, see the *Oracle7 Server Administrator's Guide*

ORA-00315

ORA-00315   **log *name* of thread *num*, wrong thread # *num* in header**

Cause: The online redo log file is corrupted or is an old version.

Action: Refer to the *Oracle7 Server Administrator's Guide* for recovery procedures.

ORA-00316

ORA-00316 **log *name* of thread *num*, type *str* in header is not redo log file**

Cause: The online redo log file is corrupted or is an old version.

Action: Refer to the *Oracle7 Server Administrator's Guide* for recovery procedures.

ORA-00317

ORA-00317   **file type *str* in header is not log file**

Cause: This is not an archived redo log file.

Action: Refer to the *Oracle7 Server Administrator's Guide* for recovery procedures.

ORA-00318

ORA-00318 **log** *name* **of thread** *num***, expected file size** *num* **does not match** *num*

Cause: The file size indicated in the control file did not match the file size contained in the redo log file.

Action: Refer to the *Oracle7 Server Administrator's Guide* for recovery procedures.

ORA-00319

ORA-00319   **log *name* of thread *num* has incorrect log reset status**

Cause: An online redo log file has log reset data that is different from the log resetdata listed in the control file. The redo log file is probably an incorrectly restored backup.

Action: Refer to the *Oracle7 Server Administrator's Guide* for recovery procedures.

ORA-00320

ORA-00320 **cannot read file header from log *name* of thread *num***

Cause: The system is unable to read the redo log file header because the file is not available or the file is corrupted.

Action: Specify the correct redo log file, then retry the operation.

ORA-00321

ORA-00321   **log *name* of thread *num*, cannot update log file header**

Cause: The system cannot write to the redo log file.

Action: Restore access to the redo log file; then retry the operation.

ORA-00322

ORA-00322 **log *name* of thread *num* is not current copy**

Cause: An online redo log file appears to be an incorrectly restored backup, according to a check of the redo log file header.

Action: Refer to the *Oracle7 Server Administrator's Guide* for recovery procedures.

ORA-00323

ORA-00323   **current log of thread *num* not usable and all others need archiving**

Cause: An attempt to open a thread failed because it is necessary to switch to another online log, but all the other online redo logs need to be archived before they can be used.

Action: Archive the online redo logs for the thread, then retry the operation.

ORA-00324

ORA-00324   **log file** *name* **translated name** *name* **too long,** *num* **characters exceeds***num* **limit**

Cause: The translated name for a redo log file is too long.

Action: Respecify a redo log filename that produces a shorter filename on translation.

ORA-00325

ORA-00325   **archived log for thread *num*, wrong thread # *num* in header**

Cause: The archived redo log file is corrupted, or it belongs to another thread. This redo log file cannot be used.

Action: Find and use the correct archived redo log file, then retry the operation.

ORA-00326

ORA-00326   **log begins at change *num*, need earlier change *num***

Cause: The archived redo log file supplied for recovery was generated after the redo log file that is needed. Cannot use the log for applying redo information atthis time.

Action: Find and install the correct archived redo log file, then retry the operation.

ORA-00327

ORA-00327   **log *name* of thread *num*, physical size *num* less than needed *num***

Cause: A redo log file has shrunk in size. This is usually caused by a computer operator's mistake or an operating system error.

Action: Restore the redo log file from backup. If the database was shut down cleanly, no further action should be required. Otherwise, manual recovery may be necessary.

ORA-00328

ORA-00328   **archived log ends at change *num*, need later change *num***

Cause: The archived redo log file supplied for recovery was generated before the log that is needed. Cannot use the log for applying redo information at this time.

Action: Find and install the correct archived redo log file, then retry the operation.

ORA-00329

ORA-00329   **archived log begins at change *num*, need change *num***

Cause: The archived redo log file supplied for recovery is not the correct log. An earlier redo log file is needed.

Action: Find and install the correct archived redo log file, then retry the operation.

ORA-00330

ORA-00330   **archived log ends at change *num*, need change *num***

Cause: The archived redo log file supplied for recovery is not the correct log. A later redo log file is needed.

Action: Find and install the correct archived redo log file, then retry the operation.

ORA-00331

ORA-00331 **log version *num* incompatible with Oracle version *num***

Cause: The log was created under an older, incompatible version of Oracle, or an attempt was made to use the wrong log.

Action: Specify the correct redo log file or recover the database with the old software. Then shut down and restart Oracle with the new software.

ORA-00332

ORA-00332	**archived log is too small - may be incompletely archived**
Cause:	The log occupies less space than is allocated to it. This may result from a shutdown abort while the archiver was writing it.
Action:	Get a complete version of the file, either the online version or one that was successfully archived, and use that for recovery.

ORA-00333

ORA-00333   **redo log read error block *num* count *num***

Cause: An error occurred while reading the redo log file. Other messages will accompany this message and will give the name of the file.

Action: Restore access to the file or get another copy of the file.

ORA-00334

ORA-00334   **archived log: *name***

Cause: This message reports the filename involved with other errors.

Action: See the associated messages for a description of the problem.

ORA-00335

ORA-00335   **online log *name*: No log with this number, log does not exist**

Cause: This message reports the filename involved with other messages.

Action: Correct the underlying problems mentioned in the other messages.

ORA-00336

ORA-00336   **log file size *num* blocks is less than minimum *num* blocks**

Cause: The redo log file size specified in the CREATE DATABASE statement is too small.

Action: Increase the redo log file size and re-enter the statement.

ORA-00337

ORA-00337 **log file *name* does not exist and no size specified**

Cause: The system could not add a redo log file because it could not find an existing file or a size for creating the file.

Action: Specify a size for the redo log file, then retry the operation.

ORA-00338

ORA-00338   **log *name* of thread *num* is more recent than control file**

Cause: The control file change sequence number in the redo log file is greater thanthe number in the control file. This implies that the wrong control file is being used. Note that repeatedly causing this error to happen by reopening the database may cause the error to stop happening without correcting the problem. Every attempt to open the database advances the control file change sequence number.

Action: Use the correct control file or perform recovery using a backup of the control file.

ORA-00339

ORA-00339   **archived log does not contain any redo**

Cause: The archived log being applied is not the correct log. The log being applied is possibly a copy of a log or was an online log being prepared to be used as the current log.

Action: Restore the correct redo log file.

ORA-00340

ORA-00340   **I/O error processing online log *name* of thread *num***

Cause: An I/O error occurred on the named online redo log file, probably because the file was not accessible or because it was corrupted.

Action: Restore access to the file or restore the file from a backup.

ORA-00341

ORA-00341   **log *name* of thread *num*, wrong log # *num* in header**

Cause: The internal information in an online redo log file does not match the control file information.

Action: Refer to the *Oracle7 Server Administrator's Guide* for recovery procedures.

ORA-00342

ORA-00342   **archived log was created before last RESETLOGS**

Cause: Recovery was given a log that was created before the last ALTER DATABASE OPEN RESETLOGS command. There should be another log created since then that contains the correct redo.

Action: Find and install the correct archived redo log file, then retry the operation.

ORA-00343

ORA-00343   **too many errors, log member closed**

Cause: The maximum number of errors on this log member has been exceeded. Other messages will accompany this message.

Action: Correct the underlying problems mentioned in the other messages.

ORA-00344

ORA-00344 **unable to recreate online log** *name*

Cause: An I/O failure occurred when attempting to recreate an online log as part of either an ALTER DATABASE OPEN RESETLOGS or ALTER DATABASE CLEAR log file command.

Action: Check the accompanying messages. Check that the file is accessible.

ORA-00345

ORA-00345   **redo log write error block *num* count *num***

Cause: An I/O error has occurred while writing the log, probably because the file was not accessible, or a device failed.

Action: Make the file accessible or restore the device, then restart the system. If the log is lost, apply media or incomplete recovery.

ORA-00346

ORA-00346 **log member marked as STALE**

Cause: A redo log file member is no longer complete. Other messages will accompany this message.

Action: Correct the underlying problem mentioned in the other messages.

ORA-00347

ORA-00347   log *name* of thread *num* expected block size *num* does not match *num*

Cause: During online recovery, the block size specified in the control file did not match the block size indicated in the redo log file.

Action: Restore the correct redo log file from a backup or reset the online redo log files.

ORA-00348

ORA-00348   **single-process redo failure. Must abort instance**

Cause: A failure occurred during a single-process redo log operation. This error does not occur during normal multi-process operations.

Action: Shut down and restart the database.

ORA-00349

ORA-00349   **failure obtaining block size for** *name*

Cause: The operating system was unable to determine the block size for the named file.

Action: Check the accompanying messages, then restore the device or specify another file.

ORA-00350

ORA-00350	**log *name* of thread *num* needs to be archived**

Cause: Media recovery is enabled and the specified log was not archived.

Action: Archive the redo log file or disable media recovery. If the command supports the UNARCHIVED option, then it can be used. However, this may result in making backups unusable and forcing some offline files to be dropped.

ORA-00351

ORA-00351   **recover-to time invalid**

Cause: The time specified in a RECOVER DATABASE... UNTIL is before January 1, 1988.

Action: Specify a time after January 1, 1988.

ORA-00352

ORA-00352 **all logs for thread *num* need to be archived - cannot enable**

Cause: An attempt was made to enable a thread when all of the online redo log files in the thread needed to be archived, and media recovery has been enabled. There is no online redo log file that can be made the new current log for the thread.

Action: Archive a redo log file for the thread.

ORA-00353

ORA-00353 **log corruption near block *num* time stamp *str***

Cause: Some type of redo log file corruption has been discovered. This message describes the location of the corruption. Other messages will accompany this message and describe the type of corruption.

Action: Perform recovery with a good version of the redo log file or perform time-based recovery up to the indicated time.

ORA-00354

ORA-00354   **corrupt redo log block header**

Cause: The block header on the block in the redo log file is not valid. The block number and time-stamp are given in an accompanying message.

Action: Perform recovery with a good version of the redo log file or perform cancel-based recovery up to, but not including, the corrupted redo log file.

ORA-00355

ORA-00355   **change numbers out of order**

Cause: A change number found in the redo log file is lower than a previously encountered change number. The redo log file is corrupted in some way. The corruption may be at the earlier change number or at this one. The block number and time-stamp are given in an accompanying message.

Action: Perform recovery with a good version of the redo log file or perform time-based recovery up to the indicated time.

ORA-00356

ORA-00356   **inconsistent lengths in change description**

Cause: A change record in the redo log file contains lengths that do not add up to a consistent value. The redo log file is corrupted in some way.

Action: Perform recovery with a good version of the redo log file or perform time-based recovery up to the indicated time.

ORA-00357

ORA-00357   **too many members specified for log file, the maximum is *num***

Cause: An attempt was made to add a redo log group or add a redo log group member that would result in a set of online redo logs with too many members. The maximum number of members is set when the database is created.

Action: Use fewer redo log group members or use the CREATE CONTROLFILE command and set the parameter MAXLOGMEMBERS to a larger value.

ORA-00358

ORA-00358   **too many file members specified, the maximum is *num***

Cause: A CREATE or ALTER statement specified too many members in the file list.

Action: Specify a number of files that is within the given limit or use the CREATE CONTROLFILE command and set the parameter MAXLOGMEMBERS to a larger value.

ORA-00359

ORA-00359 **log file group *num* does not exist**

Cause: An attempt to add or drop a redo log group member specified a redo log file group number that does not exist.

Action: Check the configuration of the redo log files and retry the command with a valid group number.

ORA-00360

ORA-00360 **not a log file member:** *name*

Cause: An invalid filename was given to drop a redo log group member. Either the file is not a part of the database, or it is a datafile.

Action: Specify a valid member name and retry the command.

ORA-00361

ORA-00361   **cannot remove last log member** *name* **group** *num*

Cause: An attempt was made to remove the last member of a redo log group.

Action: If desired, delete the entire log by using the DROP log file command.

ORA-00362

ORA-00362   **member is required to form a valid log file in group** *name*

Cause: A request to drop a redo log group member was denied because it would remove data required to form a complete online redo log group.

Action: If desired, delete the entire log after archiving, if required, using the DROP log file command.

ORA-00363

ORA-00363 **log is not the archived version**

Cause: The log given for recovery is a backup of the online version from the time it was the current log. The archived version of the log would not be marked as the end of the thread. This message can also be caused by failing to list the current log of an enabled thread in a CREATE CONTROLFILE command.

Action: Find the archived version of the log and specify its name. If this is media recovery immediately following a CREATE CONTROLFILE, be sure the current log for this thread was included.

ORA-00364

ORA-00364 **cannot write header to new log member**

Cause: An I/O error occurred when attempting to write the header to a redo log group member that is being added to an existing group.

Action: See accompanying messages. Fix the problem or use another file.

ORA-00365

ORA-00365   **the specified log is not the correct next log**

Cause: The specified log failed to pass checks to ensure it corresponds to the log that was generated against a cold backup image of the database.

Action: Find the log that was generated by this copy of the database and give that filename to recover.

ORA-00369

ORA-00369   **current log of thread *num* not useable and other log being cleared**

Cause: Attempt to open thread failed because it is necessary to switch redo generation to another online log, but all the other logs are being cleared or need to be archived before they can be used.

Action: If the ALTER DATABASE CLEAR log file command is still active then wait for it to complete. Otherwise, re-enter the CLEAR command. If there are other online logs for the thread that are not being cleared, then archive the logs.

ORA-00371

ORA-00371   **no free buffer handles available**

Cause: The value of the initialization parameter DB_HANDLES is too small.

Action: Shut down the instance, increase the value of DB_HANDLES in the initialization parameter file, and then restart the instance. Under normal circumstances, let DB_HANDLES take its default value by omitting the parameter from the initialization parameter file.

ORA-00372

ORA-00372   **file *name* cannot be modified at this time**

Cause: An attempt was made to access a file being taken off line, or the database may be closing.

Action: Access the file after it is taken off line. Attempts will either succeed because the datafile is back online or fail with a message describing which file is off line. Repeat until successful.

ORA-00373

ORA-00373   **online log version *num* incompatible with Oracle version *num***

Cause: The online log was written by an incompatible version of Oracle. This can occur when the redo log file was created by either a newer or older version of Oracle.

Action: Recover the database using a compatible version, shut it down cleanly; then restart with the current software.

ORA-00374

ORA-00374   **parameter DB_BLOCK_SIZE =** *num* **invalid, valid range** *num..num*

Cause: An invalid value has been used for the DB_BLOCK_SIZEinitialization parameter.

Action: Adjust the parameter and restart the instance.

ORA-00375

ORA-00375   **unable to get default DB_BLOCK_SIZE**

Cause: The system was unable to determine the default DB_BLOCK_SIZE.

Action: See the corresponding operating system-specific message. As a work-around, specify the block size with the parameter DB_BLOCK_SIZE in the initialization parameter file. See also your operating system-specific Oracle documentation.

ORA-00376

ORA-00376   **file *name* cannot be read at this time**

Cause: An attempt was made to read from a file that is not readable. The most likely cause is that the file is off line.

Action: Check the state of the file. Bring the file online, if necessary.

ORA-00390

ORA-00390   log *name* of thread *num* is being cleared, cannot become current log

Cause: An attempt to switch to a new online log for the redo thread failed because no reusable log could be found. This log is being cleared and will be useable when the clearing completes. The command that began the clearing may have terminated without completing the clearing.

Action: If the clear command is still executing, then wait for its completion. If it terminated, then re-enter the clear command or drop the log.

ORA-00391

ORA-00391   **all threads must switch to new log format at the same time**

Cause: An attempt to switch the current log of a single thread is not allowed because the compatibility requirements force a new log format version number.When changing log formats, all threads must switch to the new format at the same time.

Action: Open the database to cause the coordinated log switch. If that is not possible, then return to the same software version and compatibility setting last used to open the database.

ORA-00392

ORA-00392 **log** *name* **of thread** *num* **is being cleared, operation not allowed**

Cause: An operation encountered this online log in the middle of being cleared. The command that began the clearing may have terminated without completing the clearing.

Action: If the clear command is still executing, then wait for its completion. If it terminated, then re-enter the clear command or drop the log.

ORA-00393

ORA-00393   **log *name* of thread *num* is needed for recovery of offline datafiles**

Cause: Log cannot be cleared because the redo in it is needed to recover offline datafiles. It has not been archived so there is no other copy available. If the log is cleared, the tablespaces containing the files will have to be dropped.

Action: Archive the log, then repeat the clear command. If archiving is not possible and dropping the tablespaces is acceptable, then add the clause UNRECOVERABLE DATAFILE at the end of the clear command.

00400-00420: Oracle Compatibility

This section lists messages generated when different versions of Oracle communicate.

ORA-00400

ORA-00400   **invalid release value *num* for parameter *name***

Cause: The release level given for the specified initialization parameter is invalid.

Action: Correct the parameter value in the initialization parameter file and retry.

ORA-00401

ORA-00401 **the value for parameter *name* is not supported by this release**

Cause: The value specified cannot be supported by this release of the software.

Action: Choose an appropriate value or remove the initialization parameter value to use the default value.

ORA-00402

ORA-00402   **database changes by release *num* cannot be used by release *num***

Cause: Changes have been made to the database that require a newer software release or that violate the open compatibility initialization parameters.

Action: Use a version of the software that can understand the changes or relax the compatibility requirements in the initialization parameter file.

ORA-00403

ORA-00403 *str str* **is not the same as other instances** *str*

Cause: The recovery compatible, or recovery not compatible, initialization parameters were changed for another running instance.

Action: Change the initialization parameters of the current instance to match other instances already running.

ORA-00404

ORA-00404   **conversion data file not found:** *name*

Cause: The file used for converting the database from Oracle Version 6 to Oracle7 could not be found.

Action: Check that the conversion process has been started on this database and that the datafile name is accessible.

ORA-00405

ORA-00405 **compatibility type** *name*

Cause: Reporting a type associated with another message.

Action: See accompanying message.

ORA-00406

ORA-00406 **COMPATIBLE parameter needs to be *num* or greater**

Cause: The specification for the initialization parameter COMPATIBLE is too low to allow the requested operation.

Action: Shut down and specify a higher COMPATIBLE value, then restart.

ORA-00407

ORA-00407   **rolling upgrade from release *num.num* to *num.num* is not allowed**

Cause: Another instance has the database mounted and that instance is for a different release of Oracle than you are using.

Action: Shut down and start up all instances with the new release of Oracle.

00436-00437: Licensing

This section lists messages generated when the installed Oracle product discovers a discrepancy in licensing codes.

ORA-00436

ORA-00436   **Oracle is not licensed. Contact Oracle Corporation for assistance**

Cause: This installed Oracle software is not licensed to run on this CPU. This can also occur if Oracle software has been installed incorrectly, for example, with the wrong licensing codes.

Action: Check that Oracle is installed correctly. Then contact customer support.

ORA-00437

ORA-00437 **Oracle feature is not licensed. Contact Oracle Corporation for assistance**

Cause: This installed Oracle feature is not licensed to run on this CPU. This can also occur if Oracle software has been installed incorrectly, for example, with the wrong licensing codes.

Action: Check that Oracle is installed correctly. Then contact customer support.

00440-00485: Background Processes

This section lists messages generated when background processes of the Oracle Server are started or shut down.

ORA-00443

ORA-00443 **background process *name* did not start**

Cause: The specified process did not start.

Action: Check that the executable image is in the correct place with the correct protections and that there is enough memory.

ORA-00444

ORA-00444   **background process *name* failed while starting**

Cause: This message is usually caused by a faulty or non-existent background process image.

Action: Check the accompanying messages, if any, and the background process trace file. Correct the problem mentioned in the other messages or obtain a proper background process image, then retry the operation.

ORA-00445

ORA-00445 **background process *name* did not start after *num* seconds**

Cause: The specified process did not start.

Action: Check and, if necessary, correct problems indicated by one or more of the following:
- the size of the SGA
- the operating system-specific initialization parameters
- accompanying messages
- the background trace file
- the executable image is not in the right location with the correct protections

See also your operating system-specific Oracle documentation.

ORA-00446

ORA-00446   **background process started when not expected**

Cause: A background process started after Oracle was already running.

Action: Check the accompanying messages, if any, and the background process trace file. Correct the problem mentioned in the other messages. If no user on site started the process, report the message to customer support.

ORA-00447

ORA-00447   **fatal error in background process**

Cause: One of the background processes completed unexpectedly.

Action: Restart the system. Check and, if necessary, correct the problem indicated by the background trace file in BACKGROUND_DUMP_DEST.

ORA-00448

ORA-00448   **normal completion of background process**

Cause: One of the background processes completed normally as requested by the user.

Action: If you are solving a problem, check for other messages and the background process trace file. Correct the problem mentioned in the other messages. Then shut down and restart Oracle.

ORA-00449

ORA-00449 **background process *name* unexpectedly terminated with error *num***

Cause: A foreground process needing service from a background process has discovered the background process died.

Action: Refer to the message code given in the message and the trace file for the foreground and the background processes.

ORA-00470

ORA-00470   **LGWR process terminated with error**

Cause: The log writer process terminated abnormally.

Action: Check the accompanying messages, if any, and the background process trace file. Correct the problem mentioned in the other messages. Then shut down and restart the instance. If the trace file mentions any other backgroundprocess errors, check the trace file for the mentioned process until the root error is found.

ORA-00471

ORA-00471   **DBWR process terminated with error**

Cause: The Database Writer process terminated abnormally.

Action: Check the accompanying messages, if any, and the background process trace file. Correct the problem mentioned in the other messages. Then shut down and restart the instance. If the trace file mentions any other background process messages, check the trace file for the mentioned process until the root message is found.

ORA-00472

ORA-00472   **PMON process terminated with error**

Cause: The Process Monitor process terminated abnormally.

Action: Check the accompanying messages, if any, and the background process trace file. Correct the problem mentioned in the other messages. Then shut down and restart the instance. If the trace file mentions any other background process messages, check the trace file for the mentioned process until the root message is found.

ORA-00473

ORA-00473   **ARCH process terminated with error**

Cause: The Archiver process terminated abnormally.

Action: Check the accompanying messages, if any, and the background process trace file. Correct the problem mentioned in the other messages. Then shut down and restart the instance. If the trace file mentions any other background process messages, check the trace file for the mentioned process until the root message is found.

ORA-00474

ORA-00474 **SMON process terminated with error**

Cause: The System Monitor process terminated abnormally.

Action: Check the accompanying messages, if any, and the background process trace file. Correct the problem mentioned in the other messages. Then shut down and restart the instance. If the trace file mentions any other background process messages, check the trace file for the mentioned process until the root message is found.

ORA-00475

ORA-00475   **TRWR process terminated with error.**

Cause: The system tracing process died.

Action: Restart the instance.

ORA-00476

ORA-00476   RECO process terminated with error

Cause: The distributed transaction, two-phase commit, recovery process died.

Action: Restart the instance.

ORA-00477

ORA-00477   **SNP* process terminated with an error**

Cause: A snapshot refresh process died.

Action: Restart the instance.

ORA-00480

ORA-00480   **LCK* process terminated with error**

Cause: One Lock process terminated abnormally.

Action: Check the accompanying messages, if any, and the background process trace file. Correct the problem mentioned in the other messages. Then shut down and restart the instance. If the trace file mentions any other background process messages, check the trace file for the mentioned process until the root message is found.

ORA-00483

ORA-00483   **during shutdown a process abnormally terminated**

Cause: One of the background processes did not exit normally when the instance shut down.

Action: Use the SHUTDOWN ABORT command. Check the accompanying messages, if any, and the background process trace file. Correct the problem mentioned in the other messages. Then restart the instance. If the trace file mentions any other background process messages, check the trace file for the mentioned process until the root message is found.

00486-00569: Interrupt Handlers

ORA-00568

ORA-00568   **maximum number of interrupt handlers exceeded**

Cause: The number of registered interrupt handling routines for when the break key is entered exceeds the maximum allowed.

Action: Reduce the number of registered interrupt handlers.

00570-00599: SQL*Connect Opening and Reading Files

See "Related Publications" in the Preface, which indicates the Oracle manual containing the messages for this range.

00600-00639: Oracle Exceptions

This section lists messages generated when an internal exception is generated within Oracle.

ORA-00600

ORA-00600 **internal error code, arguments: [*num*], [*?*], [*?*], [*?*], [*?*], [*?*]**

Cause: This is a catchall internal error message for Oracle program exceptions. It indicates that a process has met a low-level, unexpected condition.

Various causes of this message include:

- time-outs
- file corruption
- failed data checks in memory
- hardware, memory, or I/O errors
- incorrectly restored files

The first argument is the internal message number. Other arguments are various numbers, names, and character strings. The numbers may change meanings between different versions of the Oracle Server.

Action: Report this error to customer support after gathering the following information:

- events that led up to the error
- the operations that were attempted that led to the error
- the conditions of the operating system and database at the time ofthe error
- any unusual circumstances that occurred before receiving theORA-00600 message
- contents of any trace files generated by the error
- the relevant portions of the Alert file

Note: The cause of this message may manifest itself as different errors at different times. Be aware of the history of errors that occurred before this internal error.

ORA-00601

ORA-00601   **cleanup lock conflict**

Cause: The Process Monitor process encountered a lock conflict while trying to recover processes. This is an internal error message not usually issued.

Action: Contact customer support.

ORA-00602

ORA-00602 **internal programming exception**

Cause: An internal programming exception has occurred.

Action: Report this error as a program bug to customer support.

ORA-00603

ORA-00603   **Oracle Server session terminated by fatal error**

Cause: An Oracle Server session is in an unrecoverable state.

Action: Log in to Oracle again so a new server session will be created automatically. Examine the session trace file for more information.

ORA-00604

ORA-00604   **error occurred at recursive SQL level** *num*

Cause: An error occurred while processing a recursive SQL statement. A recursive SQL statement is one that applies to internal dictionary tables.

Action: If the situation described in the next message on the stack can be corrected, do so; otherwise, contact customer support.

ORA-00606

ORA-00606   internal error code

Cause: A call to deferred UPI functions was made in non-deferred mode.

Action: Contact customer support.

00640-00699: SQL*Connect

00700-00709: Dictionary Cache

This section lists messages generated when the Oracle dictionary cache is accessed. Note that *row cache* is a synonym for *dictionary cache*.

ORA-00701

ORA-00701 **object necessary for warm starting database cannot be altered**

Cause: An attempt was made to alter or drop a table, cluster, or index defined in the control file's bootstrap segment and needed to warm start the database.

Action: Correct the spelling of the object name or remove it from the ALTER or DROP statement.

ORA-00702

ORA-00702 **bootstrap version *version* inconsistent with version *version***

Cause: The version of the bootstrap data in a file is incompatible with the current version of the software.

Action: Restore a version of the software that is compatible with the datafiles.

ORA-00703

ORA-00703 **maximum number of dictionary cache instance locks exceeded**

Cause: An operation requested a resource that was unavailable. The maximum number of dictionary cache instance locks is specified by the ROW_CACHE_INSTANCE_LOCKS parameter in the initialization parameter file. When this maximum is reached, no more requests are processed.

Action: Try the operation again in a few minutes. If this error occurs often, contact customer support.

ORA-00704

ORA-00704   **bootstrap process failure**

Cause: An error occurred when processing bootstrap data. Refer to the accompanying messages for more information about the cause of the problem.

Action: Correct the problems mentioned in the other messages. If the problem persists, contact customer support.

00816-00816: Message Translation

ORA-00816

ORA-00816   **error message translation failed**

Cause: There is an internal error where a routine was unable to translate a message code.

Action: Contact customer support.

00900-00999: SQL Parsing

This section lists some of the messages generated when SQL statements are parsed by the Oracle Server. Most, but not all, messages in this section indicate incorrect SQL syntax. For SQL syntax, refer to *Oracle7 Server SQL Reference.*

ORA-00900

ORA-00900	invalid SQL statement

Cause: The statement is not recognized as a valid SQL statement.

This error can occur if the Procedural Option is not installed and a SQL statement is issued that requires this option; for example, a CREATE PROCEDURE statement. You can determine if the Procedural Option is installed by starting SQL*Plus. If the PL/SQL banner is not displayed, then the option is not installed.

Action: Correct the syntax or install the Procedural Option.

ORA-00901

ORA-00901   invalid CREATE command

Cause: The CREATE command was not followed by a valid CREATE option.

Action: Correct the syntax.

ORA-00902

ORA-00902 invalid datatype

Cause: The datatype entered in the CREATE or ALTER TABLE statement is not valid.

Action: Correct the syntax.

ORA-00903

ORA-00903   **invalid table name**

Cause: A table or cluster name is invalid or does not exist. This message is also issued if an invalid cluster name or no cluster name is specified in an ALTER CLUSTER or DROP CLUSTER statement.

Action: Check spelling. A valid table name or cluster name must begin with a letter and may contain only alphanumeric characters and the special characters $, _, and #. The name must be less than or equal to 30 characters and cannot be a reserved word.

ORA-00904

ORA-00904	**invalid column name**

Cause: The column name entered is either missing or invalid.

Action: Enter a valid column name. A valid column name must begin with a letter, be less than or equal to 30 characters, and consist of only alphanumeric characters and the special characters $, _, and #. If it contains other characters, it must be enclosed in double quotation marks. It may not be a reserved word.

ORA-00905

ORA-00905   **missing keyword**

Cause: A required keyword is missing.

Action: Correct the syntax.

ORA-00906

ORA-00906	**missing left parenthesis**

Cause: A required left parenthesis has been omitted. Certain commands, such as CREATE TABLE, CREATE CLUSTER, and INSERT, require a list of items enclosed in parentheses. Parentheses also are required around subqueries in WHERE clauses and in UPDATE *table* SET *column* = (SELECT...) statements.

Action: Correct the syntax, inserting a left parenthesis where required, and retry the statement.

ORA-00907

ORA-00907   **missing right parenthesis**

Cause: A left parenthesis has been entered without a closing right parenthesis, or extra information was contained in the parentheses. All parentheses must be entered in pairs.

Action: Correct the syntax and retry the statement.

ORA-00908

ORA-00908   **missing NULL keyword**

Cause: Either of the following:

In a CREATE TABLE or ALTER TABLE statement, NOT was entered to specify that no null values are allowed in that column, but the keyword NULL was omitted.

In the IS [NOT] NULL logical operator, the keyword NULL wasnot found.

For example, the following statement generates this message:

```
SELECT * FROM EMP WHERE DEPTNO IS NOT;
```

The keyword NULL must follow the keywords IS NOT.

Action: Correct the syntax.

ORA-00909

ORA-00909 **invalid number of arguments**

Cause: An Oracle function was referenced with an incorrect number of arguments.All Oracle functions, except for SYSDATE, require at least one argument.

Action: Correct the syntax of the function by entering the required numberof arguments.

ORA-00910

ORA-00910   **specified length too long for its datatype**

Cause: No size was specified for a character field or the size was invalid. A maximum length must be specified for each character column. The maximum value for this length varies for each character datatype, such as CHAR or VARCHAR2.

Action: Enter a maximum length for the field.

ORA-00911

ORA-00911   invalid character

Cause: Special characters are valid only in certain places. If special characters other than $, _, and # are used in a name and the name is not enclosed in double quotation marks ("), this message will be issued. One exception to this rule is for database names; in this case, double quotes are stripped out and ignored.

Action: Remove the invalid character from the statement or enclose the object name in double quotation marks.

ORA-00913

ORA-00913   too many values

Cause: The SQL statement requires two sets of values equal in number. This error occurs when the second set contains more items than the first set. For example, the subquery in a WHERE or HAVING clause may return too many columns, or a VALUES or SELECT clause may return more columns than are listed inthe INSERT.

Action: Check the number of items in each set and change the SQL statement to make them equal.

ORA-00914

ORA-00914   missing ADD keyword

Cause: The keyword ADD does not precede one of the following:

- the keyword log file in an ALTER DATABASE statement
- a column element or table constraint in an ALTER TABLE statement
- the keyword DATAFILE in an ALTER TABLESPACE statement

Action: Specify the keyword ADD in the ALTER statement.

ORA-00915

ORA-00915   **network access of dictionary table not currently allowed**

Cause: This is an internal error message not usually issued.

Action: Contact customer support.

ORA-00917

ORA-00917   missing comma

Cause: A required comma has been omitted from a list of columns or values in an INSERT statement or a list of the form ((C,D),(E,F), ...).

Action: Correct the syntax.

ORA-00918

ORA-00918   **column ambiguously defined**

Cause: A column name used in a join exists in more than one table and is thus referenced ambiguously. In a join, any column name that occurs in more than one of the tables must be prefixed by its table name when referenced. The column should be referenced as TABLE.COLUMN or TABLE_ALIAS.COLUMN. For example, if tables EMP and DEPT are being joined and both contain the column DEPTNO, then all references to DEPTNO should be prefixed with the table name, as in EMP.DEPTNO or E.DEPTNO.

Action: Prefix references to column names that exist in multiple tables with either the table name or a table alias and a period (.), as in the examples above.

ORA-00919

ORA-00919   invalid function

Cause: An entry was formatted like a function call, but it is not recognizable as an Oracle function.

Action: Correct the syntax.

ORA-00920

ORA-00920	**invalid relational operator**

Cause: A search condition was entered with an invalid or missing relational operator.

Action: Include a valid relational operator such as =, !=, ^=, <>, >, <, >=, <=, ALL, ANY, [NOT] BETWEEN, EXISTS, [NOT] IN, IS [NOT] NULL, or [NOT] LIKE in the condition.

ORA-00921

ORA-00921   unexpected end of SQL command

Cause: The SQL command was not complete. Part of a valid command was entered, but at least one major component was omitted.

Action: Correct the syntax.

ORA-00922

ORA-00922   **missing or invalid option**

Cause: An invalid option was specified in defining a column or storage clause. The valid option in specifying a column is NOT NULL to specify that the column cannot contain any NULL values. Only constraints may follow the datatype. Specifying a maximum length on a DATE or LONG datatype also causes this error.

Action: Correct the syntax. Remove the erroneous option or length specification from the column or storage specification.

ORA-00923

ORA-00923   **FROM keyword not found where expected**

Cause: In a SELECT or REVOKE statement, the keyword FROM was either missing, misplaced, or misspelled. The keyword FROM must follow the last selected item in a SELECT statement or the privileges in a REVOKE statement.

Action: Correct the syntax. Insert the keyword FROM where appropriate. The SELECT list itself also may be in error. If quotation marks were used in an alias, check that double quotation marks enclose the alias. Also, check to see if a reserved word was used as an alias.

ORA-00924

ORA-00924   missing BY keyword

Cause: The keyword BY was omitted in a GROUP BY, ORDER BY, or CONNECT BY clause. In a GRANT statement, the keyword IDENTIFIED must also be followed by the keyword BY.

Action: Correct the syntax. Insert the keyword BY where required and then retry the statement.

ORA-00925

ORA-00925   missing INTO keyword

Cause: An INSERT statement has been entered without the keyword INTO.

Action: Correct the syntax. Insert the keyword INTO where required and then retry the statement.

ORA-00926

ORA-00926   **missing VALUES keyword**

Cause: An INSERT statement has been entered without the keyword VALUES or SELECT. Either a VALUES clause or a SELECT subquery must follow the INSERT INTO clause.

Action: Correct the syntax. Enter either a VALUES clause or a subquery after the INSERT INTO clause.

ORA-00927

ORA-00927   missing equal sign

Cause: An equal sign has been omitted in one of the following places:
- in the SET clause of an UPDATE statement
- following "!" in a search condition to signify not equal

Action: Correct the syntax. Insert the equal sign where required and retry the statement.

ORA-00928

ORA-00928   missing SELECT keyword

Cause: A SELECT subquery must be included in a CREATE VIEW statement.

Action: Correct the syntax. Insert the required SELECT clause after the CREATE VIEW clause and then retry the statement.

ORA-00929

ORA-00929   missing period

Cause: This is an internal error message not usually issued.

Action: Contact customer support.

ORA-00930

ORA-00930   **missing asterisk**

Cause: This is an internal error message not usually issued.

Action: Contact customer support.

ORA-00931

ORA-00931   missing identifier

Cause: This is an internal error message not usually issued.

Action: Contact customer support.

ORA-00932

ORA-00932   inconsistent datatypes

Cause: Either

- An attempt was made to perform an operation on incompatible datatypes. For example, adding a character field to a date field (dates may only be added to numeric fields) or concatenating a character field with a long field.
- An attempt was made to perform an operation on a database object (such as a table or view) that is not intended for normal use. For example, system tables cannot be modified by a user. Note that on rare occasions this error occurs because a misspelled object name matched a restricted object's name.
- An attempt was made to use an undocumented view.

Action: If the cause is

- different datatypes, then use consistent datatypes. For example, convert the character field to a numeric field with the TO_NUMBER function before adding it to the date field. Functions may not be used withlong fields.
- an object not intended for normal use, then do not access the restricted object.

ORA-00933

ORA-00933   SQL command not properly ended

Cause: The SQL statement ends with an inappropriate clause. For example, an ORDER BY clause may have been included in a CREATE VIEW or INSERT statement. ORDER BY may not be used to create an ordered view or to insertin a certain order. Also, an improper SQL ending occurs if IN clause is used with only one argument (IN(X), for example). An IN clause must have two or more arguments.

Action: Correct the syntax by removing the inappropriate clauses. It may be possible to duplicate the removed clause with another SQL statement. For example, to order the rows of a view, do so when querying the view and not when creating it. This error can also occur in SQL*Forms applications if a continuation line is indented. Check for indented lines and delete these spaces.

ORA-00934

ORA-00934 **group function is not allowed here**

Cause: One of the group functions, such as AVG, COUNT, MAX, MIN, SUM, STDDEV, or VARIANCE, was used in a WHERE or GROUP BY clause.

Action: Remove the group function from the WHERE or GROUP BY clause.

The desired result may be achieved by including the function in a subquery or HAVING clause.

ORA-00935

ORA-00935   **group function is nested too deeply**

Cause: This is an internal error message not usually issued.

Action: Contact customer support.

ORA-00936

ORA-00936   **missing expression**

Cause: A required part of a clause or expression has been omitted. For example, a SELECT statement may have been entered without a list of columns or expressions or with an incomplete expression. This message is also issued in cases where a reserved word is misused, as in SELECT TABLE.

Action: Check the statement syntax and specify the missing component.

ORA-00937

ORA-00937   **not a single-group group function**

Cause: A SELECT list cannot include both a group function, such as AVG, COUNT, MAX, MIN, SUM, STDDEV, or VARIANCE, and an individual column expression, unless the individual column expression is included in a GROUP BY clause.

Action: Drop either the group function or the individual column expression from the SELECT list or add a GROUP BY clause that includes all individual column expressions listed.

ORA-00938

ORA-00938   **not enough arguments for function**

Cause: The function was referenced with too few arguments.

Action: Check the function syntax and specify the required number of arguments.

ORA-00939

ORA-00939   **too many arguments for function**

Cause: The function was referenced with too many arguments.

Action: Check the function syntax and specify only the required number of arguments.

ORA-00940

ORA-00940   invalid ALTER command

Cause: An invalid ALTER option was specified.

Action: Correct the syntax.

ORA-00941

ORA-00941   **missing cluster name**

Cause: The cluster name was either missing or invalid.

Action: Specify a valid cluster name. A valid cluster name must start with a letter, be less than or equal to 30 characters, and contain only alphanumeric characters or the special characters $, _, and #. It may not be a reserved word. The name must be specified immediately following the keywords CREATE CLUSTER.

ORA-00942

ORA-00942   table or view does not exist

Cause: The table or view entered does not exist, a synonym that is not allowed here was used, or a view was referenced where a table is required. Existing user tables and views can be listed by querying the data dictionary. Certain privileges may be required to access the table. If an application returned this message, the table the application tried to access does not exist in the database, or the application does not have access to it.

For Trusted Oracle7 configured in DBMS MAC mode, you may get this message if you attempt to access an object that you do not dominate.

Action: Check each of the following:
- the spelling of the table or view name.
- that a view is not specified where a table is required.
- that an existing table or view name exists. Contact the database administrator if the table needs to be created or if user or application privileges are required to access the table.

Also, if attempting to access a table or view in another schema, make certain the correct schema is referenced and that access to the object is granted.

For Trusted Oracle7 in DMBS MAC mode, if the cause is an object existing at a higher label, alter your current label to dominate the creation label of the object. Then retry the command.

ORA-00943

ORA-00943   cluster does not exist

Cause: The current user owns no cluster by the specified name.

For Trusted Oracle7 configured in DBMS MAC mode, you may get this message if you attempt to access an object that you do not dominate.

Action: Specify a valid cluster name following the keyword CLUSTER, then retry the statement.

For Trusted Oracle7 in DMBS MAC mode, if the cause is an object existing at a higher label, alter your current label to dominate the creation label of the object. Then retry the command.

ORA-00944

ORA-00944   **insufficient number of clustered columns**

Cause: An attempt was made to create a table with fewer cluster columns than were specified in the CREATE CLUSTER statement. The CLUSTER clause of a CREATE TABLE statement must specify all cluster columns that were defined when the cluster was created.

Action: Specify all cluster columns in the CREATE TABLE statement, then retry it.

ORA-00945

ORA-00945   **specified clustered column does not exist**

Cause: A column specified in the cluster clause of a CREATE TABLE statement is not a column in this table.

For Trusted Oracle7 configured in DBMS MAC mode, you may get this message if you attempt to access an object that you do not dominate.

Action: Re-execute the statement, using the names of columns defined for the table.

For Trusted Oracle7 in DMBS MAC mode, if the cause is an object existing at a higher label, alter your current label to dominate the creation label of the object. Then retry the command.

ORA-00946

ORA-00946   missing TO keyword

Cause: A GRANT statement was specified without the keyword TO, or an invalid form of the GRANT command was entered.

Action: Check the syntax for the GRANT command, insert the keyword TO where required, and retry the statement.

ORA-00947

ORA-00947   not enough values

Cause: This error occurs when a SQL statement requires two sets of values equal in number, but the second set contains fewer items than the first set. This can occur in a WHERE or HAVING clause in which a nested SELECT returns too few columns as in:

```
WHERE (A,B) IN (SELECT C FROM ...)
```

Another common cause of this error is an INSERT statement in which the VALUES or SELECT clause does not contain enough values needed for the INSERT, as in

```
INSERT INTO EMP(EMPNO,ENAME) VALUES('JONES')
```

Action: Check the number of items in each set and change the SQL statement to make them equal.

ORA-00948

ORA-00948   ALTER CLUSTER statement no longer supported

Cause: The ALTER CLUSTER statement has been withdrawn.

Action: To add data to a cluster from an existing table, use the following series of SQL statements:

```
CREATE TABLE newtable SELECT * FROM oldtable CLUSTER clustername;
DROP oldtable;
RENAME TABLE newtable oldtable;
```

ORA-00949

ORA-00949   **illegal reference to remote database**

Cause: This is an internal error message not usually issued.

Action: Contact customer support.

ORA-00950

ORA-00950   **invalid DROP option**

Cause: A DROP command was not followed by a valid DROP option, such as CLUSTER, DATABASE LINK, INDEX, ROLLBACK SEGMENT, SEQUENCE, SYNONYM, TABLE, TABLESPACE, or VIEW.

Action: Check the command syntax, specify a valid DROP option, and then retry the statement.

ORA-00951

ORA-00951   **cluster not empty**

Cause: A DROP CLUSTER statement specified a cluster that is not empty. A cluster may not be dropped if it contains any tables, unless the optional INCLUDING TABLES clause is specified. Tables may also be removed from a cluster by using the DROP TABLE command.

Action: Either specify the INCLUDING TABLES clause in the DROP CLUSTER statement or remove all tables from the cluster with the DROP TABLE command before issuing the DROP CLUSTER command.

ORA-00952

ORA-00952   missing GROUP keyword

Cause: Groups are not currently implemented.

Action: No user action required.

ORA-00953

ORA-00953 **invalid index name**

Cause: In a CREATE INDEX, DROP INDEX, or VALIDATE INDEX statement, the index name was missing or invalid.

Action: Specify a valid index name after the keyword INDEX. To drop or validate an existing index, check the name by querying the data dictionary. To create a new index, check the syntax before retrying.

ORA-00954

ORA-00954   missing IDENTIFIED keyword

Cause: A GRANT CONNECT statement was issued without the keyword IDENTIFIED.

Action: Check the syntax and insert the keyword IDENTIFIED after the last username. The format is

```
GRANT CONNECT TO user-list IDENTIFIED BY password-list;
```

ORA-00955

ORA-00955 *name* **is already used by an existing object**

Cause: An attempt was made to create a database object (such as a table, view, cluster, index, or synonym) that already exists. A user's database objects must have distinct names.

Action: Enter a unique name for the database object or modify or drop the existing object so it can be reused.

ORA-00956

ORA-00956   **missing or invalid auditing option**

Cause: An AUDIT or NOAUDIT command was not followed by a valid option or the keyword ALL. For example, when AUDITing tables an option such as ALTER, AUDIT, COMMENT, DELETE, GRANT, INDEX, INSERT, LOCK, RENAME, SELECT, or UPDATE must be specified.

Action: Correct the syntax.

ORA-00957

ORA-00957   **duplicate column name**

Cause: A column name was specified twice in a CREATE or INSERT statement. Column names must be unique within a table, view, or cluster.

Action: In a CREATE statement, change one of the column names to a new, unique column name. In an INSERT statement, remove one of the duplicate names.

ORA-00958

ORA-00958   missing CHECK keyword

Cause: The keyword CHECK should follow the keyword WITH in the WITH OPTION clause of the CREATE VIEW statement.

Action: Check the statement syntax and insert the keyword CHECK where required. Then retry the statement.

ORA-00959

ORA-00959   tablespace **"*name*" does not exist**

Cause: A statement specified the name of a tablespace that does not exist.

For Trusted Oracle7 configured in DBMS MAC mode, you may get this message if you attempt to access an object that you do not dominate.

Action: Enter the name of an existing tablespace. For a list of tablespace names, query the data dictionary. If a tablespace is dropped and re-created with the same name, use ALTER USER to reset the default or temporary tablespace name, because the new tablespace is not the same as the dropped tablespace, even though they have the same name.

For Trusted Oracle7 in DMBS MAC mode, if the cause is an object existing at a higher label, alter your current label to dominate the creation label of the object. Then retry the command.

ORA-00960

ORA-00960   ambiguous column naming in select list

Cause: A column name in the order-by list matches more than one select list column.

Action: Remove the duplicate column naming in the select list.

ORA-00962

ORA-00962   too many group-by or order-by expressions

Cause: The group-by or order-by column list contains more than 255 expressions.

Action: Use no more than 255 expressions in the group-by or order-by list.

ORA-00964

ORA-00964   **table name not in FROM list**

Cause: A table specified in a query's SELECT list is not named in the FROM clause list.

Action: Check spelling of the table names, check that each table name in the SELECT list matches a table name in the FROM list, and then retry the statement.

ORA-00965

ORA-00965   **column aliases not allowed for "*"**

Cause: An alias was used with the return-all-columns function (*) in the SELECT list. For example:

<code>SELECT * COL_ALIAS FROM EMP;</code>

Action: Either specify individual columns or do not specify an alias with a "*".

ORA-00966

ORA-00966	**missing TABLE keyword**

Cause: A LOCK statement was specified and the keyword TABLE was missing, misspelled, or misplaced. A LOCK statement must begin withLOCK TABLE *tablename*.

Action: Correct the syntax.

ORA-00967

ORA-00967   **missing WHERE keyword**

Cause: The keyword WHERE in a SELECT statement was missing, misspelled, or misplaced.

Action: Correct the syntax.

ORA-00968

ORA-00968   missing INDEX keyword

Cause: The keyword INDEX in a CREATE UNIQUE INDEX or VALIDATE INDEX statement was missing, misspelled, or misplaced.

Action: Correct the syntax.

ORA-00969

ORA-00969   **missing ON keyword**

Cause: The keyword ON in a GRANT, REVOKE, or CREATE INDEX statement was missing, misspelled, or misplaced.

Action: Check syntax and spelling, and use the keyword ON where required.

ORA-00970

ORA-00970	**missing WITH keyword**

Cause: The keyword START was specified without the keyword WITH.Both keywords are necessary if a START WITH clause is desired in a tree-structured query.

Action: Change the keyword START to the keywords START WITH. Then retry the statement.

ORA-00971

ORA-00971   missing SET keyword

Cause: The keyword SET in an UPDATE statement is missing, misspelled,or misplaced.

Action: Check syntax and spelling, and use the keyword SET after the name of the table to be updated.

ORA-00972

ORA-00972   identifier is too long

Cause: The name of a schema object exceeds 30 characters. Schema objects are tables, clusters, views, indexes, synonyms, tablespaces, and usernames.

Action: Shorten the name to 30 characters or less.

ORA-00974

ORA-00974   **invalid PCTFREE value percentage**

Cause: The percentage of free space specified in a CREATE INDEX statement is not between 0 and 100. A PCTFREE value of 0 means the entire block is available. The value 100 is not useful because it means that no data may be inserted. The default is 10.

Action: Specify a PCTFREE value between 0 and 100. Then retry the statement.

ORA-00975

ORA-00975   date + date not allowed

Cause: An attempt was made to add two date fields together. Dates may be added only to numeric fields, not to other dates.

Action: Use the Oracle function TO_NUMBER to convert one of the date fields to a numeric field before adding it to the other date field.

ORA-00976

ORA-00976 LEVEL, PRIOR, or ROWNUM not allowed here

Cause: The use of the PRIOR clause, the pseudo-column LEVEL, or ROWNUM is incorrect in this context.

Action: Check the syntax for the SQL statement. Then remove or relocate the keyword PRIOR, LEVEL, or ROWNUM.

ORA-00977

ORA-00977   duplicate auditing option

Cause: An AUDIT or NOAUDIT statement specified the same option more than once.

Action: Either specify ALL without other auditing options or remove the duplicate auditing specifications.

ORA-00978

ORA-00978   **nested group function without GROUP BY**

Cause: A group function, such as AVG, COUNT, MAX, MIN, SUM, STDDEV, or VARIANCE, was used within another group function, as in MAX(COUNT(*)), without a corresponding GROUP BY clause.

Action: Either add a GROUP BY clause or remove the extra level of nesting.

ORA-00979

ORA-00979   **not a GROUP BY expression**

Cause: The GROUP BY clause does not contain all the expressions in the SELECT clause. SELECT expressions that are not included in a group function, such as AVG, COUNT, MAX, MIN, SUM, STDDEV, or VARIANCE, must be listed in the GROUP BY clause.

Action: Include in the GROUP BY clause all SELECT expressions that are not group function arguments.

ORA-00980

ORA-00980	**synonym translation is no longer valid**
Cause:	The synonym used is based on a table, view, or synonym that no longer exists.
Action:	Replace the synonym with the name of the object it references or re-create the synonym so that it refers to a valid table, view, or synonym.

ORA-00981

ORA-00981   cannot mix table and system auditing options

Cause: Both table-wide and system-wide options were specified within a single AUDIT statement.

Action: Check the AUDIT command syntax. Then retry one or more AUDIT statements.

ORA-00982

ORA-00982   missing plus sign

Cause: A left parenthesis appeared in a join condition, but a plus sign (+) did not follow. A left parenthesis in a join condition usually signals an outer-join specification and so a plus sign is expected to follow. To specify an outer join on a column in a join operation, follow the column reference in the join condition with a plus sign (+) enclosed in parentheses.

Action: Correct the SQL syntax and retry the statement.

ORA-00984

ORA-00984   column not allowed here

Cause: A column name was used in an expression where it is not permitted, such as in the VALUES clause of an INSERT statement.

Action: Check the syntax of the statement and use column names only where appropriate.

ORA-00985

ORA-00985   **invalid program name**

Cause: Probably a syntax error.

Action: Correct syntax.

ORA-00986

ORA-00986   missing or invalid group names(s)

Cause: Probably a syntax error.

Action: Correct syntax.

ORA-00987

ORA-00987   missing or invalid username(s)

Cause: No username was specified in a GRANT statement or one of the specified usernames is invalid. Valid usernames must be specified following the keyword TO in a GRANT statement to define a user. A username must begin with a letter, consist only of alphanumeric characters and the special characters $, _, and #, and be less than or equal to 30 characters. If it contains other characters, it must be enclosed in double quotation marks. It may not be a reserved word.

Action: Specify a valid username, or list of usernames, following the keyword TO in the GRANT statement.

ORA-00988

ORA-00988   **missing or invalid password(s)**

Cause: More usernames than passwords were specified in a GRANT statement.A valid password must be specified for each username listed in theGRANT statement.

Action: Enter a valid password for each username.

ORA-00989

ORA-00989   **too many passwords for usernames given**

Cause: More passwords than usernames were specified in a GRANT statement. Only one password may be entered for each username listed in the GRANT statement.

Action: Enter an equal number of usernames and passwords.

ORA-00990

ORA-00990	**missing or invalid privilege**

Cause: No privileges were specified in a GRANT privilege statement, or one of the specified privileges is invalid.

Action: Enter one or more valid privileges such as SELECT, INSERT, DELETE, UPDATE, ALTER, INDEX, REFERENCES, or ALL. More than one privilege may be granted by entering the privileges in a list separated by commas (,) or by specifying the keyword ALL to grant all privileges.

ORA-00991

| ORA-00991 | invalid format for GRANT command |

Cause: An invalid form of the GRANT command was entered. There are two forms of the GRANT command. The first form is used to grant a user database access privileges. The second form is used to grant a user object privileges.

Action: Check the command syntax and retry the statement.

ORA-00992

ORA-00992 **invalid format for REVOKE command**

Cause: An invalid form of the REVOKE command was entered. There are two forms of the REVOKE command. The first form is used to revoke a user's database access privileges. The second form is used to revoke a user's object privileges.

Action: Check the command syntax and retry the statement.

ORA-00993

ORA-00993   **missing GRANT keyword**

Cause: The keyword WITH was specified at the end of a GRANT statement without the keyword GRANT. To grant privileges to a user and the permission to grant those privileges to another user, you must specify the keywords WITH GRANT OPTION at the end of the GRANT statement.

Action: Change the keyword WITH to the keywords WITH GRANT OPTION, then retry the statement.

ORA-00994

ORA-00994 **missing OPTION keyword**

Cause: The keywords WITH GRANT were specified at the end of a GRANT statement without the keyword OPTION.

Action: Change the keywords WITH GRANT to the keywords WITH GRANT OPTION. Then retry the statement.

ORA-00995

ORA-00995   **missing or invalid synonym identifier**

Cause: In a CREATE or DROP SYNONYM statement, the synonym name was either missing or invalid.

Action: Check syntax and spelling. A valid synonym name must be specified immediately following the keyword SYNONYM in both statements. Valid synonym names must begin with a letter, consist of alphanumeric characters and the special characters $, _, and #, and be less than or equal to 30 characters. They may not be reserved words.

ORA-00996

ORA-00996   **the concatenate operator is ||, not |**

Cause: A single bar (|) was interpreted as an attempt to specify concatenation, but the concatenation operator is a double bar (||).

Action: Enter a double bar (||) for concatenation or remove the single bar (|) if concatenation was not intended.

ORA-00997

ORA-00997 **illegal use of LONG datatype**

Cause: A value of datatype LONG was used in a function or in a DISTINCT, WHERE, CONNECT BY, GROUP BY, or ORDER BY clause. A LONG value can only be used in a SELECT clause.

Action: Remove the LONG value from the function or clause.

ORA-00998

ORA-00998   **must name this expression with a column alias**

Cause: An expression or function was used in a CREATE VIEW statement, but no corresponding column name was specified. When expressions or functions are used in a view, all column names for the view must be explicitly specified in the CREATE VIEW statement.

Action: Enter a column name for each column in the view in parentheses after theview name.

ORA-00999

ORA-00999 **invalid view name**

Cause: In a CREATE VIEW statement, the view name was missing or invalid.

Action: Enter a valid view name following CREATE VIEW. Valid view names must begin with a letter, consist of only alphanumeric characters and the special characters $, _, and #, be less than or equal to 30 characters, and may not be reserved words. If the view name contains other characters, it must be enclosed in double quotation marks.

01000-01099: User Program Interface

This section lists messages generated when using the UPI to the Oracle Server. Precompilers and SQL*Forms are examples of products that use the UPI.

ORA-01000

ORA-01000   **maximum open cursors exceeded**

Cause: A host language program attempted to open too many cursors. The initialization parameter OPEN_CURSORS determines the maximum number of cursors per user.

Action: Modify the program to use fewer cursors. If this error occurs often, shut down Oracle, increase the value of OPEN_CURSORS, and then restart Oracle.

ORA-01001

ORA-01001   **invalid cursor**

Cause: Either a host language program call specified an invalid cursor or the values of the AREASIZE and MAXOPENCURSORS options in the precompiler command were too small. All cursors must be opened using the OOPEN call before being referenced in any of the following calls: SQL, DESCRIBE, NAME, DEFINE, BIND, EXEC, FETCH, and CLOSE. The Logon Data Area (LDA)must be defined by using OLON or OLOGON. If the LDA is not defined, this message is issued for the following calls: OPEN, COM, CON, ROL,and LOGOFF.

Action: Check the erroneous call statement. Specify a correct LDA area or open the cursor as required. If there is no problem with the cursor, it may benecessary to increase the AREASIZE and MAXOPENCURSORS options before precompiling.

ORA-01002

ORA-01002   **fetch out of sequence**

Cause: In a host language program, a FETCH call was issued out of sequence. A successful parse-and-execute call must be issued before a fetch. This can occur if an attempt was made to FETCH from an active set after all records have been fetched. This may be caused by fetching from a SELECT FOR UPDATE cursor after a commit. A PL/SQL cursor loop implicitly does fetches and may also cause this error.

Action: Parse and execute a SQL statement before attempting to fetch the data.

ORA-01003

ORA-01003   **no statement parsed**

Cause: A host language program call referenced a cursor with no associated parsed SQL statement. A SQL call (for example, OSQL3) must be used to pass a SQL statement to Oracle and to associate the statement with an open cursor. A cursor must already have an associated SQL statement if referenced in any of the following calls: DESCRIBE, NAME, DEFINE, BIND, EXECUTE,and FETCH.

Action: Do the SQL call, for example, OSQL, to pass the required SQL statement before referencing the cursor.

ORA-01004

ORA-01004   **default username feature not supported; logon denied**

Cause: An attempt was made to use automatic logon on a system not supporting this feature.

Action: Provide the complete username and password to log on to Oracle.

ORA-01005

ORA-01005   null password given; logon denied

Cause: An invalid password was given when logging on.

Action: Provide a valid password.

ORA-01006

ORA-01006   **bind variable does not exist**

Cause: A program issued a BIND call for a variable not listed in the associated SQL statement. Only those variables prefixed by either a colon (:) or ampersand (&) in the SQL statement may be referenced in a BIND call, OBIND or OBINDN. This error may also be caused by a mismatch between a Precompiler program and the related library, SQLLIB.

Action: Modify the BIND call to reference one of the substitute variables specified in the associated SQL statement.

ORA-01007

ORA-01007   **variable not in select list**

Cause: A reference was made to a variable not listed in the SELECT clause. In OCI, this can occur if the number passed for the position parameter is less than one or greater than the number of variables in the SELECT clause in any of the following calls: DESCRIBE, NAME, or DEFINE. In SQL*Forms or SQL*Report, specifying more variables in an INTO clause than in the SELECT clause also causes this error.

Action: In OCI, specify a position number between 1 and the number of variables in the SELECT clause. In SQL*Forms or SQL*Report, specify an equal number of variables in the SELECT and INTO clauses.

ORA-01008

ORA-01008 not all variables bound

Cause: A SQL statement containing substitution variables was executed without all variables bound. All substitution variables must have a substituted value before the SQL statement is executed.

Action: In OCI, use an OBIND or OBINDN call to substitute the required values.

ORA-01009

ORA-01009   missing mandatory parameter

Cause: A host language program call did not pass all required parameters. The syntax and parameter description for each call is shown in the *Programmer's Guide to the Oracle Precompilers.*

Action: Check the syntax for the call and enter all required parameters.

ORA-01010

ORA-01010   **invalid OCI operation**

Cause: One of the following:

- You attempted an invalid OCI operation.

- You are using an Oracle client application linked version 7.1, or higher libraries, the environment ORA_ENCRYPT_LOGIN is set to TRUE, and you attempted to connect to a version 7.0, or lower, Oracle Server.

- You are connected to a version 7.1, or higher, Oracle Server, the initialization parameter DBLINK_ENCRYPT_LOGIN is set to TRUE, and you attempted to use a database link pointing to a version 7.0, or lower, Oracle Server.

Action: For the above causes:

- Do not use the invalid OCI operation.

- If you do not wish to use encrypted connect passwords in your distributed database, set ORA_ENCRYPT_LOGIN to FALSE.If you wish to use encrypted connect passwords, you must upgrade all Oracle Servers to version 7.1, or higher.

- If you do not wish to use encrypted database links in your distributed database, set DBLINK_ENCRYPT_LOGIN to FALSE.If you wish to use encrypted database links, you must upgrade all Oracle Servers to version 7.1, or higher.

ORA-01011

ORA-01011   **cannot use Oracle7 compatibility mode with an Oracle Version 6 server**

Cause: An attempt was made to use Oracle7 compatibility mode when accessing Oracle Version 6.

Action: Do not use Oracle7 compatibility mode when accessing Oracle Version 6.

ORA-01012

ORA-01012   **not logged on**

Cause: A host language program issued an Oracle call, other than OLON or OLOGON, without being logged on to Oracle. This can occur when a user process attempts to access the database after the instance it is connected to terminates, forcing the process to disconnect.

Action: Log on to Oracle, by calling OLON or OLOGON, before issuing any Oracle calls. When the instance has been restarted, retry the action.

ORA-01013

ORA-01013   **user requested cancel of current operation**

Cause: The user interrupted an Oracle operation by entering CTRL-C, ControlC, or another canceling operation. This forces the current operation to end. This is an informational message only.

Action: Continue with the next operation.

ORA-01014

ORA-01014   **Oracle shutdown in progress**

Cause: A user tried to log on to Oracle while an instance shutdown was in progress. Oracle logons are disabled while Oracle is being shut down.

Action: Wait until Oracle is brought back up before attempting to log on.

ORA-01015

ORA-01015   **logon called recursively**

Cause: This is an internal error message not usually issued.

Action: Contact customer support.

ORA-01016

ORA-01016   **this function can be called only after a fetch**

Cause: The cursor is in an invalid state.

Action: Ensure the appropriate OCI/UPI function is called after the fetch and before the offending function.

ORA-01017

ORA-01017   **invalid username/password; logon denied**

Cause: An invalid username or password was entered in an attempt to log on to Oracle. The username and password must be the same as was specified in a GRANT CONNECT statement. If the username and password are entered together, the format is: *username/password*.

When Trusted Oracle7 is configured in OS MAC mode, this error may occur if attempting to query a table/view in a secondary database when the username was not created or not granted the CREATE SESSION privilege in the secondary database.

When Trusted Oracle7 is configured in DBMS MAC mode, this error may occur if granted the CREATE SESSION system privilege at a higher label than that attempted at login.

Action: Enter a valid username and password combination in the correct format.

For Trusted Oracle7 users, if the cause of this error is that the username was either not created or not granted the CREATE SESSION system privilege in a secondary database, ask the database administrator to authorize the username to access the secondary database. Alternatively, if the cause of this error is that the username is granted the CREATE SESSION system privilege at a higher label than that of the attempted login, either log in at that higher label or ask the database administrator to regrant the privilege at the appropriate label.

ORA-01018

ORA-01018   column does not have a LONG datatype

Cause: An attempt was made to fetch data using the LONG fetch option, but the specified column was not LONG.

Action: Re-execute the fetch without the LONG fetch option or create the table with a LONG column.

ORA-01019

ORA-01019   unable to allocate memory in the user side

Cause: The user side memory allocator returned an error.

Action: Increase the size of the process heap or switch to the old set of calls.

ORA-01020

ORA-01020 **unknown context state**

Cause: This is an internal error message not usually issued.

Action: Contact customer support.

ORA-01022

ORA-01022   **database operation not supported in this configuration**

Cause: The attempted database operation does not conform to the user programming interface (UPI) for the two communicating Oracle servers.

This problem occurs when you use a 7.0.13 server and try to use PL/SQL or remote procedure calls on a 7.0.12 server.

Action: You may need to upgrade one or more of your Oracle servers or relink your user-side application with new libraries.

ORA-01023

ORA-01023   **cursor context not found (invalid cursor number)**

Cause: The cursor number is not a valid open cursor.

Action: Make sure that the cursor is open.

ORA-01024

ORA-01024   **invalid datatype in OCI call**

Cause: An OCI program call specified an invalid datatype. In OCI calls, Oracle datatypes are specified as numbers between 1 and 7. Datatypes are described in the *Programmer's Guide to the Oracle Precompilers.*

Action: Check the datatype description and enter the correct number for the datatype.

ORA-01025

ORA-01025 **UPI parameter out of range**

Cause: An integer parameter to a UPI function is out of range. This is an internal error message not usually issued.

Action: Contact customer support.

ORA-01026

ORA-01026   **multiple buffers of size > 2000 in the bind list**

Cause: There is more than one long buffer in the bind list.

Action: Change the buffer size to be less than 255 for the bind variable bound to a normal column.

ORA-01027

| **ORA-01027** | **bind variables not allowed for data definition operations** |

Cause: An attempt was made to use a bind variable in a SQL data definition statement; for example, a CREATE TABLE AS SELECT statement in which the SELECT's WHERE clause refers to a variable.

Action: Remove the bind variable. Then retry the SQL statement.

ORA-01030

ORA-01030   SELECT ... INTO variable does not exist

Cause: The SELECT... INTO specified in the bind call does not correspond to a variable in the SQL statement.

Action: If it is not possible to correct the statement, call customer support.

ORA-01031

ORA-01031   **insufficient privileges**

Cause: An attempt was made to change the current username or password without the appropriate privilege. This error also occurs if attempting to UPDATE a table with only SELECT privileges, if attempting to CONNECT INTERNAL, or if attempting to install a database without the necessary operating-system privileges.

When Trusted Oracle7 is configured in DBMS MAC, this error may occur if the user was granted the necessary privilege at a higher label than the current login.

Action: Ask the database administrator to perform the operation or grant therequired privileges.

For Trusted Oracle7 users getting this error although granted the appropriate privilege at a higher label, ask the database administrator to regrant the privilege at the appropriate label.

ORA-01032

ORA-01032   no such userid

Cause: This is an internal error message related to Export/Import.

Action: Contact customer support.

ORA-01033

ORA-01033   **Oracle startup or shutdown in progress**

Cause: An attempt was made to log on while Oracle is being started up or shut down.

Action: Wait a few minutes. Then retry the operation.

ORA-01034

ORA-01034 **Oracle not available**

Cause: Oracle was not started up. Possible causes include the following:

- The SGA requires more space than was allocated for it.

- The operating system variable pointing to the instance was improperly defined.

Action: Refer to accompanying messages for possible causes and correct the problem mentioned in the other messages. Retry after Oracle has been initialized.

If Oracle has been initialized, then on some operating systems, verify that Oracle was linked correctly. See the platform-specific Oracle documentation.

ORA-01035

ORA-01035   **Oracle only available to users with RESTRICTED SESSION privilege**

Cause: Logins were disallowed because an instance started in restricted mode. Only users with the RESTRICTED SESSION system privilege can log on.

Action: Request that Oracle be restarted without the restricted option or obtain the RESTRICTED SESSION system privilege.

ORA-01036

ORA-01036   **illegal variable *name/num***

Cause: Unable to find bind context on user side.

Action: Make sure that the variable being bound is in the SQL statement.

ORA-01037

ORA-01037   maximum cursor memory exceeded

Cause: An attempt was made to process a complex SQL statement that consumed all available memory of the cursor.

Action: Simplify the complex SQL statement.

ORA-01038

ORA-01038   **cannot write datafile version *num* with Oracle Version *num***

Cause: An attempt was made to write datafile headers in an old format. The new format cannot be used until after the database has been verified as being compatible with this software version.

Action: Open the database to advance to the new file formats, then repeat the operation. If the operation is required before the database can be opened, use the previous software release to do the operation.

ORA-01039

ORA-01039   **insufficient privileges on underlying objects of the view**

Cause: An attempt was made to use another user's view without the necessary privileges on the underlying objects of the view.

Action: Obtain the necessary privileges or do not perform the offending operation.

ORA-01040

ORA-01040   **invalid character in password; logon denied**

Cause: There are multi-byte characters in the password or some characters in the password are not in the US7ASCII range.

Action: Retry password with valid characters.

ORA-01041

ORA-01041 **internal error. HOSTDEF extension does not exist**

Cause: The pointer to the HOSTDEF extension in HOSTDEF is null.

Action: Report as a bug to customer support.

ORA-01042

ORA-01042 **detaching a session with open cursors not allowed**

Cause:　　　An attempt was made to detach a session that has open cursors.

Action:　　　Close all the cursors before detaching the session.

ORA-01043

ORA-01043   **user side memory corruption [*num*], [*num*], [*num*], [*num*]**

Cause: The application code corrupted some of the user memory.

Action: Make certain that the application code is not overwriting memory. Contact customer support.

ORA-01044

ORA-01044   **size *num* of buffer bound to variable *name* exceeds maximum *num***

Cause: An attempt was made to bind a buffer whose total size would exceed the maximum size allowed. The total array size for arrays is calculated as (element_size) * (number of elements)

Action: Reduce the buffer size.

ORA-01045

ORA-01045   user *name* lacks CREATE SESSION privilege; logon denied

Cause: An attempt was made to connect to a userid that does not have create session privilege.

Action: If required, GRANT the user the CREATE SESSION privilege.

ORA-01046

ORA-01046   **cannot acquire space to extend context area**

Cause: Oracle could not extend the current area any further because the operating system would not supply any more space. A system-specific message should appear following this message.

Action: Close some cursors and try again or check operating system quotas to allow use of more virtual memory. See also your operating system-specific Oracle documentation.

ORA-01047

ORA-01047   **the above error occurred in schema=*name*, package=*name*, procedure=*name***

Cause: This message is displayed after a previous message to provide more specific information.

Action: See the cause and action for the previous message.

ORA-01048

ORA-01048   couldn't find the specified procedure in the given context

Cause: The user of the procedure specified a deferred RPC that does not exist.

Action: Ensure that the procedure was specified correctly and that the replication process can locate the procedure.

ORA-01049

ORA-01049   **bind by name is not supported in streamed RPC**

Cause: This error occurs when a newer server version requests from an older server version an operation that is not supported.

Action: You may need to upgrade one or more of your Oracle servers or relink your user application with new libraries.

ORA-01050

ORA-01050   **cannot acquire space to open context area**

Cause: Oracle could not open a new context area because the operating system would not supply any more space. A system-specific message should appear following this message.

Action: Close some cursors and try again or check operating system quotas to allow use of more virtual memory. See also your operating system-specific Oracle documentation.

ORA-01051

ORA-01051   deferred RPC buffer format invalid

Cause: The deferred RPC data in SYS.DEF$_CALL is corrupted.

Action: Contact customer support.

ORA-01053

ORA-01053 **user storage address cannot be read**

Cause: A bind variable or other user area could not be read by Oracle.

Action: Check that binds are done correctly on valid user buffers, then retry the operation.

ORA-01054

ORA-01054   user storage address cannot be written

Cause: A define, FETCH... INTO or SELECT... INTO, variable or other user area could not be written to by Oracle.

Action: Check that INTO variables and indicators are correctly specified. Then retry the operation.

ORA-01057

ORA-01057   **invalid or ambiguous block.field reference in user exit**

Cause: The reference to a *block.field* identifier in a user exit is incorrect or ambiguous, probably because it is misspelled or incomplete.

Action: Check syntax and identifier spelling. Then correct the reference.

ORA-01058

ORA-01058   internal New Upi interface error

Cause: An attempt was made to delete a non-existent HSTDEF extension.

Action: Contact customer support.

ORA-01059

ORA-01059   **parse expected before a bind or execute**

Cause: The client application attempted to bind a variable or execute a cursor opened in a PL/SQL block before the statement was parsed.

Action: Ensure the statement is parsed before a bind or execute.

ORA-01060

ORA-01060   **array binds or executes not allowed**

Cause: The client application attempted to bind an array of cursors or attempted to repeatedly execute against a PL/SQL block with a bind variable of type cursor.

Action: Bind a single cursor or execute the PL/SQL block one.

ORA-01070

ORA-01070   **using an old version of Oracle for the server**

Cause: An attempt was made to run an older, obsolete Oracle7 Server.

Action: Upgrade the server.

ORA-01071

ORA-01071 **cannot perform operation without starting up Oracle**

Cause: An attempt was made to perform an operation before Oracle was started.

Action: Start up Oracle. Then retry the operation.

ORA-01072

ORA-01072 **cannot stop Oracle; Oracle not running**

Cause: An attempt was made to stop Oracle, but Oracle was not running.

Action: No action required.

ORA-01073

ORA-01073   fatal connection error: unrecognized call type

Cause: An illegal internal operation was attempted. This is an internal error message not usually issued.

Action: Contact customer support.

ORA-01074

ORA-01074   cannot shut down Oracle; inside a logon session - log off first

Cause: An attempt was made to shut down Oracle inside a logon session.

Action: Log off before shutting down Oracle.

ORA-01075

ORA-01075   **currently logged on**

Cause: An attempt was made to log on while already logged on.

Action: No action required.

ORA-01076

ORA-01076 **multiple logons per process not yet supported**

Cause: Oracle does not support multiple logons per process.

Action: No action required.

ORA-01077

ORA-01077   **background process initialization failure**

Cause: A failure occurred during initialization of the background processes.

Action: Refer to the diagnostic information in the accompanying message stack or in the trace file and take appropriate action.

ORA-01078

ORA-01078   **failure in processing initialization parameters**

Cause: A failure occurred during processing of the initialization parameters during system startup.

Action: Refer to the diagnostic information in the accompanying message stack and take appropriate action.

ORA-01079

ORA-01079   **Oracle database was not properly created, operation aborted**

Cause: There was an error when the database or control file was created.

Action: Check the message signaled when the database was first created or when the control file was re-created. Take appropriate actions to re-create the database or a new control file.

ORA-01080

ORA-01080 **error in shutting down Oracle**

Cause: A failure occurred during system shutdown.

Action: Refer to the diagnostic information in the accompanying message stack and take appropriate action.

ORA-01081

ORA-01081   **cannot start already-running Oracle - shut it down first**

Cause: An attempt was made to start Oracle while it was already running.

Action: Shut down Oracle first, if you want to restart it.

ORA-01083

ORA-01083   **value of parameter *name* is inconsistent with that of other servers**

Cause: The value of the given initialization parameter is required to be the same for all servers in the parallel configuration.

Action: Change the value in the initialization parameter file to match that of the other control files.

ORA-01084

ORA-01084   **invalid argument in OCI call**

Cause: The failing OCI call contains an argument with an invalid value.

Action: Use valid argument values. For more information, see the *Programmer's Guide to the Oracle Call Interface* and the appropriate programming language supplement.

ORA-01085

ORA-01085 **preceding errors in deferred RPC to** *name.name.name*

Cause: Errors were encountered when the named procedure was executed as a deferred, remote procedure call.

Action: Correct the cause of the preceding errors.

ORA-01086

ORA-01086   **savepoint *name* never established**

Cause: An attempt was made to roll back to a savepoint that was never established.

Action: No action required.

ORA-01087

ORA-01087   **cannot start up Oracle - currently logged on**

Cause: An attempt was made to start up Oracle by a user who is currently logged on.

Action: Log off. Then issue the STARTUP command.

ORA-01088

ORA-01088   **cannot shut down Oracle while active processes exist**

Cause: Users are still logged on to the instance.

Action: Either wait for all users to log off or issue the SHUTDOWN IMMEDIATE command to force the users off the system. Alternatively, issue the SHUTDOWN ABORT command to shut down the database without waiting for users to be forced off.

ORA-01089

ORA-01089   immediate shutdown in progress - no operations are permitted

Cause: The SHUTDOWN IMMEDIATE command was used to shut down a running Oracle instance, terminating any active operations.

Action: Wait for the instance to be restarted or contact the database administrator.

ORA-01090

ORA-01090 **shutdown in progress - connection is not permitted**

Cause: The SHUTDOWN command was used to shut down a running Oracle instance, disallowing any connects to Oracle.

Action: Wait for the instance to restart or contact the database administrator.

ORA-01091

ORA-01091   **failure during startup force**

Cause: Unable to destroy the old SGA.

Action: Manually remove the old SGA. See your operating system-specific Oracle documentation for instructions. Then retry the STARTUP command.

ORA-01092

ORA-01092 **Oracle instance terminated. Disconnection forced**

Cause: The instance connected to was terminated abnormally, probably due to a SHUTDOWN ABORT. The current process was forced to disconnect from the instance.

Action: Contact the database administrator to determine when the instance is restarted. Attempt to reconnect after the instance is running again.

ORA-01093

ORA-01093   **ALTER DATABASE CLOSE only permitted with no sessions connected**

Cause: There is at lease one more session, other than the current one, logged in to the instance. ALTER DATABASE CLOSE is not permitted when other sessions are active.

Action: Find the other sessions and disconnect them. Then retry the ALTER DATABASE CLOSE command. Also, issue the SHUTDOWN IMMEDIATE command to force users off the system, or issue the SHUTDOWN ABORT command to shut down the database without waiting for users to be forced off.

ORA-01094

ORA-01094 **ALTER DATABASE CLOSE in progress. Connections not permitted**

Cause: An attempt was made to connect while the database is being closed with the ALTER DATABASE CLOSE command.

Action: Attempt to connect again when the database is open. Contact the database administrator to find out when the database will be open.

ORA-01095

ORA-01095   **DML statement processed zero rows**

Cause: A DML cursor from the OTEX() call processed 0 rows when executed.

Action: Caller can either do a rollback or ignore the message and execute the rest of the cursors in the cursor array.

ORA-01096

ORA-01096   **program version *num* incompatible with instance *num***

Cause: The program was linked with a different version of the server than the instance to which it is attempting to connect.

Action: Relink the program against the same version of Oracle as the instance or restart (startup force) the database using the program's version of the SERVER.

ORA-01097

ORA-01097 **cannot shutdown while in a transaction - commit or rollback**

Cause: An attempt was made to shut down the database while a transaction was in progress.

Action: Either commit or rollback the current transaction and then attempt to shut down the database.

ORA-01098

ORA-01098   **program interface error during LONG insert**

Cause: The application program interface could not insert or update a LONG column because of an internal error when a TTCGETV call fails. For example, a network read error or one of the parameters passed to the call is invalid.

Action: Retry the insert or update.

ORA-01099

ORA-01099   **cannot mount database in SHARED mode if started in single process mode**

Cause: An attempt was made to mount a database in parallel mode with the initialization parameter SINGLE_PROCESS set to TRUE.

Action: Either mount the database in EXCLUSIVE mode or set the initialization parameter SINGLE_PROCESS to FALSE before starting the instance in parallel (shared) mode.

01100-01250: Oracle Files

This sections lists messages generated when files are accessed by the Oracle Server.

ORA-01100

ORA-01100 **database already mounted**

Cause: An attempt was made to mount a database with the name of a currently mounted database.

Action: No action required.

ORA-01101

ORA-01101 **database being created currently mounted by some other instance**

Cause: An attempt was made to create a database with the name of a currently mounted database.

Action: Either change the database name or shut down the other instance.

ORA-01102

ORA-01102   **cannot mount database in exclusive mode**

Cause: An instance tried to mount the database in exclusive mode, but some other instance has already mounted the database in exclusive or parallel mode.

Action: Either mount the database in parallel mode or shut down all other instances before mounting the database in exclusive mode.

ORA-01103

ORA-01103   **database name *name* in control file is not *name***

Cause: The database name used does not match that in the control file.

Action: Ensure the correct control file and database name are used.

ORA-01104

ORA-01104   **number of control files *num* does not equal *num***

Cause: The number of control files used by this instance disagrees with the number of control files in an existing instance that is accessing the same database.

Action: Make sure all control files are listed in the initialization parameter CONTROL_FILES, then retry the operation.

ORA-01105

ORA-01105   **mount is incompatible with mounts by other instances**

Cause: An attempt was made to mount the database, but another instance has already mounted a database by the same name, and the mounts are not compatible. Additional messages will accompany this message to report why the mounts are incompatible.

Action: See the accompanying messages for the appropriate action to take.

ORA-01106

ORA-01106   **database must be closed before dismounting**

Cause: An attempt was made to dismount a database before it was closed.

Action: Close the database, then retry the operation.

ORA-01107

ORA-01107 **database must be mounted for media recovery**

Cause: An attempt to perform media recovery was made, but the database isnot mounted.

Action: Mount the database, then retry the operation.

ORA-01108

ORA-01108   **media recovery active on file** *name*

Cause: Media recovery is actively being applied to the given file. The file cannot be used for normal database access or crash recovery.

Action: Wait for media recovery to complete or cancel the media recovery session.

ORA-01109

ORA-01109   database not open

Cause: An attempt was made to perform an operation on an unopened database.

Action: Open the database, then retry the operation.

ORA-01110

ORA-01110 **datafile *name*: *str***

Cause: This message reports the filename involved with other messages.

Action: See the associated messages for a description of the problem.

ORA-01111

ORA-01111   **name for datafile *name* is unknown - rename to correct file**

Cause: The datafile was missing from a CREATE CONTROLFILE command or backup control file recovery was done with a control file that was saved before the file was created.

Action: Rename the missing file to the name of the real file.

ORA-01112

ORA-01112   media recovery not started

Cause: An attempt was made to continue media recovery, but media recovery had not been started.

Action: No action required.

ORA-01113

ORA-01113   **file *name* needs media recovery**

Cause: An attempt was made to open a datafile that is in need of media recovery.

Action: First apply media recovery to the datafile identified in the message, then retry the operation.

ORA-01114

ORA-01114   **IO error writing block to file *name* block # *num***

Cause: The device on which the file resides is probably off line.

Action: Restore access to the device, then retry the operation.

ORA-01115

ORA-01115   **IO error reading block from file *name* block # *num***

Cause: The device on which the file resides is probably off line.

Action: Restore access to the device, then retry the operation.

ORA-01116

ORA-01116   **error in opening datafile** *name*

Cause: Usually the file is not accessible.

Action: Make sure the datafile is in the expected location and can be accessed properly, then retry the operation.

ORA-01117

ORA-01117   **adding file *name* with illegal block size *num*, limit is *num***

Cause: An attempt was made to add a datafile with a block size that is greater than the maximum block size allowed.

Action: Retry the operation with a smaller block size.

ORA-01118

ORA-01118   **cannot add any more datafiles: limit of *num* exceeded**

Cause: An attempt to add a datafile failed because the limit for such files had already been reached.

Action: If more database space is required, export the database and re-create it with a higher limit for datafiles and perhaps increased file size.

ORA-01119

ORA-01119   **error in creating datafile _name_**

Cause: Insufficient space on device.

Action: Decrease the size of the requested file or allocate a file on another device with more available space. If a file size was not specified in the statement, then specify a file size smaller than the available space on the device.

ORA-01120

ORA-01120 **cannot remove online datafile *name*; database is open or mounted parallel**

Cause: An attempt to remove an online datafile failed because the file was not closed or was opened in parallel mode. This message can occur when trying to drop a tablespace. A tablespace cannot be dropped while users are accessing its data, index, rollback, or temporary segments or while the database is mounted in parallel mode.

Action: Shut down Oracle and mount the database in exclusive mode to drop a tablespace. To prevent users from opening the tablespace, take it off line or put the instance in restricted access mode.

ORA-01121

ORA-01121   **cannot rename datafile *name* - file is in use or recovery**

Cause: An attempt to rename an online datafile failed because the file was not closed or was being recovered. Either the file is online and the datafile is open to some instance, or another process is currently performing media recovery on the file.

Action: Close the file or shut down Oracle and mount the database in exclusive mode. End all recovery sessions. To prevent users from opening the file, take its tablespace off line or put the instance into restricted mode.

ORA-01122

ORA-01122   **datafile *name* - failed verification check**

Cause: The information in the datafile is inconsistent with information from the control file. This could be for any of the following reasons:

- The control file is from a time earlier than the datafiles.
- The datafile size does not match the size specified in the control file.
- The datafile is corrupted.

Action: Make certain that the datafiles and control files are the correct files for this database, then retry the operation.

ORA-01123

ORA-01123 **cannot start online backup; media recovery not enabled**

Cause: An attempt to start backup of an online tablespace failed because archiving was not enabled.

Action: Enable archiving and retry the operation.

ORA-01124

ORA-01124 **cannot recover online file *name* - file is in use or recovery**

Cause: An attempt to do media recover found that the file was not available for recovery. Either it is online and the database is open in some instance, or another process is currently doing media recovery on the file.

Action: Do not do media recovery.

ORA-01125

ORA-01125   **cannot disable media recovery - file *name* has online backup set**

Cause: An attempt to disable media recovery encountered an online recovery still in progress.

Action: Wait for the recovery to finish before retrying the operation.

ORA-01126

ORA-01126   **database must be mounted exclusive and not open for this operation**

Cause: An operation failed because the database was not mounted in exclusive mode.

Action: Shut down Oracle and mount the database in exclusive mode.

ORA-01127

ORA-01127   **database name *name* exceeds size limit of *num* characters**

Cause: The specified database name is too long.

Action: Shorten the database name to eight characters or fewer, then retry the operation.

ORA-01128

ORA-01128   **cannot start online backup - file *name* is offline**

Cause: A file to be backed up in the online backup is offline.

Action: Either of the following:

- Bring the file online for the online backup.
- Do an offline backup.

ORA-01129

ORA-01129   **user's default tablespace does not exist**

Cause: A user's default or temporary tablespace was dropped.

Action: Re-create the tablespace that was dropped or change the user's default or temporary tablespace.

ORA-01130

ORA-01130   **datafile version *num* incompatible with Oracle Version *num***

Cause: The named datafile was created under an incompatible version of Oracle, or the file is invalid or nonexistent.

Action: Shut down and restart the correct version of Oracle or check the references to the datafile and make sure it exists.

ORA-01131

ORA-01131   **DB_FILES files initialization parameter value *num* exceeds limit of *num***

Cause: The specified value of the initialization parameter DB_FILES is too large.

Action: Reduce the value of the DB_FILES parameter and retry the operation.

ORA-01132

ORA-01132   **length of datafile name *name* exceeds limit of *num* characters**

Cause: The specified datafile name is too long.

Action: Shorten the filename and retry the operation.

ORA-01133

ORA-01133   **length of log file name *name* exceeds limit of *num* characters**

Cause: The specified redo log filename is too long.

Action: Shorten the filename and retry the operation.

ORA-01134

ORA-01134 **database mounted exclusive by another instance**

Cause: An attempt to access a database failed because it is mounted in exclusive mode by another instance.

Action: Either shut down the other instance or wait for the other instance to either close the database or mount it in parallel mode, before retrying the operation.

ORA-01135

ORA-01135   **file _name_ accessed for DML query is offline**

Cause: A query failed because it referenced a datafile that belongs to an offline tablespace. An offline tablespace must be brought online to access its data.

Action: Wait until the tablespace is brought online before executing the query.

ORA-01136

ORA-01136   **specified size of file *num num* blocks is less than original size of*num* blocks**

Cause: An attempt was made to specify a size in the AS clause of the ALTER DATABASE CREATE datafile command that is smaller than the size needed.

Action: Issue the statement again using the correct size for the file.

ORA-01137

ORA-01137   **datafile *name* is still in the middle of going offline**

Cause: It was not possible to get the lock for a file that is offline when attempting to bring it online. The most likely cause is that the lock is still held by the instance that is took it offline.

Action: Wait for the other instance to release the lock on the file.

ORA-01138

ORA-01138   **database must either be open in this instance or not at all**

Cause: The requested operation cannot be done when the database is mounted but not open by this instance, and another instance has the database open.

Action: Execute the operation in the open instance, open the database in the current instance, or close the database in the other instances.

ORA-01139

ORA-01139   **RESET LOGS option only valid after an incomplete database recovery**

Cause: The RESET LOGS option was specified in an ALTER DATABASE OPEN statement, but there has been no incomplete recovery session.

Action: Re-execute the statement without specifying RESET LOGS.

ORA-01140

ORA-01140   **cannot end online backup - all files are offline**

Cause: All the files were offline when attempting to end an online backup.

Action: None. Online backup does not need to be ended for this tablespace.

ORA-01141

ORA-01141   **error renaming datafile *name* - new file *name* not found**

Cause: An attempt to change a datafile's name in the control file failed because no file was found with the new name.

Action: Make sure that the datafile has been properly renamed by the operating system, then retry the operation.

ORA-01142

ORA-01142   **cannot end online backup - none of the files are in backup**

Cause: None of the files were found to be in the online backup when attempting to end the online backup.

Action: None. Online backup does not need to be ended for this tablespace.

ORA-01143

ORA-01143   **cannot disable media recovery - file *name* needs media recovery**

Cause: An attempt to disable media recovery found a file that needs media recovery. Therefore, media recovery cannot be disabled.

Action: Recover the offending file or drop the tablespace to which it belongs, then retry the operation.

ORA-01144

ORA-01144   **file size *num* blocks exceeds maximum of *num* blocks**

Cause: The specified file size is larger than the maximum allowable size.

Action: Specify a smaller size.

ORA-01145

ORA-01145   **offline immediate disallowed unless media recovery enabled**

Cause: ALTER TABLESPACE ... OFFLINE IMMEDIATE or ALTER DATABASE DATAFILE ... OFFLINE is only allowed if database is in ARCHIVEREDOLOG mode.

Action: Take the tablespace offline as usual or do a shutdown abort before attempting the required operation. It is worthwhile reconsidering the backup strategy for the database. The attempted operations *can* be done if ARCHIVEREDO LOG mode is enabled.

ORA-01146

ORA-01146   **cannot start online backup - file *name* is already in backup**

Cause: An attempt was made to start an online backup, but an online backup was already started for one of the datafiles.

Action: End the first backup before beginning another.

ORA-01147

ORA-01147   **SYSTEM tablespace file *name* is offline**

Cause: A file belonging to the SYSTEM tablespace has been marked offline by the database administrator. The database cannot be started until all SYSTEM tablespace files are online and can be opened.

Action: Bring the file online.

ORA-01148

ORA-01148   **database must be mounted exclusive for this operation**

Cause: Database was not mounted exclusive.

Action: Mount the database in exclusive mode and retry the operation.

ORA-01149

ORA-01149   **cannot shutdown - file *name* has online backup set**

Cause: A normal shutdown was attempted while the online backup was in progress.

Action: End the online backup and then shut down.

ORA-01150

ORA-01150   **cannot prevent writes - file *name* has online backup set**

Cause: Tablespace cannot be taken offline or made read-only when online backup is running. Ending the backup entails writing the file header, which this operation, were it permitted, would prevent.

Action: End the online backup and then take the tablespace offline or make it read-only.

ORA-01151

ORA-01151   **use media recovery to recover block, restore backup if needed**

Cause: Crash recovery or instance recovery could not apply a change to a block because it was not the next change. This can happen if the block was corrupted and then repaired during recovery. This message is usually accompanied by ORA-01172.

Action: There is additional information for ORA-01172. Perform a RECOVER datafile for the file containing the block. If this does not resolve the problem, restore the file from a backup and recover it again.

ORA-01152

ORA-01152   file *name* was not restored from a sufficiently old backup

Cause: An incomplete recovery session was started, but an insufficient number of redo logs were applied to make the database consistent. This file is still in the future of the last redo log applied. The most likely cause of this message is forgetting to restore the file from backup before doing incomplete recovery.

Action: Apply additional redo log files until the database is consistent or restore the datafiles from an older backup and repeat recovery.

ORA-01153

ORA-01153   **an incompatible media recovery is active**

Cause: An attempt was made to start an incompatible media recovery or to open and reset the redo log files during media recovery. Media recovery sessions are incompatible if they attempt to recover the same datafile. Incomplete media recovery or OPEN RESETLOGS is incompatible with any media recovery.

Action: Wait for the completion of or cancel the other media recovery session.

ORA-01154

ORA-01154   **database busy. Open, Close, mount, and dismount not allowed now**

Cause: Some operation is in progress that expects the state of the instance to remain open or mounted.

Action: Wait for the operation to complete and try again. If attempting a normal database shutdown, try SHUTDOWN ABORT.

ORA-01155

ORA-01155   **the database is being opened, closed, mounted, or dismounted**

Cause: The attempted operation will not succeed while the instance is in one of the states mentioned in the message.

Action: Wait for the open, close, mount, or dismount to complete and try again. If attempting a normal database shutdown, try SHUTDOWN ABORT.

ORA-01156

ORA-01156	**recovery in progress may need access to files**

Cause: Either media or instance recovery is in progress. The recovery may need access to the files that the attempted operation tried to use.

Action: Wait for the recovery to complete and try again.

ORA-01157

ORA-01157   **cannot identify datafile *name* - file not found**

Cause: The background process was not able to find one of the datafiles. The database will prohibit access to this file but other files will be unaffected. However, the first instance to open the database will need to access all online datafiles. Accompanying messages from the operating system will describe why the file was not found.

Action: Make the file available to the database. Then either open the database or do ALTER SYSTEM CHECK datafiles. See also your operating system-specific Oracle documentation.

ORA-01158

ORA-01158   **database *name* already mounted**

Cause: Another instance has mounted a database of this name already.

Action: Find and shut down the instance that has this database mounted before issuing the CREATE CONTROLFILE statement.

ORA-01159

ORA-01159   **file *name* is not from same database as previous files - wrong database id**

Cause: Not all of the files specified in the CREATE CONTROLFILE statement are from the same database.

Action: Check the list of files specified in the CREATE CONTROLFILE statement and remove files that are not part of the same database.

ORA-01160

ORA-01160 **file *name* is not a *str* - it is of type *str***

Cause: The file in the datafile or log file section of the CREATE CONTROLFILE statement is not of the type listed in the command line.

Action: Check the file and determine its type. Enter the command again using the correct type for the file.

ORA-01161

ORA-01161   **database name *name* in file header does not match given name of *name***

Cause: The database name given at the command line does not match the database name found in the file header. The database name specified at the command line is incorrect.

Action: Enter the command again with the correct name for the database.

ORA-01162

ORA-01162   **block size *num* in file header does not match DB_BLOCK_SIZE *num***

Cause: CREATE CONTROLFILE discovered that the block size for this file is incompatible with the initialization parameter DB_BLOCK_SIZE used to allocate cache buffers. If this is not the first file, then there is a mixture of block sizes, or the file is corrupt.

Action: If this is the first file in the command, then correct DB_BLOCK_SIZE to match the file and restart the instance; if it is not the first file, find the correct version of the file.

ORA-01163

ORA-01163   **SIZE clause indicates *num* blocks, but should match header *num***

Cause: The size specified in bytes in the SIZE clause of the CREATE CONTROLFILE statement does not equate to the number of blocks recorded in the header.

Action: Specify the correct filename and size, in bytes.

ORA-01164

ORA-01164 **MAXLOGFILES may not exceed *num***

Cause: The value for MAXLOGFILES specified on the command line is greaterthan *num*.

Action: Retry the command with a value of MAXLOGFILES that is *num* or less.

ORA-01165

ORA-01165 **MAXDATAFILES may not exceed *num***

Cause: The value for MAXDATAFILES specified on the command line is greaterthan *num*.

Action: Retry the command with a value of MAXDATAFILES that is *num* or less.

ORA-01166

ORA-01166   **file number *num* is larger than *num num***

Cause: In the CREATE CONTROLFILE statement, the file mentioned has a file number that is larger than that specified in MAXDATAFILES or MAXLOGFILES, or the file number is larger than the maximums specified in the initialization parameter, DB_FILES.

Action: Increase the values of MAXLOGFILES, MAXDATAFILES, or of the parameter DB_FILES.

ORA-01167

ORA-01167   **two files are the same file/group number or the same file**

Cause: There is an overlap of file numbers in the files specified on the command line, or the same file is specified twice. If they are not the exact same file, then one is likely to be a backup of the other. If they are two members of the same redo log, they must be specified together in a group file spec.

Action: Confirm that the file mentioned is not a repeat of a file already mentionedin the command. If they are different files, then omit the earlier backup. If they are members of the same redo log, ensure they are in the same groupfile specification.

ORA-01168

ORA-01168   **physical block size *num* does not match size *num* of other members**

Cause: The file is located on a device with a different physical block size than the other members in the group.

Action: Use a physical device with matching block size.

ORA-01169

ORA-01169 **datafile number 1 not found. Must be present**

Cause: datafile number 1 was not specified in a CREATE CONTROLFILE statement.

Action: Locate datafile number 1 and retry the CREATE CONTROLFILE statement, making certain to include datafile on the command line.

ORA-01170

ORA-01170   **file not found** *name*

Cause: A file specified in the CREATE CONTROLFILE statement was not found. All datafiles and all redo log files, if NORESETLOGS was used, must be accessible by the process that issues the CREATE CONTROLFILE statement.

Action: Check the statement for a typing mistake in the filename and check for the existence of all files. Then issue the statement again after correcting the filenames.

ORA-01171

ORA-01171   datafile *str* is going offline due to error advancing checkpoint

Cause: The checkpoint in the file header could not be advanced. See accompanying messages for the reason. The datafile will be taken offline the same as for a write error of a data block.

Action: See accompanying messages for details. Restore access to the file, do media recovery, and bring it back online.

ORA-01172

ORA-01172   **recovery of thread *num* stuck at block *num* of file *name***

Cause: Crash recovery or instance recovery could not apply a change to a block because it was not the next change. This can happen if the block was corrupted and then repaired during recovery.

Action: Perform a RECOVER datafile for the file containing the block. If this does not resolve the problem, then restore the file from a backup and recover it again.

ORA-01173

ORA-01173   **data dictionary indicates missing datafile from system tablespace**

Cause: Either

- The database is recovered to a point in time in the future of thecontrol file.
- A datafile from the system tablespace is omitted from the issued CREATE CONTROLFILE statement.

Action: Either

- Recover the database from a more recent control file.
- Re-create the control file, ensuring all datafiles are included for the system tablespace in the command line.

ORA-01174

ORA-01174   **DB_FILES is *num* but needs to be *num* to be compatible**

Cause: The maximum number of datafiles supported by this instance is not the same as for the other instances. All instances must be able to open all the files any instance can open.

Action: Change the value of the DB_FILES initialization parameter to be compatible.

ORA-01175

ORA-01175   **data dictionary has more than the *num* files allowed by the instance**

Cause: The data dictionary has more files than the instance can support.

Action: Increase the value of the initialization parameter DB_FILES, then shut down and restart the instance.

ORA-01176

ORA-01176   **data dictionary has more than the *num* files allowed by the control file**

Cause: After a CREATE CONTROLFILE statement, the data dictionary has more datafiles than supported by the control file.

Action: Re-create the control file with a larger value for MAXDATAFILES.

ORA-01177

ORA-01177   **datafile does not match dictionary - probably old incarnation**

Cause: When comparing the control file with the data dictionary after a CREATE CONTROLFILE or OPEN RESETLOGS, it was noted that this datafile was inconsistent with the dictionary. Most likely the file is a backup of a file that was dropped from the database, and the same file number was reused for a new file. It may also be that an incomplete recovery stopped at a time when this file number was used for another datafile.

Action: Do a CREATE CONTROLFILE with the correct file or none at all.

ORA-01178

ORA-01178   **file *name* created before last CREATE CONTROLFILE, cannot be recreated**

Cause: An attempt was made to use the ALTER DATABASE CREATE datafile to re-create a datafile that existed at the last CREATE CONTROLFILE command. The information needed to re-create the file was lost with the control file that existed when the file was added to the database.

Action: Find a backup of the file and recover it. Perform incomplete recovery to the time before the file was originally created.

ORA-01179

ORA-01179   file *name* does not exist

Cause: During datafile recovery, a file was listed that was not part of the database.

Action: Recheck the filename. Remember to use single quotation marks at the SQL*DBA command line. Remember also that the filename is translated in the operating environment of SQL*DBA.

ORA-01180

ORA-01180 **cannot create datafile 1**

Cause: datafile 1 cannot be created with the ALTERDATABASECREATEDATAFILE command.

Action: Either recover the file from a backup or re-create the database.

ORA-01181

ORA-01181   file *name* created before last RESETLOGS, cannot be recreated

Cause: An attempt was made to use the ALTER DATABASE CREATE datafile command to re-create a datafile that existed before the last time the database was opened using the RESETLOGS option.

Action: Find a backup of the file and recover the backup file. Perform incomplete recovery to a time before the file was originally created.

ORA-01182

ORA-01182   **cannot create datafile *name* - file is in use or recovery**

Cause: An attempt was made to use the ALTER DATABASE CREATE datafile command to re-create a datafile that is currently online in an open instance or is currently being recovered.

Action: Close the database to all instances or end all recovery sessions. Then take the file offline and retry the operation.

ORA-01183

ORA-01183   **cannot mount database in SHARED mode**

Cause: The database is mounted in exclusive mode by another instance. It is **not** possible to mount a database in SHARED mode if it is mounted in exclusive mode by another instance.

Action: Shut down the other instance and try again.

ORA-01184

ORA-01184   **log file group *num* already exists**

Cause: An ALTER DATABASE ADD log file command specified a log number for the new redo log which is already in use.

Action: Specify a different redo log file number or let the database choose an unused value.

ORA-01185

ORA-01185 **log file group number *num* is invalid**

Cause: An ALTER DATABASE ADD log file command specified a redo log number for the new redo log that is too large.

Action: Specify a valid redo log file number.

ORA-01186

ORA-01186   **verification tests failed on file *name***

Cause: The datafile did not pass the checks to ensure it is part of the database. See the accompanying messages for the reason the verification failed.

Action: Make the correct file available to database. Then either open the database or execute ALTER DATABASE CHECK.

ORA-01187

ORA-01187   **cannot read from file *name* because it failed verification tests**

Cause: The datafile did not pass the checks to ensure it is part of the database. Reading the file is not allowed until it is verified.

Action: Make the correct file available to database. Then either open the database or execute ALTER DATABASE CHECK.

ORA-01188

ORA-01188   **block size *num* in header does not match physical block size**

Cause: A redo log file member given to CREATE CONTROLFILE is on a physical device that has a different block size than the device originally used to create the redo log.

Action: Move the file to a device with the correct block size or use the RESETLOGS option to CREATE CONTROLFILE.

ORA-01189

ORA-01189   **file is from a different RESETLOGS than previous files**

Cause: In a CREATE CONTROLFILE command, either this file or all previous files were backups from before the last RESETLOGS. This also may occur if this is a file that is offline and has been offline since before the last RESETLOGS.

Action: If the file was taken offline normally before the last RESETLOGS and is still offline, omit it from the CREATE CONTROLFILE command. Rename and online the file after the database is open. Otherwise, find the version of the mentioned file consistent with the rest of the datafiles and retry the command.

ORA-01190

ORA-01190   **control file or datafile *name* is from before the last RESETLOGS**

Cause: An attempt was made to perform media recovery when the redo log reset information in a datafile does not match the control file. Either the datafile or the control file must be a backup that was made before the most recent ALTER DATABASE OPEN RESETLOGS.

Action: Restore the file from a more recent backup.

ORA-01191

ORA-01191   **file *name* is already offline - cannot do a normal offline**

Cause: An attempt was made to take a tablespace offline as usual, but the file named in the message was already offline.

Action: Bring the datafile online first or use the IMMEDIATE option when taking the tablespace offline.

ORA-01192

ORA-01192   **must have at least one enabled thread**

Cause: At least two redo log files from at least one thread *must* be specified in the CREATE CONTROLFILE command line.

Action: Find the missing redo log files and retry the command with the newly found redo log files included in the command line.

ORA-01193

ORA-01193   **file *name* is not the same file seen at start of recovery**

Cause: An attempt was made to perform media recovery on a file, but a different copy of the file now exists since the last media recovery. Perhaps an invalid backup of the file was restored.

Action: Find the correct version of the file, then retry media recovery.

ORA-01194

ORA-01194   **file *name* needs more recovery to be consistent**

Cause: An incomplete recovery session was started, but an insufficient number of redo logs were applied to make the file consistent. The named file was not closed cleanly when it was last opened by the database. The most likely cause of this message is forgetting to restore the file from a backup before doing incomplete recovery.

Action: The file must be recovered to a time when it was not being updated. Either apply more redo logs until the file is consistent or restore the file from an older backup and repeat recovery.

ORA-01195

ORA-01195   **online backup of file *name* needs more recovery to be consistent**

Cause: An incomplete recovery session was started, but an insufficient number of redo logs were applied to make the file consistent. The reported file is an online backup that must be recovered to the time the backup ended.

Action: Either apply more redo logs until the file is consistent or restore the file from an older backup and repeat the recovery.

ORA-01196

ORA-01196   **file *name* is inconsistent due to a failed media recovery session**

Cause: A media recovery session failed while the named file was being recovered. The file is in an inconsistent state and no more recovery was successfully completed on this file.

Action: Either apply more redo logs until the file is consistent or restore the file from an older backup and repeat the recovery.

ORA-01197

ORA-01197 **thread *num* only contains one log**

Cause: During a CREATE CONTROLFILE statement, all threads represented in the redo logs must be represented by at least two redo logs: a last redo log and a second redo log. The named thread does not contain two redo logs.

Action: Either find more redo logs from the named thread or remove all references to redo logs from that thread, then retry the command.

ORA-01198

ORA-01198   **must specify size for log file *name* if RESETLOGS**

Cause: File sizes must be given for all redo log files if doing a CREATE CONTROLFILE with the RESETLOGS option.

Action: Retry the command with the appropriate redo log file size.

ORA-01200

ORA-01200 **actual file size of *num* is smaller than correct size of *num* blocks**

Cause: The size of the file, as returned by the operating system, is smaller than the size of the file as indicated in the file header and the control file. Somehow the file has been truncated.

Action: Restore a good copy of the datafile from a backup and perform recovery as needed.

ORA-01201

ORA-01201   **file size *num* in header does not match size *num* in control file**

Cause: The file sizes in the control file and in the file header do not match. One of the files probably is corrupted.

Action: Replace the corrupted file with a good copy and perform recovery as needed.

ORA-01202

ORA-01202   **wrong incarnation of this file - wrong creation time**

Cause: The creation time in the file header is not the same as the creation time in the control file. This is probably a copy of a file that was dropped.

Action: Restore a current copy of the datafile and perform recovery as needed.

ORA-01203

ORA-01203   wrong incarnation of this file - wrong creation SCN

Cause: The creation change number in the file header is not the same as the creation change number in the control file. This is probably a copy of a file that was dropped.

Action: Restore a current copy of the datafile and perform recovery as needed.

ORA-01204

ORA-01204   **wrong file - file number is *num* rather than *num***

Cause: The file number in the file header is not correct. This is probably a restored backup of the wrong file, but from the same database.

Action: Restore a copy of the correct datafile and perform recovery as needed.

ORA-01205

ORA-01205   **not a datafile - type number in header is *num***

Cause: The file type in the header is not correct for a datafile. This is probably a redo log file or control file.

Action: Restore a copy of the correct datafile and perform recovery as needed.

ORA-01206

ORA-01206 **file is not part of this database - wrong database id**

Cause: The database ID in the file header does not match the database ID in the control file. The file may be from a different database, or it may not be a datafile at all. If the database was rebuilt, this may be a file from before the rebuild.

Action: Restore a copy of the correct datafile and perform recovery as needed.

ORA-01207

ORA-01207 **file is more recent than control file - old control file**

Cause: The control file change sequence number in the datafile is greater than the number in the control file. This implies that the wrong control file is being used.

Note: If this message occurs repeatedly, by opening the database many times, the message may stop occurring without the problem being corrected. Every attempt to open the database will advance the control file change sequence number until it is great enough.

Action: Use the current control file or perform cancel-based recovery to make the control file current. Be sure to follow all restrictions on performing a cancel-based recovery.

ORA-01208

ORA-01208   **datafile is an old version - not accessing current version**

Cause: The checkpoint in the file header is less recent than in the control file. If opening a database that is already open by another instance or if another instance just caused this file to be placed online, then the database is probably looking at a different version of the file. Otherwise, a backup of the file was probably restored while the file was still in use.

Action: Make correct file available to database. Then either open the database or do ALTER DATABASE CHECK.

ORA-01209

ORA-01209   **datafile is from before the last RESETLOGS**

Cause: The reset redo log data in the file header does not match the control file. If the database is closed or the file is offline, this is an old backup that was taken before the last ALTER DATABASE OPEN RESETLOGS command. If opening a database that is already open by another instance or if another instance just put this file online, then we are probably looking at a different version of the file. Otherwise a backup of the file was probably restored while the file was in use.

Action: Make the correct file available to the database. Then either open the database or issue an ALTER DATABASE CHECK statement.

ORA-01210

ORA-01210   **datafile header is media corrupt**

Cause: The file header block is internally inconsistent. The beginning of the block has a header with a checksum and other data for insuring the consistency of the block. It is possible that the last disk write did not operate correctly. The most likely problem is that this is not a datafile for any database.

Action: Make the correct file available to the database. Refer to any trace file generated after this message for more information. If the trace file indicates that the checksum is wrong, restore the file from a backup and perform media recovery.

ORA-01211

ORA-01211 **Version 6 datafile is not from conversion to Oracle7**

Cause: The file is not a copy of the file last used under Version 6. When converting a database from Version 6 to Oracle7, the conversion utility must be run the last time the database is opened under Version 6. Only the datafiles that were current when the conversion was done may be accessed by Oracle7. Either this datafile is a backup taken from before the conversion or the database was opened by Version 6 after the conversion.

Action: Have the operating system make the correct datafile available to the database or repeat the Version 6 to Oracle7 conversion.

ORA-01212

ORA-01212 **MAXLOGMEMBERS may not exceed** *num*

Cause: The value specified for MAXLOGMEMBERS is too large.

Action: Retry the command with a smaller value for MAXLOGMEMBERS.

ORA-01213

ORA-01213   MAXINSTANCES may not exceed *num*

Cause: The value specified for MAXINSTANCES is too large.

Action: Retry the command with a smaller value for MAXINSTANCES.

ORA-01214

ORA-01214 **MAXLOGHISTORY may not exceed** *num*

Cause: The value specified for MAXLOGHISTORY is too large.

Action: Retry the command with a smaller value for MAXLOGHISTORY.

ORA-01215

ORA-01215   **enabled thread *num* is missing after CREATE CONTROLFILE**

Cause: A CREATE CONTROLFILE command did not list all of the enabled threads for the database.

Action: Issue the CREATE CONTROLFILE command again and include all of the enabled threads.

ORA-01216

ORA-01216   **thread *num* is expected to be disabled after CREATE CONTROLFILE**

Cause: A thread specified in the CREATE CONTROLFILE command was enabled, but the datafiles indicate that the thread should be disabled. This is probably because the redo log files supplied to the CREATE CONTROLFILE command are from a time before the thread was disabled.

Action: This thread is not required for the operation of the database. Retry the command without specifying the thread. If desired, the thread can be re-created after the database has been opened.

ORA-01217

ORA-01217   **log file member belongs to a different redo log file group**

Cause: A member of a redo log file group specified in the CREATE CONTROLFILE command is not part of the same group as the previous members.

Action: Group together the correct members of the redo log file group for the CREATE CONTROLFILE command.

ORA-01218

ORA-01218   **log file member is not from the same point-in-time**

Cause: A member of a redo log file group specified in the CREATE CONTROLFILE command is from a different point in time from the previous members. One of the members specified may be an older, possibly a backup copy, version of the redo log.

Action: Find the correct version of the redo log or leave it out of the CREATE CONTROLFILE command.

ORA-01219

ORA-01219   database not open: queries allowed on fixed tables/views only

Cause: A query was issued against an object not recognized as a fixed table or fixed view before the database was opened.

Action: Revise the query to include only fixed objects or open the database.

ORA-01220

ORA-01220 **file based sort illegal before database is open**

Cause: A query issued against a fixed table or view required a temporary segment for sorting before the database was open. Only in-memory sorts are supported before the database is open.

Action: Rephrase the query to avoid a large sort and increase the SORT_AREA_SIZE parameter in the initialization parameter file to enable the sort to be done in memory.

ORA-01221

ORA-01221   datafile *name* is not the same file to a background process

Cause: When the database writer opens the datafile, it is accessing a different physical file than the foreground doing the recovery. The time-stamp set in the file header by the foreground was not found by the background. It may be that the background process could not read the file at all.

Action: Look in the DBWR trace file for the error it received when attempting toread the file header. Reconfigure the operating system as needed to havethe filename successfully access the same file when opened by a background process.

ORA-01222

ORA-01222   **MAXINSTANCES of *num* requires MAXLOGFILES be at least *num*, not *num***

Cause: An attempt was made to create a database or control file that does not have room for at least two redo logs per thread of redo. A thread of redo must have two online redo logs in order to be enabled. It does not make sense to allow more redo threads than can be supported by the logs.

Action: Either reduce the MAXINSTANCES argument or increase MAXLOGFILES.

ORA-01223

ORA-01223 **RESETLOGS must be specified to set a new database name**

Cause: The SET database name option was specified to CREATE CONTROLFILE, but RESETLOGS was not specified. The database name can only be changed when opening the database with RESETLOGS.

Action: Either add the RESETLOGS option or drop the SET option toCREATE CONTROLFILE.

ORA-01224

ORA-01224   **group number in header *name* does not match GROUP *name***

Cause: Group number specified at CREATE CONTROLFILE does not match the group number stored in the header. Most likely the specification is wrong.

Action: Omit the GROUP option or give the correct one.

ORA-01225

ORA-01225   **thread number *num* is greater than MAXINSTANCES *num***

Cause: The redo log is for a thread greater than the MAXINSTANCES arguments.

Action: Increase the value for MAXINSTANCES and retry the command.

ORA-01226

ORA-01226 **file header of redo log member is inconsistent with other member**

Cause: The redo log file member in the accompanying message is for the same group as the previous members, but other fields in the header are different. Either a file header is corrupted, or some file is a member of a deleted redo log.

Action: Correct the redo log member or omit this member from the command.

ORA-01227

ORA-01227 **log** *name* **is inconsistent with other log**

Cause: The redo log file in the accompanying message is inconsistent with the contents of other redo logs given in the CREATE CONTROLFILE command. Either a file header is corrupted, or some file is an old copy rather than the current version. The problem may not be with the redo log listed because all that can be detected is that there is an inconsistency. All redo log files listed in the command must be the current versions of the online redo logs.

Action: Find the correct online redo logs or use the RESETLOGS option.

ORA-01228

ORA-01228   **SET DATABASE option required to install seed database**

Cause: The SET DATABASE option was not included in the CREATE CONTROLFILE command when installing a seed database. The database does not have a database ID because it is intended to be installed at multiple sites, and each site needs to be a different database with its own database ID. Both the SET DATABASE and RESETLOGS options must be specified to create the control file for this database.

Action: Retry the command with the SET DATABASE and RESETLOGS options.

ORA-01229

ORA-01229   datafile *name* is inconsistent with logs

Cause: The datafile in the accompanying message is inconsistent with the contents of the redo logs given in the CREATE CONTROLFILE command. The most likely cause is that one or more of the online redo logs was missing from the command. It is also possible that one or more of the redo logs is an old copy rather than the current version. All online redo log files must be listed in the command and must be the current versions of the online redo logs.

Action: Retry the command with the correct online redo logs or use theRESETLOGS option.

ORA-01230

ORA-01230   **cannot make read only - file *name* is offline**

Cause: An attempt to make a tablespace read-only encountered an offline file.

Action: Bring the file online and retry the operation.

ORA-01231

ORA-01231   cannot make read write - file *name* is offline

Cause: An attempt to make a tablespace read-write encountered an offline file.

Action: Bring the file online and retry the operation.

ORA-01232

ORA-01232   **cannot start online backup - file *name* is read only**

Cause: An attempt to start an online backup encountered a read-only file.

Action: Read-only files do not need to be set into online backup mode. Start the backup without using the BEGIN BACKUP command.

ORA-01233

ORA-01233   **file name is read only - cannot recover using backup control file**

Cause: An attempt to do media recovery using a backup control file encountered a read-only file. Typically, read-only files do not need to be recovered. However, if the recovery is using a backup control file, all online files must be recovered.

Action: Take the read-only file offline and bring the tablespace online only after opening the database. Ensure that the backup control file was in use when the currently read-only file was read-write. If such a control file is unavailable, you may be able to re-create it using the CREATE CONTROLFILE command.

01400-01489: SQL Execution

This section lists messages generated during SQL execution.

ORA-01400

ORA-01400   **primary key or mandatory (NOT NULL) column is missing or NULLduring insert**

Cause: When inserting or updating rows, a value for a column defined as NOT NULL was not specified.

Action: Specify a value for each NOT NULL column or modify the table definition to allow NULL values in columns now defined as NOT NULL.

ORA-01401

ORA-01401   **inserted value too large for column**

Cause: The value entered is larger than the maximum width defined for the column.

Action: Enter a value smaller than the column width or use the MODIFY option with ALTER TABLE to expand the column width.

ORA-01402

ORA-01402	view WITH CHECK OPTION where-clause violation

Cause: An INSERT or UPDATE statement was attempted on a view created with the CHECK OPTION. This would have resulted in the creation of a row that would not satisfy the view's WHERE clause.

Action: Examine the view's WHERE clause in the dictionary table VIEWS. If the current view does not have the CHECK OPTION, then its FROM clause must reference a second view that is defined using the CHECK OPTION. The second view's WHERE clause must also be satisfied by any INSERT or UPDATE statements. To insert the row, it may be necessary to insert it directly into the underlying table, rather than through the view.

ORA-01403

ORA-01403   no data found

Cause: In a host language program, all records have been fetched. The return code from the fetch was +4, indicating that all records have been returned from the SQL query.

Action: Terminate processing for the SELECT statement.

ORA-01404

ORA-01404   **ALTER COLUMN will make a concatenated index too large**

Cause: Increasing the length of a column would cause the combined length of the columns specified in a previous CREATE INDEX statement to exceed the maximum index length (255). The total index length is computed as the sum of the width of all indexed columns plus the number of indexed columns. Date fields are calculated as a length of 7, character fields are calculated at their defined width, and numeric fields are length 22.

Action: The only way to alter the column is to drop the affected index. The index cannot be recreated if to do so would exceed the maximum index width.

ORA-01405

ORA-01405   **fetched column value is NULL**

Cause: The INTO clause of a FETCH operation contained a NULL value, and no indicator was used. The column buffer in the program remained unchanged, and the cursor return code was +2. This is an error unless you are running Oracle7 with DBMS=6, emulating version 6, in which case it is only a warning.

Action: You may do any of the following:

- Use the NVL function to convert the retrieved NULL to another value, such as zero or blank. This is the simplest solution.

- Use an indicator to record the presence of the NULL. You probably should use this option when you want a specific action to be taken when a NULL arises.

- Revise the cursor definition so that no columns possibly containing NULL values are retrieved.

ORA-01406

ORA-01406   **fetched column value was truncated**

Cause: In a host language program, a FETCH operation was forced to truncate a character string. The program buffer area for this column was not large enough to contain the entire string. The cursor return code from the fetch was +3.

Action: Increase the column buffer area to hold the largest column value or perform other appropriate processing.

ORA-01407

ORA-01407   **cannot update mandatory NOT NULL column to NULL**

Cause: An attempt was made to update data in a NOT NULL column to a NULL value.

Action: Specify a value for the column or modify the table definition to accept NULL values in that column.

ORA-01408

ORA-01408   **such column list already indexed**

Cause: A CREATE INDEX statement specified a column that is already indexed. A single column may be indexed only once. Additional indexes may be created on the column if it is used as a portion of a concatenated index, that is, if the index consists of multiple columns.

Action: Do not attempt to re-index the column, as it is unnecessary. To create a concatenated key, specify one or more additional columns in the CREATE INDEX statement.

ORA-01409

ORA-01409   **NOSORT option may not be used; rows are not in ascending order**

Cause: Creation of index with NOSORT option when rows were not ascending. The NOSORT option may only be used for indexes on groups of rows that already are in ascending order.

For non-unique indexes the ROWID is considered part of the index key. This means that two rows that appear to be stored in ascending order may not be. If you create an index NOSORT, and two of the rows in the table have the same index values, but get split across two extents, the data block address of the first block in the second extent can be less than the data block address of the last block in the first extent. If these addresses are not in ascending order, the ROWIDs are not either. Since these ROWIDs are considered part of the index key, the index key is not in ascending order, and the create index NOSORT fails.

Action: Create the index without the NOSORT option or ensure that the table is stored in one extent.

ORA-01410

ORA-01410   invalid ROWID

Cause: A ROWID was entered incorrectly. ROWIDs must be entered as formatted hexadecimal strings using only numbers and the characters A through F.A typical ROWID format is '000001F8.0001.0006'.

Action: Check the format, then enter the ROWID using the correct format. ROWID format: block id, row in block, file id.

ORA-01411

ORA-01411   **cannot store the length of column in the indicator**

Cause: Oracle tried to fetch a column more than 64K long and could not store the length of the column in the given indicator size of 2 bytes.

Action: Use the new bind type with callbacks to fetch the long column.

ORA-01412

ORA-01412   zero length not allowed for this datatype

Cause: The length for datatype 97 is 0.

Action: Specify the correct length for the datatype.

ORA-01413

ORA-01413   **illegal value in packed decimal number buffer**

Cause: The user buffer bound by the user as a packed decimal number contained an illegal value.

Action: Use a legal value.

ORA-01414

ORA-01414   **invalid array length when trying to bind array**

Cause: An attempt was made to bind an array without either a current array length pointer or a zero maximum array length.

Action: Specify a valid length.

ORA-01415

ORA-01415   **outer-join (+) may not be applied to the label pseudo-column**

Cause: The user specified an outer-join (+) following a reference to the label pseudo-column.

Action: Work around the problem by defining a view on the table and perform an outer join on the view column corresponding to a label.

ORA-01416

ORA-01416   **two tables cannot be outer-joined to each other**

Cause: Two tables in a join operation specified an outer join with respect to each other. If an outer join is specified on one of the tables in a join condition, it may not be specified on the other table.

Action: Remove the outer join specification (+) from one of the tables, then retry the operation.

ORA-01417

ORA-01417   a table may be outer joined to at most one other table

Cause: A table in a join operation specified an outer join to more than one other table. A table may specify an outer join to only one other table.

Action: Specify only one outer join (+) to this table, then retry the operation.

ORA-01418

ORA-01418   **specified index does not exist**

Cause: An ALTER INDEX, DROP INDEX, or VALIDATE INDEX statement specified the name of an index that does not exist. Only existing indexes can be altered, dropped, or validated. Existing indexes may be listed by querying the data dictionary.

Action: Specify the name of an existing index in the ALTER INDEX, DROP INDEX, or VALIDATE INDEX statement.

ORA-01422

ORA-01422   **exact fetch returns more than requested number of rows**

Cause: More rows were returned from an exact fetch than specified.

Action: Rewrite the query to return fewer rows or specify more rows in the exact fetch.

ORA-01423

ORA-01423   **error encountered while checking for extra rows in exact fetch**

Cause: An error was encountered during the execution of an exact fetch. This message will be followed by more descriptive messages.

Action: See the accompanying messages and take appropriate action.

ORA-01424

ORA-01424   **missing or illegal character following the escape character**

Cause: The character following the escape character in LIKE pattern is missing or not one of the wildcard characters '%' or '_'.

Action: Remove the escape character or specify the missing character.

ORA-01425

ORA-01425   **escape character must be character string of length 1**

Cause: Given escape character for LIKE is not a character string of length 1.

Action: Change it to a character string of length 1.

ORA-01426

ORA-01426   numeric overflow

Cause: Evaluation of a value expression has caused an overflow or possibly an underflow.

Action: Rewrite the expression as a series of expressions with fewer operands than the 'overloaded' expression.

ORA-01427

ORA-01427 **single-row subquery returns more than one row**

Cause: The outer query must use one of the keywords ANY, ALL, IN, or NOT IN to specify values to compare because the subquery returned more than one row.

Action: Use ANY, ALL, IN, or NOT IN to specify which values to compare or reword the query so only one row is retrieved.

ORA-01428

ORA-01428   **argument *num* is out of range**

Cause: An illegal value for a mathematical function argument was specified. For example

SELECT SQRT(-1) "Square Root" FROM DUAL;

Action: See *Oracle7 Server SQL Reference* manual for valid input and ranges of the mathematical functions.

ORA-01429

ORA-01429   **a date in binary format has an out of range value**

Cause: The date, in binary format, for an external type DATE was out of range for one or more of the seven possible bytes.

Action: Select values for each of the seven bytes that conform to the required date specification or use the TO_DATE function with character data.

ORA-01430

ORA-01430 **column being added already exists in table**

Cause: An ALTER TABLE ADD statement specified the name of a column that is already in the table. All column names must be unique within a table.

Action: Specify a unique name for the new column, then re-execute the statement.

ORA-01431

ORA-01431   **internal inconsistency in GRANT command**

Cause: An internal error occurred while attempting to execute a GRANT statement.

Action: Contact customer support.

ORA-01432

ORA-01432 **public synonym to be dropped does not exist**

Cause: The synonym specified in DROP PUBLIC SYNONYM is not a valid public synonym. It may be a private synonym.

Action: Correct the synonym name or use DROP SYNONYM if the synonym is not public.

ORA-01433

ORA-01433   **synonym to be created is already defined**

Cause: A CREATE SYNONYM statement specified a synonym name that is the same as an existing synonym, table, view, or cluster. Synonyms may not have the same name as any other synonym, table, view, or cluster available to the user creating the synonym.

Action: Specify a unique name for the synonym, then re-execute the statement.

ORA-01434

ORA-01434   **private synonym to be dropped does not exist**

Cause: A DROP SYNONYM statement specified a synonym that does not exist. Existing synonym names may be listed by querying the data dictionary.

Action: Specify the name of an existing synonym in the DROP SYNONYM statement.

ORA-01435

ORA-01435 **user does not exist**

Cause: This message is caused by any reference to a non-existent user. For example, it occurs if a SELECT, GRANT, or REVOKE statement specifies a username that does not exist. Only a GRANT CONNECT statement may specify a new username. All other GRANT and REVOKE statements must specify existing usernames. If specified in a SELECT statement, usernames must already exist.

Action: Specify only existing usernames in the SELECT, GRANT, or REVOKE statement or ask the database administrator to define the new username.

ORA-01436

ORA-01436   **CONNECT BY loop in user data**

Cause: The condition specified in a CONNECT BY clause caused a loop in the query, where the next record to be selected is a descendant of itself. When this happens, there can be no end to the query.

Action: Check the CONNECT BY clause and remove the circular reference.

ORA-01437

ORA-01437   **cannot have join with CONNECT BY**

Cause: A join operation was specified with a CONNECT BY clause. If a CONNECT BY clause is used in a SELECT statement for a tree-structured query, only one table may be referenced in the query.

Action: Remove either the CONNECT BY clause or the join operation from the SQL statement.

ORA-01438

| **ORA-01438** | **value larger than specified precision allows for this column** |

Cause: When inserting or updating records, a numeric value was entered that exceeded the precision defined for the column.

Action: Enter a value that complies with the numeric column's precision, or use the MODIFY option with the ALTER TABLE command to expand the precision.

ORA-01439

ORA-01439   **column to be modified must be empty to change datatype**

Cause: An ALTER TABLE MODIFY statement attempted to change the datatype of a column containing data. A column whose datatype is to be altered must contain only NULL values.

Action: To alter the datatype, first set all values in the column to NULL.

ORA-01440

ORA-01440   **column to be modified must be empty to decrease precision or scale**

Cause: An ALTER TABLE MODIFY statement attempted to decrease the scale or precision of a numeric column containing data. In order to decrease either of these values, the column must contain only NULL values. An attempt to increase the scale without also increasing the precision will also cause this message.

Action: Set all values in the column to NULL before decreasing the numeric precision or scale. If attempting to increase the scale, increase the precision in accordance with the scale or set all values in the column to NULL first.

ORA-01441

ORA-01441   **column to be modified must be empty to decrease column length**

Cause: An ALTER TABLE MODIFY statement attempted to decrease the size of a character field containing data. A column whose maximum size is to be decreased must contain only NULL values.

Action: Set all values in column to NULL before decreasing the maximum size.

ORA-01442

ORA-01442 column to be modified to NOT NULL is already NOT NULL

Cause: An ALTER TABLE MODIFY statement attempted to change a column specification unnecessarily, from NOT NULL to NOT NULL.

Action: No action required.

ORA-01443

ORA-01443   **internal inconsistency; illegal datatype in resultant view column**

Cause: An internal error occurred in referencing a view.

Action: Contact customer support.

ORA-01444

ORA-01444   internal inconsistency; internal datatype maps to invalid external type

Cause: This is an internal error message not normally issued.

Action: Contact customer support.

ORA-01445

ORA-01445 **cannot select ROWID from view of more than one table**

Cause: A SELECT statement attempted to select ROWIDs from a view derived from a join operation. Because the rows selected in the view do not correspond to underlying physical records, no ROWIDs can be returned.

Action: Remove ROWID from the view selection clause, then re-execute the statement.

ORA-01446

ORA-01446   **cannot select ROWID from view with DISTINCT, GROUP BY, etc.**

Cause: A SELECT statement attempted to select ROWIDs from a view containing columns derived from functions or expressions. Because the rows selected in the view do not correspond to underlying physical records, no ROWIDs canbe returned.

Action: Remove ROWID from the view selection clause, then re-execute the statement.

ORA-01447

ORA-01447   **ALTER TABLE does not operate on clustered columns**

Cause: An ALTER TABLE MODIFY statement specified a column used to cluster the table. Clustered columns may not be altered.

Action: To alter the column, first recreate the table in non-clustered form. The column's size can be increased at the same time.

ORA-01448

ORA-01448   **index must be dropped before changing to desired type**

Cause: An ALTER TABLE MODIFY statement attempted to change an indexed character column to a LONG column. Columns with the datatype LONG may not be indexed, so the index must be dropped before the modification.

Action: Drop all indexes referencing the column before changing its datatype to LONG.

ORA-01449

ORA-01449 **column contains NULL values; cannot alter to NOT NULL**

Cause: An ALTER TABLE MODIFY statement attempted to change the definition of a column containing NULL values to NOT NULL. The column may not currently contain any NULL values if it is to be altered to NOT NULL.

Action: Set all NULL values in the column to values other than NULL before ALTERING the column to NOT NULL.

ORA-01450

ORA-01450   **maximum key length exceeded**

Cause: The combined length of all the columns specified in a CREATE INDEX statement exceeded the maximum index length. The maximum index length varies by operating system. The total index length is computed as the sum of the width of all indexed columns plus the number of indexed columns. Date fields have a length of 7, character fields have their defined length, and numeric fields have a length of 22. Numeric length = (precision/2) + 1. If negative, add +1.

Action: Select columns to be indexed so the total index length does not exceed the maximum index length for the operating system. See also your operating system-specific Oracle documentation.

ORA-01451

ORA-01451 **column to be modified to NULL cannot be modified to NULL**

Cause: The column may already allow NULL values, the NOT NULL constraint is part of a primary key or check constraint, or an ALTER TABLE MODIFY statement attempted to change a column specification unnecessarily, from NULL to NULL.

Action: If a primary key or check constraint is enforcing the NOT NULL constraint, then drop that constraint.

ORA-01452

ORA-01452   cannot CREATE UNIQUE INDEX; duplicate keys found

Cause: A CREATE UNIQUE INDEX statement specified one or more columns that currently contain duplicate values. All values in the indexed columns must be unique by row to create a UNIQUE INDEX.

Action: If the entries need not be unique, remove the keyword UNIQUE from the CREATE INDEX statement, then re-execute the statement. If the entries must be unique, as in a primary key, then remove duplicate values before creating the UNIQUE index.

ORA-01453

ORA-01453   **SET TRANSACTION must be first statement of transaction**

Cause: A transaction was not processed properly because the SET TRANSACTION statement was not the first statement.

Action: Commit or roll back the current transaction before using the statement SET TRANSACTION.

ORA-01454

ORA-01454 **cannot convert column into numeric datatype**

Cause: A non-numeric value could not be converted into a number value.

Action: Check the value to make sure it contains only numbers, a sign, a decimal point, and the character "E" or "e", then retry the operation.

ORA-01455

ORA-01455   **converting column overflows integer datatype**

Cause: The converted form of the specified expression was too large for thespecified datatype.

Action: Define a larger datatype or correct the data.

ORA-01456

ORA-01456   **may not perform insert/delete/update operation inside a READONLY transaction**

Cause: A non-DDL INSERT/DELETE/UPDATE or SELECT FOR UPDATE operation was attempted.

Action: Commit or roll back the current transaction, then retry the operation.

ORA-01457

ORA-01457 **converting column overflows decimal datatype**

Cause: The converted form of the specified expression was too large for the specified type. The problem also occurs in COBOL programs when using COMP-3 in the picture clause, which is acceptable to the Pro*COBOL Precompiler and to COBOL but results in this error.

Action: Define a larger datatype or correct the data.

ORA-01458

ORA-01458   **invalid length inside variable character string**

Cause: An attempt was made to bind or define a variable character string with a buffer length less than the two-byte minimum requirement.

Action: Increase the buffer size or use a different type.

ORA-01459

ORA-01459   **invalid length for variable character string**

Cause: The buffer length was less than the minimum required (two bytes) or greater than its length at bind time minus two bytes.

Action: None. Buffer length is set correctly by Oracle at fetch time.

ORA-01460

ORA-01460   **unimplemented or unreasonable conversion requested**

Cause: The requested format conversion is not supported.

Action: Remove the requested conversion from the SQL statement. Check the syntax for the TO_CHAR, TO_DATE, and TO_NUMBER functions to see which conversions are supported.

ORA-01461

ORA-01461   **can bind a LONG value only for insert into a LONG column**

Cause: An attempt was made to insert a value from a LONG datatype into another datatype. This is not allowed.

Action: Do not try to insert LONG datatypes into other types of columns.

ORA-01462

ORA-01462   **cannot insert string literals longer than 2000 characters**

Cause: The longest literal supported by Oracle consists of 2000 characters.

Action: Reduce the number of characters in the literal to 2000 characters or fewer or use the VARCHAR2 or LONG datatype to insert strings exceeding 2000 characters.

ORA-01463

ORA-01463 **cannot modify column datatype with current constraints**

Cause: An attempt was made to modify the datatype of a column that has referential constraints or that has check constraints that only allow changing the datatype from CHAR to VARCHAR or vice versa.

Action: Remove the constraint(s) or do not perform the offending operation.

ORA-01464

ORA-01464   **circular grant (granting to grant ancestor) of table or view**

Cause: The user in the TO clause of the GRANT statement has already been GRANTed privileges on this table.

Action: Do not GRANT privileges on a table to the user who originally GRANTed privileges on that table. The statement in error is probably unnecessary.

ORA-01465

ORA-01465 **invalid hex number**

Cause: In an UPDATE statement following a SELECT FOR UPDATE, part of the ROWID contains invalid characters. ROWID must be expressed in the proper and expected format for ROWID and within quotes.

Action: Enter the ROWID just as it was returned in the SELECT FOR UPDATE.

ORA-01466

ORA-01466   **unable to read data -- object definition has changed**

Cause: This is a time-based read consistency error for a database object, such as a table or index. Either of the following may have happened:

- The query was parsed and executed with a snapshot older than the time the object was changed.
- The creation time-stamp of the object is greater than the current system time. This happens, for example, when the system time is set to a time earlier than the creation time of the object.

Action: If the cause is

- an old snapshot, then commit or rollback the transaction andresume work.
- a creation time-stamp in the future, ensure the system time is set correctly.

If the object creation time-stamp is still greater than the system time, then export the object's data, drop the object, recreate the object so it has a new creation time-stamp, import the object's data, and resume work.

ORA-01467

ORA-01467   sort key too long

Cause: A DISTINCT, GROUP BY, ORDER BY, or SET operation requires a sort key longer than that supported by Oracle. Either too many columns or too many group functions were specified in the SELECT statement.

Action: Reduce the number of columns or group functions involved in the operation.

ORA-01468

ORA-01468 a predicate may reference only one outer-joined table

Cause: A predicate in the WHERE clause has two columns from different tableswith "(+)".

Action: Change the WHERE clause so that each predicate has a maximum of one outer-join table.

ORA-01469

ORA-01469 **PRIOR can only be followed by a column name**

Cause: An invalid column name was specified after the PRIOR keyword.

Action: Check syntax, spelling, use a valid column name, and try again.

ORA-01471

ORA-01471 **cannot create a synonym with the same name as object**

Cause: An attempt was made to create a private synonym with the same name as the object to which it refers. This error typically occurs when a user attempts to create a private synonym with the same name as one of their objects.

Action: Choose a different synonym name or create the synonym under a different username.

ORA-01472

ORA-01472 **cannot use CONNECT BY on view with DISTINCT, GROUP BY, etc.**

Cause: CONNECT BY cannot be used on a view where there is not a correspondence between output rows and rows of the underlying table.

Action: Remove the DISTINCT or GROUP BY from the view or move the CONNECT BY clause into the view.

ORA-01473

ORA-01473   **cannot have subqueries in CONNECT BY clause**

Cause: Subqueries cannot be used in a CONNECT BY clause.

Action: Remove the subquery or move it to the WHERE clause.

ORA-01474

ORA-01474   **cannot have START WITH or PRIOR without CONNECT BY**

Cause: START WITH and PRIOR are meaningful only in connection withCONNECT BY.

Action: Check the syntax for the SQL statement and add a CONNECT BY clause, if necessary.

ORA-01475

ORA-01475   **must reparse cursor to change bind variable datatype**

Cause: After executing a statement, an attempt was made to rebind a bind variable with a datatype different from that of the original bind.

Action: Reparse the cursor before rebinding with a different datatype.

ORA-01476

ORA-01476   divisor is equal to zero

Cause: An expression attempted to divide by zero.

Action: Correct the expression, then retry the operation.

ORA-01477

ORA-01477   **user data area descriptor is too large**

Cause: This is an internal error message not normally issued.

Action: Contact customer support.

ORA-01478

ORA-01478   array bind may not include any LONG columns

Cause: An attempt was made to use array bind on a column whose maximum size is greater than 2000 bytes. This is not permitted.

Action: Do not use array bind for a LONG column. Use an ordinary bind instead.

ORA-01479

ORA-01479   **last character in the buffer is not Null**

Cause: A bind variable of type 97 does not contain NULL at the last position.

Action: Make the last character a NULL.

ORA-01480

ORA-01480   **trailing null missing from STR bind value**

Cause: A bind variable specified as type SQLT_STR is not terminated with an ASCII NULL (0) character.

Action: Check maximum lengths and contents of string bind variables.

ORA-01481

ORA-01481   invalid number format model

Cause: An invalid format parameter was used with the TO_CHAR orTO_NUMBER function.

Action: Correct the syntax, then retry the operation.

ORA-01482

ORA-01482   unsupported character set

Cause: The second or third parameter to the CONVERT function is not a supported character set.

Action: Use one of the supported character sets.

ORA-01483

ORA-01483   **invalid length for DATE or NUMBER bind variable**

Cause: A bind variable of type DATE or NUMBER is too long.

Action: Check your Oracle7 operating system-specific documentation for the maximum allowable length.

ORA-01484

ORA-01484   **arrays can only be bound to PL/SQL statements**

Cause: You tried to bind an array to a non-PL/SQL statement.

Action: Rewrite the offending code being careful to bind arrays only toPL/SQL statements.

ORA-01485

ORA-01485   **compile bind length different from execute bind length**

Cause: You bound a buffer of type DTYVCS (VARCHAR with the two byte length in front) and at execute time the length in the first two bytes is more than the maximum buffer length, given in the bind call. The number of elements in the array and the current number of elements in the array cannot be more than the maximum size of the array.

Action: Ensure that the buffer size is sufficiently large to contain the array plus two bytes for the buffer length.

ORA-01486

ORA-01486   **size of array element is too large**

Cause: An attempt was made to bind a data value that was either too large for the datatype, for example, NUMBER, or was greater than 2000 bytes, for example, VARCHAR or LONG.

Action: Find a way to convert or truncate the data value so that its length is acceptable.

ORA-01487

ORA-01487   packed decimal number too large for supplied buffer

Cause: A conversion request cannot be performed because the buffer is too small to hold the result.

Action: Increase the size of the buffer.

ORA-01488

ORA-01488   invalid nibble or byte in the input data

Cause: A conversion request cannot be performed because a digit was invalid.

Action: Fix the number and retry.

ORA-01489

ORA-01489   result of string concatenation is too long

Cause: The result of a string concatenation was larger than the maximum length of a string (2000 characters).

Action: Reduce the size of one or both of the strings to be concatenated. Make certain the total length of the concatenation result is less than 2000 characters.

01490-01499: Miscellaneous, ANALYZE, SQL Parsing, Execution

This section lists miscellaneous messages generated by secondary processes called by the current process, such as the ANALYZE command, the SQL parser, and during the execution of Oracle commands.

ORA-01490

ORA-01490   invalid ANALYZE command

Cause: The syntax of the ANALYZE command was incorrect.

Action: Check the syntax and enter the command using the correct syntax.

ORA-01491

ORA-01491   CASCADE option not valid

Cause: The CASCADE option should be used only for tables or clusters.

Action: Do not use the CASCADE option in this manner. Check the syntax of the statement and then retry.

ORA-01492

ORA-01492 LIST option not valid

Cause: The LIST option can be used only for tables or clusters.

Action: Do not use the LIST option in this manner. Check the syntax of the statement and then retry.

ORA-01493

ORA-01493   **invalid SAMPLE size specified**

Cause: The specified SAMPLE size is out of range

Action: Specify a value within the proper range.

ORA-01494

ORA-01494   invalid SIZE specified

Cause: The specified histogram SIZE value was out of range.

Action: Specify a value within the proper range.

ORA-01495

ORA-01495 **specified chain row table not found**

Cause: The specified table does not exist, or the user does not have the proper privileges to access it.

Action: Specify an existing table or obtain the privileges to access the desired table.

ORA-01496

ORA-01496   specified chain row table form incorrect

Cause: The specified table does not have the proper field definitions.

Action: Check the spelling of the table name and specify the correct table to use.

ORA-01497

ORA-01497   illegal option for ANALYZE CLUSTER

Cause: The FOR COLUMNS <column list> clause cannot be used with ANALYZE CLUSTER.

Action: Retry with a legal syntax.

ORA-01498

ORA-01498 **block Check Failure - see trace file**

Cause: An error occurred while checking a block with the ANALYZE command.

Action: Check the trace file for more descriptive messages about the problem. Correct these errors. The name of the trace file is operating system-specific, for example, ORAxxxx.TRC. It is found in the directory specified by the initialization parameter USER_DUMP_DEST. If USER_DUMP_DEST is not set, trace files are not created. It may be necessary to recreate the object. See also your operating system-specific Oracle documentation.

ORA-01499

ORA-01499   **table/Index Cross Reference Failure - see trace file**

Cause: An error occurred when validating an index or a table using the ANALYZE command. One or more entries does not point to the appropriate cross-reference.

Action: Check the trace file for more descriptive messages about the problem. Correct these errors. The name of the trace file is operating system-specific, for example, ORAxxxx.TRC. It is found in the directory specified by the initialization parameter USER_DUMP_DEST. If USER_DUMP_DEST is not set, trace files are not created. It may be necessary to recreate the object. See also your operating system-specific Oracle documentation.

01500-01699: Oracle Commands

This section lists messages generated when entering Oracle commands.

ORA-01500

ORA-01500   **failure in getting date/time**

Cause: In executing a CREATE DATABASE or ALTER TABLESPACE statement, there was a failure in getting the date and time.

Action: Contact customer support.

ORA-01501

ORA-01501   CREATE DATABASE failed

Cause: An error occurred while executing the CREATE DATABASE statement.

Action: Refer to the diagnostic information in the accompanying message stack.

ORA-01502

ORA-01502   **index *str.name* is in direct load state**

Cause: The specified index was marked invalid by a direct load.

Action: Drop the specified index.

ORA-01503

ORA-01503 CREATE CONTROLFILE failed

Cause: An error occurred during CREATE CONTROLFILE.

Action: See accompanying messages.

ORA-01504

ORA-01504   **database name *name* does not match parameter DB_NAME *str***

Cause: The name in a CREATE DATABASE or START UP command does not match the name given in the initialization parameter DB_NAME.

Action: Determine which name is correct, then resubmit the command after correcting the name in the appropriate place.

ORA-01505

ORA-01505   **error in adding log files**

Cause: In executing a CREATE DATABASE or ALTER DATABASE statement, an error occurred when adding new redo log files. The user issuing the command must CONNECT INTERNAL. The database must be mounted but closed in order to add redo log files.

Action: Check the accompanying message stack for detailed information and ensure that the session is connected as INTERNAL with the database mounted but closed.

ORA-01506

ORA-01506	**missing or illegal database name**

Cause: No database name was specified in the ALTER DATABASE statement, nor was the initialization parameter DB_NAME specified.

Action: The database name must be given either in the ALTER DATABASE statement or as the value for the initialization parameter DB_NAME. Note that the SQL*DBA command START UP issues an ALTER DATABASE statement using the database name, if any, specified as part of START UP.

ORA-01507

ORA-01507   database not mounted

Cause: An ALTER DATABASE statement specified the name of a database to alter, but no database is currently mounted.

Action: If issuing an ALTER DATABASE command via the SQL*DBA START UP command, specify the MOUNT option. If directly issuing an ALTER DATABASE DISMOUNT command, do nothing. Otherwise, reissue the ALTER DATABASE command specifying the MOUNT option.

ORA-01508

ORA-01508   **cannot create database; error in file *name* at line *num***

Cause: A CREATE DATABASE statement was unable to process the named file because an error occurred at the given line number.

Action: Check the offending line in the specified file, correct the error, then re-execute the CREATE DATABASE statement. If the error is in an Oracle-supplied file, contact customer support.

ORA-01509

ORA-01509   **specified name *name* does not match actual *name***

Cause: The database name specified in an ALTER DATABASE statement does not match the name of the currently mounted database.

Action: Correct the database name spelling or DISMOUNT the mounted database.

ORA-01510

ORA-01510   error in deleting log files

Cause: In executing an ALTER DATABASE statement, an error occurred while dropping redo log files.

Action: Check the accompanying message stack for detailed information.

ORA-01511

ORA-01511   error in renaming log/datafiles

Cause: In executing an ALTER DATABASE statement, an error occurred while renaming log or datafiles.

Action: Check the accompanying message stack for detailed information.

ORA-01512

ORA-01512   **error renaming log file** *name* **- new file** *name* **not found**

Cause: An attempt to change a redo log file's name in the control file failed because no file was found with the new name.

Action: Check that the redo log file has been properly renamed by the operating system and retry.

ORA-01513

ORA-01513   **invalid current time returned by operating system**

Cause: The operating system returned a time that was not between the years 1988 and 2121.

Action: Correct the time kept by the operating system.

ORA-01514

ORA-01514   **error in log specification; no such log**

Cause: A redo log filename or list of member names did not correspond to an existing redo log file.

Action: Specify the name of an existing redo log file, check the spelling and capitalization of the names, and then re-execute the statement.

ORA-01515

ORA-01515   **error dropping log group *num*: no such log**

Cause: An attempt was made to drop a redo log file that does not exist.

Action: Specify the name of an existing redo log file, then issue the statement again.

ORA-01516

ORA-01516 **nonexistent log/datafile** *name*

Cause: An ALTER DATABASE statement is attempting to rename a log file or a datafile that is not known to the database control file.

Action: Specify the name of an existing redo log file, then re-execute the statement.

ORA-01517

ORA-01517   **log member:** *str*

Cause: This message indicates the filename involved with other messages.

Action: See the accompanying messages and take appropriate action.

ORA-01518

ORA-01518   **CREATE DATABASE must specify more than one log file**

Cause: Only one redo log file was specified in the CREATE DATABASE statement.

Action: Specify at least two redo log files, then re-execute the statement.

ORA-01519

ORA-01519   **error while processing file *name* near line *num***

Cause: A CREATE DATABASE statement encountered a problem while processing the named file, probably because of a system installation error.

Action: Try the system installation procedure again or contact customer support. See also your operating system-specific Oracle documentation.

ORA-01520

ORA-01520 **number of datafiles to add *num* exceeds limit of *num***

Cause: A CREATE TABLESPACE statement specifies more files than are permitted for this database.

Action: Use fewer files. To exceed the limit, recreate the database with a larger value of MAXDATAFILES.

ORA-01521

ORA-01521   error in adding datafiles

Cause: During a CREATE or ALTER TABLESPACE, an error was detected while adding datafiles.

Action: Check the accompanying message stack for detailed information.

ORA-01522

ORA-01522   **file *name* to be renamed does not exist**

Cause: During an ALTER TABLESPACE RENAME, a file to be renamed was not found in the database control file.

Action: Specify the correct filename, then re-execute the statement.

ORA-01523

ORA-01523   **cannot rename datafile to *name* - file already part of database**

Cause: During an ALTER DATABASE RENAME or ALTER TABLESPACE RENAME, the new name of a file is already present in the control file.

Action: Rename the file to a name not already in use as part of the database.

ORA-01524

ORA-01524 **cannot create datafile as *name* - file already part of database**

Cause: During an ALTER DATABASE CREATE DATAFILE, the new name of a file is already present in the control file.

Action: Use a name that is not already in use in the database.

ORA-01525

ORA-01525   **error in renaming datafiles**

Cause: An error occurred when renaming files as part of ALTER TABLESPACE.

Action: Check the accompanying message stack for detailed information. All files were renamed except those mentioned in the message stack.

ORA-01526

ORA-01526   **error in opening file** *name*

Cause: A CREATE DATABASE was not able to open the file identified in the message. This is probably due to a system installation error.

Action: Try the system installation procedure again or contact customer support. See also your operating system-specific Oracle documentation.

ORA-01527

ORA-01527   **error while reading file**

Cause: A CREATE DATABASE statement was not able to read the specified file, probably because of a system installation error.

Action: Try the system installation procedure again or contact customer support. See also your operating system-specific Oracle documentation.

ORA-01528

ORA-01528   **EOF while processing SQL statement**

Cause: A CREATE DATABASE statement unexpectedly encountered an end-of-file marker while reading the specified file, probably because of a system installation error.

Action: Try the system installation procedure again or contact customer support. See also your operating system-specific Oracle documentation.

ORA-01529

ORA-01529   **error closing file** *name*

Cause: A CREATE DATABASE was not able to close the file identified in the message.

Action: Try the system installation procedure again or contact customer support. See also your operating system-specific Oracle documentation.

ORA-01530

ORA-01530   **a database already mounted by the instance**

Cause: During an ALTER DATABASE MOUNT, an attempt was made to mount a database by an instance for which there is already a mounted database.

Action: To mount the database, shut down the instance, then start up the instance and try the operation again.

ORA-01531

ORA-01531   **a database already open by the instance**

Cause: During an ALTER DATABASE, an attempt was made to open a database by an instance for which there was already an open database.

Action: To open a new database, shut down the instance, then start up the instance and try the operation again.

ORA-01532

ORA-01532   **cannot create database; instance being started elsewhere**

Cause: During a CREATE DATABASE, another user appears to be simultaneously starting the instance.

Action: Retry the operation. If the error recurs, contact customer support.

ORA-01533

ORA-01533   **cannot rename file *name*; file does not belong to tablespace**

Cause: During an ALTER TABLESPACE RENAME, a file to be renamed was not found in the named tablespace.

Action: Check syntax and spelling and correctly specify the filename and tablespace name.

ORA-01534

ORA-01534 **rollback segment** *name* **doesn't exist**

Cause: During an ALTER or DROP ROLLBACK SEGMENT, the specified rollback segment name was not found.

Action: Correct syntax and spelling and correctly specify the rollback segment name or specify a different rollback segment.

ORA-01535

ORA-01535   rollback segment *name* already exists

Cause: During a CREATE ROLLBACK SEGMENT, the specified rollback segment was found to exist already.

Action: Correct syntax and spelling and correctly specify the rollback segment name or specify a different rollback segment.

ORA-01536

ORA-01536 **space quota exceeded for tablespace "*name*"**

Cause: The space quota in the tablespace is already used up and the operation attempted the creation of a new extent in the tablespace.

Action: Either

- Drop unnecessary objects in the tablespace to reclaim space.

- Ask the database administrator to increase the tablespace quota.

ORA-01537

ORA-01537   **cannot add datafile *name* - file already part of database**

Cause: During a CREATE or ALTER TABLESPACE, a file being added is already part of the database.

Action: Correct spelling of the datafile name or use a different file.

ORA-01538

ORA-01538   **failed to acquire any rollback segment**

Cause: An instance failed to acquire the SYSTEM rollback segment, or an instance failed to acquire one rollback segment in addition to the SYSTEM rollback segment during start up in parallel mode.

Action: Check that the SYSTEM rollback segment is available. Before starting an instance in parallel mode, either create one more public rollback segment or specify an available private rollback segment in the initialization parameter ROLLBACK_SEGMENTS.

ORA-01539

ORA-01539   **tablespace *"name"* is not online**

Cause: An attempt was made to bring a tablespace off line normally, but it is not online.

Action: Check the status of the tablespace. Use IMMEDIATE or TEMPORARY options to force all files off line. Also, the database must be in ARCHIVELOG mode.

ORA-01540

ORA-01540   **tablespace "*name*" is not offline**

Cause: A tablespace could not be brought online or made read-only because it is not offline.

Action: Check the status of the tablespace, then retry the operation. You can also use the options IMMEDIATE or TEMPORARY to force all files offline. If your intention is to make the file read-only, bring the tablespace online first.

ORA-01541

ORA-01541   **system tablespace cannot be brought offline; shut down if necessary**

Cause: An attempt was made to bring tablespace SYSTEM offline.

Action: Do not attempt to take the tablespace SYSTEM offline because it must always be online. Shut down if necessary to do a recovery.

ORA-01542

ORA-01542   **tablespace "*name*" is offline, cannot allocate space in it**

Cause: An attempt was made to allocate space in an offline tablespace.

Action: Bring the tablespace online or create the object in another tablespace.

ORA-01543

ORA-01543   **tablespace "*name*" already exists**

Cause: An attempt was made to create a tablespace that already exists.

When Trusted Oracle7 is configured in DBMS MAC mode, it is possible to get this error if you attempt to create a tablespace using a name that was previously used by another tablespace. In DBMS MAC mode, it is not possible to reuse tablespace names, even if the old tablespace has been dropped.

Action: Use a different name for the new tablespace.

ORA-01544

ORA-01544   **cannot drop system rollback segment**

Cause: An attempt was made to drop the rollback segment SYSTEM.

Action: No action required.

ORA-01545

| **ORA-01545** | **rollback segment #"*name*" was not available** |

Cause: Either

- An attempt was made to bring a rollback segment online that is unavailable during startup. For example, the rollback segment is in an offline tablespace.

- An attempt was made to bring a rollback segment online that is already online. This is because the rollback segment is specified twice in the ROLLBACK_SEGMENTS parameter in the initialization parameter file, or the rollback segment is already online by another instance.

- An attempt was made to drop a rollback segment that is currently online.

Action: Either:

- Make the rollback segment available; for example, bring an offline tablespace online.

- Remove the name from the ROLLBACK_SEGMENTS parameter if the name is a duplicate or if another instance has already acquired the rollback segment.

- Bring the rollback segment offline, which may require waiting for the current transaction to finish, or, if the rollback segment needs recovery, discover which errors are holding up the rolling back of the transactions and take appropriate actions.

ORA-01546

ORA-01546   **tablespace contains active rollback segment "*name*"**

Cause: You tried to bring offline or make read-only a tablespace that contains active rollback segments.

Action: Shut down the instances that use the active rollback segments in the tablespace. Then bring it offline or make it read-only.

ORA-01548

ORA-01548   **active rollback segment "*name*" found, terminate dropping tablespace**

Cause: An attempt was made to drop a tablespace that contains active rollback segments.

Action: Shut down instances that use the active rollback segments in the tablespace and then drop the tablespace.

ORA-01549

ORA-01549   **tablespace not empty, use INCLUDING CONTENTS option**

Cause: An attempt was made to drop a non-empty tablespace.

Action: To drop all the objects in the tablespace, use the INCLUDING CONTENTS option with DROP TABLESPACE.

ORA-01550

ORA-01550   **cannot drop system tablespace**

Cause: An attempt was made to drop the tablespace SYSTEM.

Action: No action required.

ORA-01551

ORA-01551   **extended rollback segment, pinned blocks released**

Cause: This is an internal error message not normally issued.

Action: Contact customer support.

ORA-01552

ORA-01552   **cannot use system rollback segment for non-system tablespace "*name*"**

Cause: An attempt was made to use the SYSTEM rollback segment for operations involving a non-system tablespace.

Action: Create one or more rollback segments and then use ALTER ROLLBACK SEGMENT 'name' online. It may be necessary to modify the initialization parameter ROLLBACK_SEGMENTS to acquire one of the new private rollback segments. Before creating a rollback segment outside the SYSTEM tablespace, it is necessary to first create and activate a non-system rollback segment in the SYSTEM tablespace.

ORA-01553

ORA-01553   **MAXEXTENTS must be no smaller than the *num* extents currently allocated**

Cause: The number of extents allocated is greater than the MAXEXTENTS specified.

Action: Specify a larger MAXEXTENTS value.

ORA-01554

ORA-01554   **out of transaction slots in transaction tables**

Cause: There were too many concurrent transactions.

Action: Shut down Oracle, modify initialization parameters TRANSACTIONS and ROLLBACK_SEGMENTS, then restart Oracle. The TRANSACTIONS parameter must be high enough to handle any possible rollbacks.

ORA-01555

ORA-01555   snapshot too old (rollback segment too small)

Cause: One of the following:

- **Insufficient Rollback Segments**: A long running query may not be able to reconstruct the snapshot of the blocks it is reading because the rollback data is not available. This can happen when the database has many transactions that are changing data, then committing or rolling back. The rollback data can be overwritten if the rollback segments are too small for the number and size of the changes being made.

- **Corrupted Rollback Segment:** If the rollback segment is corrupt and cannot be read, a statement needing to reconstruct a before-image snapshot causes this message.

- **Trusted Oracle - Too Few Checkpoints:** When Trusted Oracle7 is configured in OS MAC mode, it is possible to get this error if the interval between checkpoints in a secondary database is too small, causing necessary rollback records to be overwritten.

- **Precompiler - Insufficient Rollback Segments:** A long running query may not be able to reconstruct the snapshot of the blocks it is reading because the rollback data is not available. This can happen if your program does not CLOSE a cursor after repeated FETCH or UPDATE statements. Alternatively, if a FETCH is executed after a COMMIT, the number of rollback records created since the last CLOSE of the current cursor will fill the available rollback segments and begin to overwrite earlier records.

Action: For the above causes:

- **Insufficient Rollback Segments;** Make a larger number of bigger rollback segments available. This will allow the rollback data for completed transactions to be kept longer.

- **Corrupted Rollback Segment:** Contact customer support and detailany other messages issued at the same time that indicate which block is corrupt.

- **Trusted Oracle - Too Few Checkpoints:** For Trusted Oracle7 users, if the cause of this error is that the necessary rollback records in an OS MAC secondary database were overwritten, decrease the value of the LOG_CHECKPOINT_TIMEOUT initialization parameter so that checkpoints are taken more frequently.

- **Precompiler - Insufficient Rollback Segments:** If you are not using the *fetch across commit* feature, ensure that you have CLOSEd cursors appropriately. Otherwise, you must increase the number and the size of rollback segments. You can estimate the size of the rollback data that your program will produce with the V$ROLLSTAT view, which contains the number of bytes written for each rollback segment. Set the current transaction to a rollback segment in a single user mode and query V$ROLLSTAT before and after the transaction. The difference in V$ROLLSTAT gives the number of rollback data bytes written for the transaction. Estimate the total number of transactions done in the loop and, after considering other concurrent transactions, create rollback segments accordingly.

Note: *Fetch across commit* is not supported by the ANSI standard. According to the ANSI standard, a cursor is invalidated when a commit is performed and should be closed and reopened. With Oracle you can *fetch across commit*. However, you should be aware that you may get this error.

ORA-01556

ORA-01556 **MINEXTENTS for rollback segment must be greater than 1**

Cause: A MINEXTENTS of less than two was specified for rollback segment.

Action: Specify a larger MINEXTENTS.

ORA-01557

ORA-01557   **rollback segment extents must be at least 3 blocks**

Cause: An extent of less than three blocks was specified for the rollback segment.

Action: Specify extents of at least three blocks, then retry the operation.

ORA-01558

ORA-01558   **out of transaction ID's in rollback segment** *num*

Cause: All the available transaction IDs have been used.

Action: Shut down the instance, restart it using another rollback segment, then drop the rollback segment that has no more transaction IDs.

ORA-01559

ORA-01559   MAXEXTENTS for rollback segment must be greater than 1

Cause: An invalid value for MAXEXTENTS was specified; the value must be greater than 1.

Action: Specify a MAXEXTENTS value of at least 2 and try again.

ORA-01560

ORA-01560 **global hash table size mismatch for GC_*name* (*num* != *num*)**

Cause: This message is only relevant for systems running the Parallel Server and occurs at instance start up. The value of the GC_*name* initialization parameter does not match that of an instance already mounted.

Action: Ensure correct initialization parameter files are being used. If necessary, modify the GC_*name* parameter in the offending initialization files, then shut down and restart those instances with corrected initialization parameter files.

ORA-01561

ORA-01561   failed to remove all objects in the tablespace specified

Cause: A user failed to remove all objects when dropping a tablespace.

Action: Repeat DROP TABLESPACE until all objects have been dropped.

ORA-01562

ORA-01562 **failed to extend rollback segment ID =** *num*

Cause: A failure occurred while trying to extend the rollback segment. The problem is usually lack of space in the database to extend the rollback segment.

Action: This is usually followed by another message stating the cause of the failure. Shut down and take appropriate action for the error that caused the failure. If no other messages follow this message, the cause is probably lack of space in the database.

ORA-01563

ORA-01563   **rollback segment is PUBLIC, need to use the keyword PUBLIC**

Cause: The keyword PUBLIC was not used to identify a public rollback segment.

Action: Use the keyword PUBLIC when identifying a public rollback segment.

ORA-01564

ORA-01564   rollback segment is not PUBLIC

Cause: The identified rollback segment is not public.

Action: Do not use the keyword PUBLIC when identifying a private rollback segment.

ORA-01565

ORA-01565 **error in identifying file** *name*

Cause: An error occurred while trying to identify a file.

Action: Check the accompanying message stack for detailed information.

ORA-01566

ORA-01566   **file specified more than once in DROP LOGFILE**

Cause: For an ALTER DATABASE, the list of files for the DROP LOGFILE option contains at least one duplicate.

Action: Remove the duplicate file specification, then re-execute the statement.

ORA-01567

ORA-01567   **dropping log *name* would leave less than two log files in thread *num***

Cause: Dropping all the files specified would leave fewer than the required two redo log files. It may be possible to clear the log rather than drop it.

Action: Either drop fewer logs or disable the thread before deleting the logs.

ORA-01568

ORA-01568   cannot set space quota on PUBLIC

Cause: An attempt was made to set a space quota for PUBLIC on a tablespace.

Action: To grant system-wide or tablespace-wide space privileges to all users, use the statement GRANT UNLIMITED TABLESPACE [ON *tablespace*] TO PUBLIC;

ORA-01569

ORA-01569   **datafile too small for system dictionary tables**

Cause: The datafile specified during creation of the database is too small to hold the system dictionary tables.

Action: Re-create the database specifying a larger file or more files.

ORA-01570

ORA-01570 **MINEXTENTS must be no larger than the *num* extents currently allocated**

Cause: The number of extents already allocated is smaller than the MINEXTENTS specified in the command.

Action: Enter the command again specifying the number of extents given in the message as the value for MINEXTENTS.

ORA-01571

ORA-01571   redo version *num* incompatible with Oracle Version *num*

Cause: This software version cannot read the current redo logs. Either crash recovery is required or there are offline database files that need media recovery. If a filename is listed then it needs media recovery.

Action: Shut down and start up using the compatible software. Do any requiredmedia recovery and open the database. Shut down and then start up using current software.

ORA-01572

ORA-01572   **global hash table size *num* for rollback segments is too small for rollback segment ID *num***

Cause: An operation requested a resource that was unavailable. On a Parallel Server system, the maximum number of systemwide rollback segments is specified by the GC_ROLLBACK_SEGMENTS parameter. When this maximum is reached, no more requests are processed.

Action: Use fewer rollback segments or increase the initialization parameter GC_ROLLBACK_SEGMENTS to a number greater than rollback segment ID *num* in every initialization parameter file of the Parallel Server. This change will not take effect until Oracle is shut down and restarted.

ORA-01573

ORA-01573   **shutting down instance, no further change allowed**

Cause: A process attempted to make changes while the database was being shut down.

Action: No action required.

ORA-01574

ORA-01574   **maximum number of concurrent transactions exceeded**

Cause: An operation requested a resource that was unavailable. The maximum number of concurrent transactions is specified by the TRANSACTIONS initialization parameter. When this maximum is reached, no more requestsare processed.

Action: Try the request again later or increase the TRANSACTIONS parameter in the initialization parameter file. This change will not take effect until Oracle is shut down and restarted.

ORA-01575

ORA-01575   **time-out waiting for space management resource**

Cause: The system timed out before it could acquire the necessary resources to do space management.

Action: Wait a few minutes, then retry the operation.

ORA-01576

ORA-01576   **instance locking protocol version *num* incompatible with OracleVersion *num***

Cause: The version of Oracle is incompatible with the given locking protocol version.

Action: Upgrade the version of Oracle used to start up instances so that they use a compatible locking protocol.

ORA-01577

ORA-01577   **cannot add log file *name* - file already part of database**

Cause: While executing a CREATE or ALTER DATABASE statement, a file being added was found to already exist on the database.

Action: Check syntax and spelling, specify a valid filename, and try again.

ORA-01578

ORA-01578   **Oracle data block corrupted (file # *num*, block # *num*)**

Cause: The given data block was corrupted, probably due to program errors.

Action: Try to restore the segment containing the given data block. This may involve dropping the segment and recreating it. If there is a trace file, report the messages recorded in it to customer support.

ORA-01579

ORA-01579   **write error occurred during recovery**

Cause: A write error occurred during media recovery.

Action: Check the trace file for the type of the write error, then take appropriate action.

ORA-01580

ORA-01580   **error creating control backup file** *name*

Cause: An operating system error occurred while attempting to create the referenced control file for backup.

Action: Check the accompanying message stack for detailed information, then take appropriate action.

ORA-01581

ORA-01581   **attempt to use rollback segment *name* new extent *name* which isbeing allocated**

Cause: Undo generated to extend a rollback segment run out of current undo block space and is attempting to write into the new extent, which has not been completely allocated.

Action: The rollback segment extending itself will be rollbacked by the system. No more extension will be possible until the next extent is freed up by the rolling back or committing of other transactions.

ORA-01582

ORA-01582 **unable to open control file for backup**

Cause: An operating system error occurred while attempting to open a control filefor backup.

Action: Check the accompanying message stack for detailed information, then take appropriate action.

ORA-01583

ORA-01583   unable to get block size of control file to be backed up

Cause: An operating system error occurred while attempting to get the block size of a control file for backup.

Action: Check the accompanying message stack for detailed information, then take appropriate action.

ORA-01584

ORA-01584   **unable to get file size of control file to be backed up**

Cause: An operating system error occurred while attempting to get the file size of a control file for backup.

Action: Check the accompanying message stack for detailed information, then take appropriate action.

ORA-01585

ORA-01585   **error identifying backup file *name***

Cause: An operating system error occurred when attempting to identify the referenced control file for backup.

Action: Check the accompanying message stack for detailed information, then take appropriate action.

ORA-01586

ORA-01586 **unable to open destination file *name* for backup**

Cause: An operating system error occurred while attempting to open a control file for backup.

Action: Check the accompanying message stack for detailed information, then take appropriate action.

ORA-01587

ORA-01587   **error during control file backup file copy**

Cause: An operating system error occurred while attempting to copy a control file for backup.

Action: Check the accompanying message stack for detailed information, then take appropriate action.

ORA-01588

ORA-01588   must use RESETLOGS option for database open

Cause: An earlier attempt to open the database with the RESETLOGS option did not complete, or recovery was done with a control file backup.

Action: Retry the operation making sure to specify the RESETLOGS option.

ORA-01589

ORA-01589   must use RESETLOGS or NORESETLOGS option for database open

Cause: Either cancel-based or time-based recovery has been performed. After these types of recovery, it is necessary to specify either the RESETLOGS or NORESETLOGS option to open the database.

Action: Specify the appropriate option.

ORA-01590

ORA-01590   **number of segment free list *num* exceeds maximum of *num***

Cause: The value of the storage clause FREELIST GROUPS is greater than the maximum allowed for the block size.

Action: Specify a number for FREELIST GROUPS less than or equal to the maximum when creating the table or index. The block size specified in the initialization parameter DB_BLOCK_SIZE cannot be changed after the database is created.

ORA-01591

ORA-01591   **lock held by in-doubt distributed transaction** *num*

Cause: An attempt was made to access a resource locked by a dead two-phase commit transaction that is in prepared state.

Action: Match the transaction number in the message with the GLOBAL_TRAN_ID column of the DBA_2PC_PENDING table to determine the database link and the state of the transaction. Attempt to repair network connections to the coordinator and commit point, if necessary. If timely repair is not possible, contact the database administrator at the commit point, if known, to resolve the pending transaction.

ORA-01592

ORA-01592   **error converting Version 6 rollback segment "*num*" to Oracle7 format**

Cause: Oracle encountered an error while converting a Version 6 rollback segment into the format used by Oracle7.

Action: Investigate the accompanying internal error message. The Version 6 database may not have shut down cleanly. It may be necessary to reload the Version 6 database from backup and shut it down, ensuring it shuts down cleanly.

ORA-01593

ORA-01593   **rollback segment optimal size *num* blocks is smaller than the computed initial size *num* blocks**

Cause: The specified OPTIMAL size is smaller than the cumulative size of the initial extents during create rollback segment.

Action: Specify a larger OPTIMAL size.

ORA-01594

ORA-01594   **attempt to wrap into rollback segment *name* extent *num* which is being freed**

Cause: Undo generated to free a rollback segment extent is attempting to write into the same extent due to small extents or too many extents to free.

Action: Increase the optimal size of the rollback segment. The rollback segment shrinking will be rolled back by the system.

ORA-01595

ORA-01595   **error freeing extent *num* of rollback segment *name***

Cause: An error occurred while freeing inactive rollback segment extents.

Action: Check the trace file and check for prior messages.

ORA-01596

ORA-01596   **fail to coalesce extents because sort area size is too small**

Cause: There are too many entries of free extents to sort in the in-memory sort area.

Action: Increase the sort area size or reduce the fragmentation in the tablespace by doing a full export followed by an import.

ORA-01597

ORA-01597   **cannot alter system rollback segment online or offline**

Cause: An attempt was made to online or offline the SYSTEM rollback segment.

Action: No action required.

ORA-01598

ORA-01598   rollback segment "*name*" is not online

Cause: The rollback segment was taken offline either manually or by SMON.

Action: Check the status of the rollback segment in DBA_ROLLBACK_SEGS.

ORA-01599

ORA-01599   **failed to acquire rollback segment "*name*", cache space is full (currently has *num* entries)**

Cause: The amount of allocated space is not enough.

Action: Take the rollback segment offline.

ORA-01600

ORA-01600   **at most one *str* in clause *str* of GC_FILES_TO_LOCKS**

Cause: The initialization parameter GC_FILES_TO_LOCKS was not specified properly.

Action: Refer to the proper specifications of the GC_FILES_TO_LOCKS parameter and specify it correctly in the initialization parameter file. If other instances are currently accessing the database, use the value that they are using.

ORA-01601

ORA-01601 **illegal bucket size in clause *str* of GC_FILES_TO_LOCKS**

Cause: The number of locks specified in a clause of the initialization parameter GC_FILES_TO_LOCKS is not a positive integer.

Action: Refer to the proper specifications of the GC_FILES_TO_LOCKS parameter and specify it correctly in the initialization parameter file. If other instances are currently accessing the database, use the value that they are using.

ORA-01602

ORA-01602   **more locks in GC_FILES_TO_LOCKS than reserved in GC_DB_LOCKS**

Cause: The total number of PCM locks specified to all database files in the initialization parameter GC_FILES_TO_LOCKS is greater than the number specified in GC_DB_BLOCKS.

Action: Correct the value of either the GC_FILES_TO_LOCKS parameter or the GC_DB_LOCKS parameter before starting the instance. If other instances are currently accessing the database, use the values that they are using.

ORA-01603

ORA-01603 **illegal grouping in clause *str* of GC_FILES_TO_LOCKS**

Cause: The initialization parameter GC_FILES_TO_LOCKS is specified incorrectly; if a clause specifies more than one range of file numbers, each range must be separated with commas.

Action: Correct the value of the GC_FILES_TO_LOCKS parameter before starting the instance. If other instances are currently accessing the database, use the value they are using. Use the following syntax:

GC_FILES_TO_LOCKS = "{*filenum*[-*filenum*]
_[,*filenum*[-*filenum*]]...= *num_of_locks*[EACH]}[:]..."

where *filenum* is a file number, as listed in V$DBFILE, and *num_of_locks* is the number of PCM locks.

ORA-01604

ORA-01604 **illegal file number range in clause *str* of GC_FILES_TO_LOCKS**

Cause: The range of file numbers specified in a clause of the initialization parameter GC_FILES_TO_LOCKS is invalid. The first file number must be smaller than the second file number, and a hyphen must separate the numbers. All numbers in the range must be valid file numbers.

Action: Correct the value of the GC_FILES_TO_LOCKS parameter before starting the instance. If other instances are currently accessing the database, use the value that they are using.

ORA-01605

ORA-01605 **missing file numbers in clause *str* of GC_FILES_TO_LOCKS**

Cause: The initialization parameter GC_FILES_TO_LOCKS is specified incorrectly. Each clause should have one or more file numbers or ranges of file numbers, as listen in V$DBFILE, separated by commas. Use the following syntax:

GC_FILES_TO_LOCKS = "{*filenum*[-*filenum*]
_[,*filenum*[-*filenum*]]...= *num_of_locks*[EACH]}[:]..."

Spaces are not allowed within the double quotation marks.

Action: Correct the value of the GC_FILES_TO_LOCKS parameter before starting the instance. If other instances are currently accessing the database, use the value that they are using.

ORA-01606

ORA-01606   **GC_FILES_TO_LOCKS not identical to that of another mounted instance**

Cause: The initialization parameter GC_FILES_TO_LOCKS is not the same as another instance mounted in parallel mode. This parameter must be the same as that for all shared instances.

Action: Modify the parameter to be compatible with the other instances, then shut down and restart the instance.

ORA-01607

ORA-01607   **GC_LCK_PROCS** *num* **is not the same as other instances** *num*

Cause: The initialization parameter GC_LCK_PROCS is not the same in other instances that have the database open.

Action: Modify the parameter to be compatible with the other instances, then shut down and restart the instance.

ORA-01608

ORA-01608   cannot bring rollback segment "*name*" online, its status is *status*

Cause: The rollback segment may have been brought online previously by the database administrator or could have been left online after a process crashed. The above status can be one of the following: INVALID, IN USE, AVAILABLE, OFFLINE, NEEDS RECOVERY.

Action: Check the status of the rollback segment in the view DBA_ROLLBACK_SEGS. Take the segment offline, if necessary.

ORA-01609

ORA-01609 **log *name* is the current log for thread *num* - cannot drop members**

Cause: A member of the current redo log for the given thread cannot be dropped.

Action: If the thread is opened, request a log switch by the instance that is using the thread. If the thread is not open, disable the thread manually, archive the log or clear it.

ORA-01610

ORA-01610   **recovery using the BACKUP CONTROLFILE option must be done**

Cause: Either an earlier database recovery session specified BACKUP CONTROLFILE or the control file was recreated with the RESETLOGS option. Only BACKUP CONTROLFILE recovery is allowed, and it must be followed by resetting the online redo log files when the database is next opened.

Action: Perform recovery using the BACKUP CONTROLFILE option.

ORA-01611

ORA-01611   **thread number *num* is invalid - must be between 1 and *num***

Cause: A thread number in a command is greater than the number of threads supported by the control file.

Action: Use a thread number that is valid.

ORA-01612

ORA-01612   **thread *num* is already enabled**

Cause: An attempt was made to enable a thread that is already enabled.

Action: No action is required or enable another thread, if desired.

ORA-01613

ORA-01613   **thread *num* only has *num* logs - at least 2 logs required to enable**

Cause: The thread cannot be enabled because it has less than two online redo log files associated with it.

Action: Add logs to the thread or pick another thread to enable.

ORA-01614

ORA-01614   **thread *num* is busy - cannot enable**

Cause: The mount enqueue for the thread could not be acquired when attempting to enable the thread. This probably means that another process has already started enabling this thread.

Action: Wait and try again or find another thread to enable.

ORA-01615

ORA-01615   **thread *num* is mounted - cannot disable**

Cause: Some instance, possibly this one, has allocated the thread for use. The thread may not be disabled while in use.

Action: To disable this thread, cleanly shut down the instance using it.

ORA-01616

ORA-01616 **thread** *num* **is open - cannot disable**

Cause: The thread is not closed. The last instance to use the thread crashed and left the thread open. A thread cannot be disabled until it is closed. The thread is still needed for crash or instance recovery.

Action: If the database is open, instance recovery should close the thread soon. Wait a few minutes until instance recovery frees the thread. Otherwise, open the database. Crash recovery will close the thread.

ORA-01617

ORA-01617   **cannot mount: *num* is not a valid thread number**

Cause: The initialization parameter THREAD is not between 1 and the maximum number of threads allowed when the database was created.

Action: Shut down the instance, change the initialization parameter to a valid value, then restart the instance.

ORA-01618

ORA-01618 **thread *num* is not enabled - cannot mount**

Cause: The initialization parameter THREAD requests a thread that is not enabled. A thread must be enabled before it can be mounted.

Action: Shut down the instance, change the initialization parameter to an enabled thread number, then restart the instance. If the database is open in another instance, then the thread may be enabled.

ORA-01619

ORA-01619   **thread *num* is mounted by another instance**

Cause: The initialization parameter THREAD requests a thread that has been mounted by another instance. Only one instance may use each thread.

Action: Shut down the instance, change the initialization parameter to an unused thread, then restart the instance.

ORA-01620

ORA-01620   **no public threads are available for mounting**

Cause: The initialization parameter THREAD is zero, its default value. There are no available threads that have been publicly enabled.

Action: Shut down the instance, change the initialization parameter to a thread that is privately enabled and not mounted, then restart the instance. If the database is open in another instance, then a thread may be publicly enabled.

ORA-01621

ORA-01621 **cannot rename member of current log if database is open**

Cause: This is a rename command for a member of the current log for an open thread. If the database is open anywhere, the log may be in use, so the rename cannot be done.

Action: Wait until the log is not current or mount the database exclusively.

ORA-01622

ORA-01622   **thread number must be specified - default not specific**

Cause: The thread was not specified when adding a log, and the currently mounted thread was chosen by default. Because the current thread was not specified explicitly, the user cannot know to which thread the log will be added.

Action: Explicitly specify the thread number either in the initialization parameter THREAD or in the ADD LOGFILE command.

ORA-01623

ORA-01623 **log *num* is current log for thread *num* - cannot drop**

Cause: A thread's current log cannot be dropped, even if the thread is closed. A disabled thread usually does not have a current log, but a half completed disable may need to be disabled again.

Action: If the database is not open, disable the thread. If the database is open and an instance has the thread open, then switch logs in the instance with the thread open.

ORA-01624

ORA-01624 **log *name* needed for crash recovery of thread *num* - cannot drop**

Cause: A log cannot be dropped or cleared until the thread's checkpoint has advanced out of the log.

Action: If the database is not open, then open it. Crash recovery will advance the checkpoint. If the database is open, force a global checkpoint. If the log is corrupted so that the database cannot be opened, it may be necessary to do an incomplete recovery until this log is cancelled.

ORA-01625

ORA-01625 **rollback segment *name* does not belong to this instance**

Cause: An attempt was made to take a rollback segment offline that does not belong to this instance.

Action: Take only those rollback segments offline that belong to the current instance. To take the rollback segment offline, connect to the instance containing it.

ORA-01626

ORA-01626   **rollback segment number *num* cannot handle more transactions**

Cause: There are too many transactions in this segment.

Action: Choose a different rollback segment or reduce the number of concurrent transactions.

ORA-01627

ORA-01627   **rollback segment number *num* is not online**

Cause: This rollback may have been taken offline by the database administrator or cleaned up by SMON.

Action: Check the status of the rollback segment in DBA_ROLLBACK_SEGMENTS to make sure the rollback segment is actually online.

ORA-01628

ORA-01628 **max # of extents *num* reached for rollback segment *num***

Cause: An attempt was made to extend a rollback segment that already has reached its maximum size or space could not be allocated in the data dictionary to contain the definition of the object.

Action: If possible, increase the value of either the MAXEXTENTS or PCTINCREASE initialization parameters or find the data dictionary table lacking space and alter the storage parameters, as described in the *Oracle7 Server Administrator's Guide.*

ORA-01629

ORA-01629   **max # of extents *num* reached saving undo for tablespace *name***

Cause: The maximum amount of space allowed for saving undo entries has been reached for an offline tablespace.

Action: Check the storage parameters for the SYSTEM tablespace. The tablespace named needs to be brought online so that the undo information can be applied.

ORA-01630

ORA-01630   **max # of extents *num* reached in temp segment in tablespace *name***

Cause: The maximum amount of space allowed for saving undo entries has been reached for a temporary segment in the named tablespace, or space could not be allocated in the data dictionary to contain the definition of the object.

Action: If possible, increase the value of either the MAXEXTENTS or PCTINCREASE initialization parameters or find the data dictionary table lacking space and alter the storage parameters as described in the *Oracle7 Server Administrator's Guide.*

ORA-01631

ORA-01631   **max # of extents *num* reached in table *name***

Cause: The maximum amount of space allowed for saving undo entries has been reached for the named table.

Action: If possible, increase the value of either the MAXEXTENTS or PCTINCREASE initialization parameters.

ORA-01632

ORA-01632   **max # of extents *num* reached in index *name***

Cause: The maximum amount of space allowed for saving undo entries has been reached for the named index.

Action: If possible, increase the value of either the MAXEXTENTS or PCTINCREASE initialization parameters.

ORA-01633

ORA-01633   **Parallel Server option needed for this operation**

Cause: The system is not configured to use the Parallel Server option, so the feature is not available.

Action: Contact Oracle Corporation to obtain the Parallel Server option.

ORA-01634

ORA-01634 **rollback segment number *str* is about to go offline**

Cause: The rollback segment specified was marked to go offline by the DBA.

Action: Bring the rollback segment online before continuing, or choose another rollback segment. If using SET TRANSACTION, use ROLLBACK SEGMENT 'name'.

ORA-01635

ORA-01635   **rollback segment #*num* specified not available**

Cause: Either

- You tried to acquire an offline rollback segment during start up.
- You tried to drop a rollback segment that contains active transactions.

Action: For the above, either:

- Bring the tablespace containing the rollback segment online or do not specify the rollback segment in the initialization parameter file.
- If the rollback segment is in use, shut down the instance using the rollback segment. Otherwise, if the rollback segment needs recovery, find out the errors that are holding back the rolling back of the transactions and take appropriate actions.

ORA-01636

ORA-01636 **rollback segment "*name*" is already online**

Cause: A rollback segment can only be used by one instance and an instance is trying to bring a rollback segment online that is already in use.

Action: Check that the values set in the initialization parameter file for parameters ROLLBACK_SEGMENTS, ROLLBACK_SEGMENT_INITIAL, and ROLLBACK_SEGMENT_COUNT are correctly set for the instance with the problem. Also check that the instance is using the correct initialization parameter file. Make sure you are not confused about the difference between private and public rollback segments. See the *Oracle7 Server Administrator's Guide* for more information about using rollback segments in parallel mode.

ORA-01637

ORA-01637 **rollback segment "*name*" is being used by another instance #*name***

Cause: A rollback segment can only be used by one instance, and an instance is trying to access a rollback segment online that is already in use.

Action: Check that the values set in the initialization parameter file for parameters ROLLBACK_SEGMENTS, ROLLBACK_SEGMENT_INITIAL, and ROLLBACK_SEGMENT_COUNT are correctly set for the instance with the problem. Also check that the instance is using the correct initialization parameter file. Make sure you are not confused about the difference between private and public rollback segments. See *Oracle7 Parallel Server* for more information about using rollback segments in parallel mode.

ORA-01638

ORA-01638 **parameter** *name* **does not allow Oracle Version** *num* **to mount parallel**

Cause: The recovery compatible initialization parameter is set too low to allow this software version to do a parallel mount.

Action: Either use an earlier software release or advance the RECOVERY_COMPATIBLE parameter. If this happens when no RECOVERY_COMPATIBLE parameter has been specified, then set it to the current software release.

ORA-01639

ORA-01639 database cannot be mounted parallel with no lock processes

Cause: The value of GC_LCK_PROCS is 0, and the database is being mounted in parallel mode.

Action: Change GC_LCK_PROCS to a value between 1 and 10 or leave it undefined to use the default.

ORA-01640

ORA-01640	**cannot make tablespace read-only with active transactions**

Cause: An attempt to make a tablespace read-only encountered active transactions in the database. All transactions must be committed or rolled back to ensure consistency. This includes any transactions that are in doubt.

Action: Put the database in restricted mode to prevent any new transactions from being started and commit and/or rollback all active transactions, resolving all in-doubt transactions.

ORA-01641

ORA-01641   **datafiles cannot be added to read-only tablespace** *name*

Cause: An attempt was made to add datafiles to a read-only tablespace.

Action: Make the tablespace read-write before attempting to add datafiles.

ORA-01642

ORA-01642   **begin backup not needed for read-only tablespace** *name*

Cause: An attempt was made to use the BEGIN BACKUP or END BACKUP statement for a tablespace that has been made read-only.

Action: Start the backup without using the BEGIN BACKUP or END BACKUP statement. Read-only files cannot be modified and therefore will be consistent.

ORA-01643

ORA-01643   **system tablespace cannot be made read-only**

Cause: An attempt was made to make the system tablespace read-only. The system tablespace must be read-write for database operations.

Action: Do not attempt this operation.

ORA-01644

ORA-01644   **tablespace *name* is already read-only**

Cause: An attempt was made to make read-only a tablespace that is already read-only.

Action: This operation is unnecessary.

ORA-01645

ORA-01645	**previous attempt to make read-write is half complete**

Cause: A failure while making a tablespace read-write occurred leaving the tablespace read-only, however the checkpoint was advanced. The tablespace will not be usable after a RESETLOGS if it's files are offline.

Action: Attempt to make the tablespace read-write again.

ORA-01646

ORA-01646   **tablespace *name* is not read-only - cannot make read-write**

Cause: An attempt was made to make read-write a tablespace that is already read-write.

Action: This operation is unnecessary.

ORA-01647

ORA-01647   **tablespace "*name*" is read-only, cannot allocate space in it**

Cause: An attempt was made to allocate space in a read-only tablespace, which isnot possible.

Action: Create the object in a read-write tablespace.

ORA-01648

ORA-01648 **log *name* is the current log of disabled thread *name***

Cause: An attempt to enable the thread failed after it was half completed. This log was left as the current log even though the thread is still disabled. Because a log switch cannot be done until the thread is enabled, the log cannot be cleared or archived.

Action: Complete the thread enable by issuing the enable command again.

ORA-01650

ORA-01650   **unable to extend rollback segment** *name* **by** *num* **in tablespace** *name*

Cause: Failed to allocate extent for the rollback segment in tablespace.

Action: Use the ALTER TABLESPACE ADD DATAFILE statement to add one or more files to the specified tablespace.

ORA-01651

ORA-01651 **unable to extend save undo segment by *num* in tablespace *name***

Cause: Failed to allocate extent for saving undo entries for the specifiedoffline tablespace.

Action: Check the storage parameters for the system tablespace. The tablespace must be brought back online so the undo can be applied.

ORA-01652

| ORA-01652 | unable to extend temp segment by *num* in tablespace *name* |

Cause: Failed to allocate extent for temp segment in tablespace.

Action: Use the ALTER TABLESPACE ADD DATAFILE statement to add one or more files to the specified tablespace or create the object in another tablespace.

ORA-01653

ORA-01653   **unable to extend table *name.name* by *num* in tablespace *name***

Cause: Failed to allocate extent for table segment in tablespace.

Action: Use the ALTER TABLESPACE ADD DATAFILE statement to add one or more files to the specified tablespace.

ORA-01654

ORA-01654 **unable to extend index *name.name* by *num* for tablespace *name***

Cause: Failed to allocate extent for index segment in tablespace.

Action: Use the ALTER TABLESPACE ADD DATAFILE statement to add one or more files to the specified tablespace.

ORA-01655

ORA-01655   **unable to extend cluster *name.nam*e by *num* for tablespace *name***

Cause: Failed to allocate extent for cluster segment in tablespace.

Action: Use the ALTER TABLESPACE ADD DATAFILE statement to add one or more files to the specified tablespace.

ORA-01656

ORA-01656   **max *num* extents *num* reached in cluster *name.name***

Cause: A cluster tried to extend past the value specified for MAXEXTENTS.

Action: If MAXEXTENTS is less than the system maximum, specify a higher value. Otherwise, you must recreate with larger INITIAL, NEXT, orPCTINCREASE parameters.

ORA-01658

ORA-01658 **unable to create INITIAL extent for segment in tablespace** *name*

Cause: Failed to find sufficient contiguous space to allocate INITIAL extent for segment being created.

Action: Use ALTER TABLESPACE ADD DATAFILE to add additional space to the tablespace or retry with a smaller value for INITIAL.

ORA-01659

ORA-01659   **unable to allocate MINEXTENTS beyond *size* in tablespace *name***

Cause: Failed to find sufficient contiguous space to allocate MINEXTENTS for the segment being created.

Action: Use ALTER TABLESPACE ADD DATAFILE to add additional space to the tablespace or retry with smaller value for MINEXTENTS, NEXT,or PCTINCREASE.

01700-01799: SQL Parsing

This section lists messages generated when SQL statements are parsed by the Oracle Server. Most, but not all, messages in this section indicate incorrect SQL syntax. For more information about SQL syntax, refer to *Oracle7 Server SQL Reference*.

ORA-01700

ORA-01700   duplicate username in list

Cause: A username was listed twice in a GRANT or REVOKE statement. Usernames must be unique and may only be specified once in a GRANT or REVOKE statement.

Action: Remove the duplicate username and retry the statement.

ORA-01701

ORA-01701   **a cluster is not appropriate here**

Cause: The name of a cluster was specified in a statement in which clusters are not permitted.

Action: Enter a valid CLUSTER statement or change the cluster name to the name of a valid object for the statement, such as table, view, index, or synonym.

ORA-01702

ORA-01702   **a view is not appropriate here**

Cause: The name of a view was specified in a statement in which views are not permitted.

Action: Enter the name of a valid object for the statement, such as table, cluster, synonym, or index.

ORA-01703

ORA-01703   **missing SYNONYM keyword**

Cause: The keyword SYNONYM is required in this context.

Action: Check the statement syntax and insert the keyword SYNONYM.

ORA-01704

ORA-01704   **string literal too long**

Cause: A quoted string specified as a constant was too long.

Action: Quoted strings may not contain more than 2000 characters.

ORA-01705

ORA-01705   **an outer join cannot be specified on a correlation column**

Cause: A correlation column, that is, a column in a subquery from a table referenced in the outer query's FROM clause, was followed by an outer-join indicator (+). This is not allowed.

Action: Revise the query. Refer to the syntax for subqueries. The outer-join indicator may follow only columns in the same query block in which their table is included in the FROM clause.

ORA-01706

ORA-01706   **user function result value was too large**

Cause: The user-written SQL function has generated a result that is larger than the maximum defined in the function table.

Action: Change the maximum in the function table or correct the user-written function.

ORA-01707

ORA-01707	**missing LIST keyword**

Cause: The keyword LIST is required in this context, for example, when using the ARCHIVE LOG statement to display the set of redo log files.

Action: Check the statement syntax, insert the keyword LIST where required, and retry the statement.

ORA-01708

ORA-01708   **ACCESS or SESSION expected**

Cause: An AUDIT or NOAUDIT statement contained the keyword BY followed by something other than ACCESS or SESSION.

Action: Check the statement syntax and make sure to follow the keyword BY with ACCESS or SESSION.

ORA-01709

ORA-01709   **program does not exist**

Cause: This feature is not currently implemented.

Action: No user action is required.

ORA-01710

ORA-01710	**missing OF keyword**

Cause: The keyword OF was not specified in a SELECT FOR UPDATE statement.

Action: Check and correct the statement syntax. To specify SELECT FOR UPDATE, add the following clause to the end of the SELECT statement:

FOR UPDATE OF *column-list*

ORA-01711

ORA-01711 **duplicate privilege listed**

Cause: A privilege was listed twice in a GRANT or REVOKE statement. A privilege may be specified only once in these statements.

Action: Remove the duplicate privilege from the list and retry the statement.

ORA-01712

ORA-01712 **cannot grant a privilege grantor does not have**

Cause: The grantor could not grant a privilege to another user because the grantor does not have the privilege. A grantor can grant only privileges that the grantor has or is authorized to grant.

Action: Either

- Specify only authorized privileges in the GRANT statement.

- Ask the database administrator to grant the user the required privileges.

- Ask the database administrator for the required privileges and then grant the other user the privileges.

ORA-01713

ORA-01713 **GRANT OPTION does not exist for that privilege**

Cause: A GRANT statement specified a privilege that the granting user has but is not authorized to grant to other users. When the user was granted the privilege, the user was not given the GRANT OPTION needed to grant those privileges to others.

Action: Change the GRANT statement to specify only authorized privileges or ask the user who granted the privileges to grant the GRANT OPTION also.

ORA-01714

ORA-01714   error in execution of user function

Cause: The user-written SQL function indicated that an error occurred during its execution. The meaning of the message is function-dependent.

Action: Ensure proper usage of the function or correct the function.

ORA-01715

ORA-01715   **UNIQUE may not be used with a cluster index**

Cause: An attempt was made to create a cluster index with the UNIQUE attribute. This is not permitted.

Action: Remove the keyword UNIQUE from the CREATE INDEX statement and retry the statement.

ORA-01716

ORA-01716   **NOSORT may not be used with a cluster index**

Cause: An attempt was made to create a cluster index using the NOSORT option. This is not permitted.

Action: Check the statement syntax, remove the keyword NOSORT from the CREATE INDEX statement, and retry the statement.

ORA-01717

ORA-01717   seccta: invalid access mode token passed

Cause: This is an internal error message not normally issued.

Action: Contact customer support.

ORA-01718

ORA-01718 BY ACCESS | SESSION clause not allowed for NOAUDIT

Cause: A BY ACCESS or BY SESSION clause was used in a NOAUDIT statement. BY ACCESS and BY SESSION are valid clauses in AUDIT statements, but are not permitted in NOAUDIT statements.

Action: Remove the BY ACCESS or BY SESSION clause from the NOAUDIT statement.

ORA-01719

ORA-01719	outer join operator (+) not allowed in operand of OR or IN

Cause: An outer join appears in an OR clause.

Action: If A and B are predicates, to get the effect of (A(+) OR B), try using the following:

(SELECT WHERE (A+ AND NOT B)) UNION ALL (SELECT WHERE (B));

ORA-01720

ORA-01720   **grant option does not exist for** *str*

Cause: A grant was being performed on a view, and the grant option was not present for an underlying object.

Action: Obtain the grant option on all underlying objects of the view.

ORA-01721

ORA-01721 **USERENV (COMMITSCN) invoked more than once in a transaction**

Cause: The USERENV function can be used only once in a transaction.

Action: Rewrite the transaction to use USERENV (COMMITSCN) only once.

ORA-01722

ORA-01722 **invalid number**

Cause: The attempted conversion of a character string to a number failed because the character string was not a valid numeric literal. Only numeric fields or character fields containing numeric data may be used in arithmetic functions or expressions. Only numeric fields may be added to or subtracted from dates.

Action: Check the character strings in the function or expression. Check that they contain only numbers, a sign, a decimal point, and the character "E" or "e" and retry the operation.

ORA-01723

ORA-01723 **zero-length columns are not allowed**

Cause: During CREATE TABLE, a zero-length column was specified, for example, CHAR(0).

Action: Correct the column declaration so that the length is at least 1 and try the operation again.

ORA-01724

ORA-01724   **floating point precision is out of range 1 to 126**

Cause: Floating point precision is too small or large.

Action: Correct and retry.

ORA-01725

ORA-01725 USERENV (COMMITSCN) not allowed here

Cause: The function USERENV (COMMITSCN) is allowed only as a top-level expression in the VALUES clause of an INSERT statement and on the right side of an assignment in an UPDATE statement.

Action: Correct the use of this function.

ORA-01726

ORA-01726   a table is not appropriate here

Cause: A table name was used in a statement in which tables are not permitted.

Action: Enter a valid table statement or use the current statement on the appropriate object.

ORA-01727

ORA-01727   **numeric precision specifier is out of range 1 to 38**

Cause: The precision specified for a number column in a CREATE/ALTER TABLE or CREATE CLUSTER statement must be a digit between 1 and 38. If no precision is specified, a default precision of 22 digits is used.

Action: Specify numeric precision between 1 and 38 and retry the statement.

ORA-01728

ORA-01728 **numeric scale specifier is out of range -84 to 127**

Cause: The scale specified for a numeric field in a CREATE/ALTER TABLE or CREATE CLUSTER statement is not in the valid range. It should be between -84 and 127.

Action: Specify a numeric scale between -84 and 127. If a number is not specified, the default scale of the column is 0 decimal places.

ORA-01729

ORA-01729   **database link name expected**

Cause: A database link name does not follow the at-sign (@) in a reference to a table in a remote database.

Action: Correct the reference and retry the operation. The correct syntax for denoting a table in a remote database follows:

username.table_name@ database_name

Spaces before and after the at-sign (@) are optional.

ORA-01730

ORA-01730   **invalid number of column names specified**

Cause: The number of column names specified in a CREATE VIEW statement did not correspond to the number of columns listed in the SELECT clause. If column names are specified in a CREATE VIEW statement, exactly one name must be specified for each column or expression in the SELECT clause.

Action: Specify one view column name for each column in the SELECT clause.

ORA-01731

ORA-01731   **circular view definition encountered**

Cause: Through a series of CREATE and DROP VIEW statements, a view was defined that refers to itself. For example, VIEW1 might include a column from VIEW2 that is defined as a column in VIEW1. View definitions may be seen by querying the data dictionary.

Action: Check the view definitions, remove any circular references, and retry the statements.

ORA-01732

ORA-01732   **data manipulation operation not legal on this view**

Cause: An attempt was made to use an UPDATE, INSERT, or DELETE statement on a view that contains expressions or functions or was derived from more than one table. If a join operation was used to create the view or the view contains virtual columns derived from functions or expressions, then the view may only be queried.

Action: UPDATE, INSERT, or DELETE rows in the base tables instead and restrict the operations on the view to queries.

ORA-01733

ORA-01733   virtual column not allowed here

Cause: An attempt was made to use an INSERT, UPDATE, or DELETE statement on an expression in a view.

Action: INSERT, UPDATE, or DELETE data in the base tables, instead of the view.

ORA-01735

ORA-01735   **invalid ALTER TABLE option**

Cause: An invalid option was specified in an ALTER TABLE statement.

Action: Check the statement syntax, specify a valid option, and retry the statement.

ORA-01736

ORA-01736   [NOT] SUCCESSFUL expected

Cause: An AUDIT or NOAUDIT statement contained WHENEVER followed by something other than SUCCESSFUL or NOT SUCCESSFUL.

Action: Correct the WHENEVER clause and retry the statement.

ORA-01737

ORA-01737   valid modes: [ROW] SHARE, [[SHARE] ROW] EXCLUSIVE, SHARE UPDATE

Cause: The lock mode entered was not recognized.

Action: Enter one of the following: SHARE, ROW SHARE, EXCLUSIVE, ROW EXCLUSIVE, SHARE ROW EXCLUSIVE, or SHARE UPDATE.

ORA-01738

ORA-01738   **missing IN keyword**

Cause: In a LOCK TABLE statement, the keyword IN was missing.

Action: Place the keyword IN and lock mode directly after the table name in the LOCK TABLE statement and retry the statement.

ORA-01739

ORA-01739   **missing MODE keyword**

Cause: In a LOCK TABLE statement, the keyword MODE was missing. The keyword MODE must directly follow the specified lock mode.

Action: Check the statement syntax, insert the keyword MODE where required, and retry the statement.

ORA-01740

ORA-01740   **missing double quote in identifier**

Cause: An initial double quote (") was found without a closing quote. If an identifier contains a blank or special characters other than $, #, or _, it must be enclosed in double quotes.

Action: Add a closing double quote (") to the end of the identifier.

ORA-01741

ORA-01741   illegal zero-length identifier

Cause: An attempt was made to use two double quotes ("") as an identifier. An identifier must be at least one character long.

Action: Insert at least one character between the double quotes in the identifier. If a blank identifier is required, specify a blank space between the doublequotes (" ").

ORA-01742

ORA-01742   comment not terminated properly

Cause: The indicated comment or hint beginning with a /* token is not terminated with a closing */ token.

Action: Check that the comment or hint is terminated properly.

ORA-01743

ORA-01743   **internal inconsistency; illegal user function index**

Cause: This is not currently implemented.

Action: No user action is required.

ORA-01744

ORA-01744   inappropriate INTO

Cause: The INTO clause may not be used in a subquery.

Action: Check the syntax, place the INTO clause in the top-level query, and retry the statement.

ORA-01745

ORA-01745   **invalid host/bind variable name**

Cause: A colon in a bind variable or INTO specification was followed by an inappropriate name, perhaps a reserved word.

Action: Change the variable name and retry the operation.

ORA-01746

ORA-01746 **indicator variable not permitted here**

Cause: An indicator variable is not permitted in this context.

Action: Remove the indicator variable and retry the operation.

ORA-01747

ORA-01747   **invalid user.table.column, table.column, or columns specification**

Cause: A column name was specified improperly in the current SQL statement.

Action: Check the statement's syntax, especially references to column names, and retry the statement.

ORA-01748

ORA-01748   only simple column names allowed here

Cause: This SQL statement does not allow a qualified column name, such as *username.table.column* or *table.column*.

Action: Remove the qualifications from the column and retry the operation.

ORA-01749

ORA-01749   may not GRANT/REVOKE privileges to/from self

Cause: Grantor is not allowed to grant or revoke object or system privileges to self.

Action: Issue the GRANT or REVOKE of system privileges from another database administrator account.

ORA-01750

ORA-01750   **UPDATE/REFERENCES may only be revoked from the whole table, notby column**

Cause: Although it is possible to GRANT update privileges on a column-by-column basis, it is only possible to REVOKE them for an entire table.

Action: Do not identify specific columns. To revoke update privileges for certain columns, use REVOKE for the entire table and GRANT the user privileges for specific columns.

ORA-01753

ORA-01753   **column definition incompatible with clustered column definition**

Cause: When adding a table to a cluster, the definition of the column in the table was inconsistent with the definition of the column in the cluster.

Action: The table cannot be added to the cluster until all cluster column definitions are consistent.

ORA-01754

ORA-01754   **a table may contain only one column of type LONG**

Cause: Only one column per table may be defined with datatype LONG.

Action: Remove the LONG datatype from all but one column and retry the operation.

ORA-01756

ORA-01756   **quoted string not properly terminated**

Cause: A quoted string must be terminated with a single quote mark (').

Action: Insert the closing quote and retry the statement.

ORA-01758

ORA-01758 **table must be empty to add mandatory NOT NULL column**

Cause: It is not possible to define a new column as NOT NULL if rows already exist in the table being modified.

Action: Retry the statement without the NOT NULL specification.

ORA-01759

ORA-01759   **user function is incorrectly defined**

Cause: A user function has been improperly defined.

Action: Contact customer support.

ORA-01760

ORA-01760   **illegal argument for function**

Cause: The argument or arguments specified for the function are not valid in this context.

Action: Check the definition of the function and correct the arguments.

ORA-01761

ORA-01761 **DML operation does not map to a unique table in the join**

Cause: The primary table is the base table against which the update, insert or delete operation is ultimately performed. If the operation is a delete, either there is no primary table in the join or there is more than one primary table. If the operation is an update or an insert, the specified columns map to more than one base table.

Action: Modify the join specification so that the situations described above do not occur.

ORA-01762

ORA-01762   vopdrv: view query block not in FROM

Cause: This is an internal error message not normally issued.

Action: Contact customer support.

ORA-01763

ORA-01763 **update or delete involves outer joined table**

Cause: If the operation is a delete, the table being deleted from is outer-joined to some other table. If the operation is an update, either the table being updated is outer-joined to some other table, or a table that is reachable from the primary table is being outer-joined to a table not reachable from the primary table.

Action: Modify the join specification so that the situations described above do not occur.

ORA-01764

ORA-01764   **new update value of join is not guaranteed to be unique**

Cause: A row of a join query table is being updated using a row of a table that is not guaranteed to have only one value for the row being updated.

Action: Modify the join specification, so that the situation described above does not occur.

ORA-01765

ORA-01765   **specifying table's owner name is not allowed**

Cause: A table or view name was specified with a period (.) or identified with an invalid prefix.

Action: Check that the object is identified correctly and, if necessary, remove the period from the name.

ORA-01766

ORA-01766 **dictionary table not allowed in this context**

Cause: The name of a data dictionary object was used outside the FROM clause of a SELECT statement. The only operation allowed on a data dictionary object is to SELECT from it.

Action: Restrict operations on data dictionary objects to queries.

ORA-01767

ORA-01767   UPDATE ... SET expression must be a subquery

Cause: An attempt was made to use a single expression without a subquery to update a list of columns.

Action: Check the syntax for using subqueries to update data and add the necessary subquery.

ORA-01768

ORA-01768 number string too long

Cause: A number entered exceeds the limit of 255 characters, including digits, sign, decimal point, and exponent.

Action: Shorten the number string, perhaps by expressing it in scientific notation, for example, 1.85E9 instead of 1,850,000,000.

ORA-01769

ORA-01769   **duplicate CLUSTER option specifications**

Cause: During a CREATE of a clustered table, an attempt was made to specify more than one CLUSTER option.

Action: Remove the extra CLUSTER option and retry the statement.

ORA-01770

ORA-01770 CLUSTER option not allowed in CREATE CLUSTER command

Cause: An attempt was made to use the CLUSTER option within a CREATECLUSTER statement.

Action: Remove the CLUSTER clause and try again.

ORA-01771

ORA-01771   illegal option for a clustered table

Cause: CLUSTER option conflicts with MAXTRANS option.

Action: Choose one option and retry the statement.

ORA-01773

ORA-01773   **may not specify column datatypes in this CREATE TABLE**

Cause: Column datatypes were specified in a SELECT statement used to create and load a table directly.

Action: Remove column datatypes. The datatypes of the SELECT list expressions are automatically used as the column datatypes.

ORA-01775

ORA-01775 **looping chain of synonyms**

Cause: Through a series of CREATE synonym statements, a synonym was defined that referred to itself. For example, the following definitions are circular:

CREATE SYNONYM s1 for s2
CREATE SYNONYM s2 for s3
CREATE SYNONYM s3 for s1

Action: Change one synonym definition so that it applies to a base table or view and retry the operation.

ORA-01777

ORA-01777   **WITH GRANT OPTION not allowed in this system**

Cause: This version of Oracle does not support the WITH GRANT OPTION.

Action: Remove the WITH GRANT option.

ORA-01778

ORA-01778   **maximum subquery nesting level exceeded**

Cause: An attempt was made to use more than 255 levels of nested subqueries.

Action: Merge some of the subqueries into their containing queries, until there are less than 255 levels of nesting.

ORA-01780

ORA-01780   **string literal required**

Cause: In a COMMENT statement, the comment following the keyword IS must be a quoted string literal.

Action: Check that the comment is a string literal.

ORA-01781

ORA-01781   UNRECOVERABLE cannot be specified without AS SELECT

Cause: UNRECOVERABLE was specified in a CREATE TABLE statement without also specifying a populating subquery with an AS clause.

Action: Do not specify UNRECOVERABLE.

ORA-01782

ORA-01782   **UNRECOVERABLE cannot be specified for a cluster or clustered table**

Cause: A CREATE CLUSTER or clustered CREATE TABLE statementspecified UNRECOVERABLE.

Action: Do not specify UNRECOVERABLE.

ORA-01783

ORA-01783   only one RECOVERABLE or UNRECOVERABLE clause may be specified

Cause: RECOVERABLE was specified more than once, UNRECOVERABLE was specified more than once, or both RECOVERABLE and UNRECOVERABLE were specified in a CREATE TABLE or CREATE INDEX statement.

Action: Remove all but one of the RECOVERABLE or UNRECOVERABLE clauses and reissue the statement.

ORA-01784

ORA-01784   **RECOVERABLE cannot be specified with database media recovery disabled**

Cause: A CREATE TABLE or CREATE INDEX statement specified RECOVERABLE when the database was running in NOARCHIVELOG mode. Since logs are not being archived, they will be overwritten and the object being created cannot be recovered from a backup taken before the object was created.

Action: Do not specify RECOVERABLE, or restart the database with media recovery enabled.

ORA-01785

ORA-01785   **ORDER BY item must be the number of a SELECT-list expression**

Cause: Either an ORDER BY item for a set expression is not a column number, or the ORDER BY item is not the number of a SELECT list column.

Action: The ORDER BY item must be a number between 1 and the number of columns in the SELECT list. Check the column number and retry the statement.

ORA-01786

ORA-01786   **FOR UPDATE of this query expression is not allowed**

Cause: An attempt was made to use a FOR UPDATE clause on the result of a set expression involving GROUP BY, DISTINCT, UNION, INTERSECT, or MINUS.

Action: Check the syntax, remove the FOR UPDATE clause, and retry the statement.

ORA-01787

ORA-01787   only one clause allowed per query block

Cause: An attempt was made to use more than one WHERE, GROUP BY, CONNECT BY, or HAVING clause in the query.

Action: Combine the duplicate clauses into a single clause and retry the statement.

ORA-01788

ORA-01788   **CONNECT BY clause required in this query block**

Cause: A START WITH clause was specified without a CONNECT BY clause.

Action: Check the syntax, specify a CONNECT BY clause, and retry the statement.

ORA-01789

ORA-01789   **query block has incorrect number of result columns**

Cause: All of the queries participating in a set expression do not contain the same number of SELECT list columns.

Action: Check that all the queries in the set expression have the same number of SELECT list columns.

ORA-01790

ORA-01790   **expression must have same datatype as corresponding expression**

Cause: A SELECT list item corresponds to a SELECT list item with a different datatype in another query of the same set expression.

Action: Check that all corresponding SELECT list items have the same datatypes. Use the TO_NUMBER, TO_CHAR, and TO_DATE functions to do explicit data conversions.

ORA-01791

ORA-01791   **not a SELECTed expression**

Cause: There is an incorrect ORDER BY item. The query is a SELECT DISTINCT query with an ORDER BY clause. In this context, all ORDER BY items must be constants, SELECT list expressions, or expressions whose operands are constants or SELECT list expressions.

Action: Remove the inappropriate ORDER BY item from the SELECT list and retry the statement.

ORA-01792

ORA-01792   maximum number of columns in a table or view is 254

Cause: While creating a table or view, more than 254 columns were specified.

Action: Remove some columns from the table or view definition. If all the information is required, split the columns into two tables or views and join the columns when querying.

ORA-01793

ORA-01793   **maximum number of index columns is 16**

Cause: While creating an index, more than 16 columns were specified.

Action: Remove some columns from the index definition. If all the columns indexed are required, split them into two indexes.

ORA-01794

ORA-01794   **maximum number of cluster columns is 16**

Cause: More than 16 columns were specified in the cluster key.

Action: Remove some of the columns from the cluster key definition. Consider concatenating multiple character strings into a single column.

ORA-01795

ORA-01795   **maximum number of expressions in a list is 254**

Cause: More than 254 columns or expressions were specified in a list.

Action: Remove some of the expressions from the list.

ORA-01796

ORA-01796   **this operator cannot be used with lists**

Cause: A relational operator is not allowed in this context. When comparing lists, use only an operator that tests for equivalence, such as =, !=, or IN.

Action: Rephrase the query so that it only compares lists for equivalence. For example, the following clause is invalid:

WHERE (A,B) > ((C,D), (E,F))

It may be necessary to compare individual columns separately. For example, to see if A and B are respectively greater than C and D, use WHERE A>B AND C>D instead of WHERE (A,B)>(C,D).

ORA-01797

ORA-01797 **this operator must be followed by ANY or ALL**

Cause: The keyword ANY or ALL was missing.

Action: Check the statement syntax and add ANY or ALL where appropriate.

ORA-01798

ORA-01798   missing EXCEPTION keyword

Cause: The EXCEPTION keyword is missing from the ENABLE clause.

Action: Correct the syntax.

ORA-01799

ORA-01799   **a column may not be outer-joined to a subquery**

Cause: An expression is not permitted to be joined to a subquery.

Action: Either remove the join or convert the subquery to a view.

01800-01899: The Date Function

This section lists messages generated when the Oracle date function is accessed.

ORA-01800

ORA-01800   **a literal in the date format was too large to process**

Cause: A literal specified in a date format was too large. If a literal is entered as a date, it must be less than 220 characters long.

Action: Use only literals less than 220 characters long in the date format.

ORA-01801

ORA-01801   **date format is too long for internal buffer**

Cause: The date format string was too long to process. This should occur only if several long literals are specified as part of a date.

Action: Remove long literals from the date format string.

ORA-01802

ORA-01802 **Julian date is out of range**

Cause: An invalid Julian date was entered. Valid Julian dates run from 1 to 3,442,447.

Action: Enter a valid Julian date between 1 and 3,442,447.

ORA-01803

ORA-01803   **failure in getting date/time**

Cause: This is an internal error message that is not likely to occur. It is usually caused by a failure in the system clock.

Action: Investigate the system clock and the process that requested the date or time. If necessary, contact customer support.

ORA-01810

ORA-01810   **format code appears twice**

Cause: A format code was listed twice in a date specification. Each format code may be specified only once in the function
               TO_DATE.

Action: Remove the duplicate format code from the date specification, then retry the operation.

ORA-01811

ORA-01811   **Julian date precludes use of day of year**

Cause: Both a Julian date and a day of the year were specified in the TO_DATE function. If a Julian date is specified, the day of the year (DDD) may not be specified, as it is contained in the Julian date.

Action: Remove the day of the year or the Julian date from the specification, then retry the operation.

ORA-01812

ORA-01812   **year may only be specified once**

Cause: More than one year format code was listed in a date specification. Only one of the following year format codes may be specified in a date: YYYY, YYY, YY, Y.

Action: Remove all but one year format from the date specification.

ORA-01813

ORA-01813   hour may only be specified once

Cause: More than one hour format code was listed in a date specification. Only one of the following hour format codes may be specified in a date: HH, HH12, HH24.

Action: Remove all but one hour format from the date specification.

ORA-01814

ORA-01814   **AM/PM conflicts with use of A.M./P.M.**

Cause: Both types of meridian indicators, AM and PM, were listed in a date specification. If one of the meridian indicator format masks, such as AM or A.M., is specified in the date, the other may not be specified.

Action: Use one meridian indicator format, with or without periods, consistently.

ORA-01815

ORA-01815   **BC/AD conflicts with use of B.C./A.D.**

Cause: Both types of BC/AD indicators were listed in a date specification. If one of the BC/AD indicator format masks, such as BC or B.C., is specified in the date, the other may not be specified.

Action: Use one BC/AD indicator format, with or without periods, consistently.

ORA-01816

ORA-01816 **month may only be specified once**

Cause: More than one month format code was listed in a date specification. Only one of the following month format codes may be specified in a date: MM,MON, MONTH.

Action: Remove all but one month format from the date specification.

ORA-01817

ORA-01817   **day of week may only be specified once**

Cause: More than one day-of-the-week format code was listed in a date specification. Only one of the following day-of-the-week format codes may be specified in a date: D, DY, DAY.

Action: Remove all but one day-of-the-week format from the date specification.

ORA-01818

ORA-01818 **HH24 precludes use of meridian indicator**

Cause: A date specification contained both a 24-hour time format code and a meridian indicator code. If hours are specified using the 24-hour time format (HH24), a meridian indicator, AM or PM, may not be specified.

Action: Remove the meridian indicator format code, AM/PM or A.M./P.M., or the 24-hour time format code, HH24, from the date specification.

ORA-01819

ORA-01819 **signed year precludes use of BC/AD**

Cause: A date specification contained both a signed year and a B.C./A.D. indicator. If the year is specified with a sign, such as SYYYY, then B.C. or A.D. is implicit in the date and must not be entered.

Action: Remove the B.C./A.D. indicator from the date specification.

ORA-01820

ORA-01820 **format code cannot appear in date input format**

Cause: A date specification contained an invalid format code. Only the following may be specified when entering a date: year, month, day, hours, minutes, seconds, Julian day, A.M./P.M. and B.C./A.D.

Action: Remove the invalid format code from the date specification.

ORA-01821

ORA-01821   **date format not recognized**

Cause: A date specification contained an invalid format code.

Action: Check that only valid date format codes are specified.

ORA-01830

ORA-01830   **date format picture ends before converting entire input string**

Cause: A valid date format picture included extra data. The first part of the format picture was converted into a valid date, but the remaining data was not required.

Action: Check the specifications for date format pictures and correct the statement.

ORA-01831

ORA-01831   **year conflicts with Julian date**

Cause: The wrong year was specified with a Julian day. If a year is specified with a Julian date, it must be the year in which the Julian date occurs.

Action: Remove the year value from the date specification or enter the correct year for the Julian date.

ORA-01832

ORA-01832   **day of year conflicts with Julian date**

Cause: A Julian date was specified with the day of the year but the day did not correspond to the Julian date. If the day of the year is specified with a Julian date, it must be the same day as the Julian date.

Action: Remove the day of the year value from the date specification or enter the correct day for the Julian date.

ORA-01833

ORA-01833   **month conflicts with Julian date**

Cause: The wrong month was specified with a Julian date. If a month is specified with a Julian date, it must be the month in which the Julian date occurs.

Action: Remove the month value from the date specification or enter the correct month for the Julian date.

ORA-01834

ORA-01834 **day of month conflicts with Julian date**

Cause: A Julian date was specified with the day of the month, but the month day did not correspond to the Julian date. If the day of the month is specified with a Julian date, it must be the same day of the month as the Julian date.

Action: Remove the day of the month value from the date specification or enter the correct day of the month for the Julian date.

ORA-01835

ORA-01835   **day of week conflicts with Julian date**

Cause: A Julian date was specified with the day of the week, but the weekday did not correspond to the Julian date. If the day of the week is specified with a Julian date, it must be the same day of the week as the Julian date.

Action: Remove the day of the week value from the date specification or enter the correct day of the week for the Julian date.

ORA-01836

ORA-01836 **hour conflicts with seconds in day**

Cause: The wrong hour was specified with seconds in the day. If an hour is specified with seconds past midnight (SSSSS), it must be the hour in which the seconds value falls.

Action: Remove the hour value from the date specification or specify the correct hour for the seconds past midnight.

ORA-01837

ORA-01837   **minutes of hour conflicts with seconds in day**

Cause: A date specification contained both minutes of the hour and seconds in the day but the values did not correspond. If both minutes in the hour (MI) and seconds past midnight (SSSSS) are specified, the minutes value must be the minute in which the seconds value will occur.

Action: Remove the minutes value from the date specification or enter the correct minute value for the specified seconds value.

ORA-01838

ORA-01838   **seconds of minute conflicts with seconds in day**

Cause: A date specification contained both seconds of the minute and seconds in the day but the values did not correspond. If both types of seconds are specified, the seconds of the minute value (SS) must be the second in which the seconds past midnight value (SSSSS) will fall.

Action: Remove the seconds of the minute value from the date specification or enter a value that corresponds to the given seconds in the day.

ORA-01839

ORA-01839   **date not valid for month specified**

Cause: The day of the month specified in the date is invalid for the given month. The day of the month (DD) must be between 1 and the number of days inthe month.

Action: Enter a valid day of the month for the specified month.

ORA-01840

ORA-01840   **input value not long enough for date format**

Cause: The data to be converted to date format was incomplete; the date format picture was longer than the input data.

Action: Either add more input or shorten the date picture format, then retry the operation.

ORA-01841

ORA-01841   **(full) year must be between -4713 and +9999**

Cause: A date specified a year that is not in the valid date range. A valid date is any date between January 1, 4712 B.C. and December 31, 4712 A.D.

Action: Enter a valid date value between 4712 B.C. and 4712 A.D.

ORA-01842

ORA-01842   **quarter must be between 1 and 4**

Cause: An invalid value was specified for the quarter of the year in a date. The quarter (Q) must be between 1 and 4.

Action: Enter a value for quarter between 1 and 4.

ORA-01843

ORA-01843   not a valid month

Cause: A date specified an invalid month. Valid months are: January-December, for format code MONTH, and Jan-Dec, for format code MON.

Action: Enter a valid month value in the correct format.

ORA-01844

ORA-01844   week of year must be between 1 and 52

Cause: An invalid value was specified for the week of the year in a date. Week of the year (WW) must be between 1 and 52.

Action: Enter a week of the year value between 1 and 52.

ORA-01845

ORA-01845   week of month must be between 1 and 5

Cause: An invalid value was specified for the week of the month in a date. Week of the month (W) must be between 1 and 5.

Action: Enter a week of the month value between 1 and 5.

ORA-01846

ORA-01846   **not a valid day of the week**

Cause: A date specified an invalid day of the week. Valid days are

- _Monday-Sunday, for format code DAY
- _Mon-Sun, for format code DY
- _1-7, for format code D

Action: Enter a valid day of the week value in the correct format.

ORA-01847

ORA-01847 **day of month must be between 1 and last day of month**

Cause: The day of the month listed in a date is invalid for the specified month. The day of the month (DD) must be between 1 and the number of days in that month.

Action: Enter a valid day value for the specified month.

ORA-01848

ORA-01848 **day of year must be between 1 and 365 (366 for leap year)**

Cause: An invalid day of the year was specified in a date. Day of the year (DDD) must be between 1 and 365 for a non-leap year or 1 and 366 for a leap year.

Action: Enter a day of the year value between 1 and 365 (or 366).

ORA-01849

ORA-01849   **hour must be between 1 and 12**

Cause: An invalid hour was specified for a date using the 12-hour time format. If a 12-hour format code (HH or HH12) is used, the specified hour must be between 1 and 12.

Action: Enter an hour value between 1 and 12.

ORA-01850

ORA-01850   **hour must be between 0 and 23**

Cause: An invalid hour was specified for a date using the 24-hour time format. If the 24-hour format code (HH24) is listed, the specified hour must be between 0 and 23.

Action: Enter an hour value between 0 and 23.

ORA-01851

ORA-01851   **minutes must be between 0 and 59**

Cause: An invalid minute value was specified in a date. Minutes must be between 0 and 59.

Action: Enter a minute value between 0 and 59.

ORA-01852

ORA-01852   **seconds must be between 0 and 59**

Cause: An invalid seconds value was specified in a date. Seconds must be between 0 and 59 if the seconds format code (SS) is used.

Action: Enter a seconds value between 0 and 59.

ORA-01853

ORA-01853   **seconds in day must be between 0 and 86399**

Cause: An invalid value for seconds in the day was specified in a date. Seconds must be between 0 and 86399 if the seconds past midnight format code (SSSSS) is used.

Action: Specify a seconds value between 0 and 86399.

ORA-01854

ORA-01854   **Julian date must be between 1 and 5373484**

Cause: An invalid Julian date was entered.

Action: Correct and retry.

ORA-01855

ORA-01855   **AM/A.M. or PM/P.M. required**

Cause: A date specification contained a meridian indicator format code but a valid meridian indicator was not specified. If a meridian indicator code, A.M./P.M. or AM/PM, is included in a date format, the date must include a valid meridian indicator.

Action: Specify the meridian indicator in the correct format, A.M./AM or P.M./PM.

ORA-01856

ORA-01856   **BC/B.C. or AD/A.D. required**

Cause: A date specification contained a BC/AD format code but a valid BC/AD indicator was not specified. If one of the BC/AD format codes, BC/AD or B.C./A.D., is specified in a date format, the date must include BC/B.C.or AD/A.D.

Action: Specify the BC/AD indicator in the date using the correct format.

ORA-01857

ORA-01857   **not a valid time zone**

Cause: An incorrect time zone code was specified for the NEW_TIME function.

Action: Correct the invalid time zone code, then retry the operation.

ORA-01858

ORA-01858   **a non-numeric character found where a digit was expected**

Cause: The input data to be converted using a date format model was incorrect. The format model expected a number but found a non-numeric character.

Action: Check the input data and the date format model to make sure the elements match in number and type, then retry the operation.

ORA-01859

ORA-01859   **a non-alphabetic character was found where a letter was expected**

Cause: The input data to be converted using a date format model was incorrect. The format model expected a letter but found a non-alphabetic character.

Action: Check the input data and the date format model to make sure the elements match in number and type, then retry the operation.

ORA-01860

ORA-01860   week of year must be between 1 and 53

Cause: A week number outside the range 1 to 53 was specified.

Action: Use only a value within the valid range for the week number.

ORA-01861

ORA-01861   **literal does not match format string**

Cause: Literals in the input string must be the same length as the literals in the format string, except for the leading white space characters. If the "FX" modifier is specified, the literal must match exactly, including the leading white space.

Action: Correct the format string to match the string literal.

ORA-01862

ORA-01862 **wrong number of digits for this format item**

Cause: The number of digits in this format item is not equal to the number required for the format. This error usually occurs when the "FX" modifier (format exact) has been specified.

Action: Resubmit the command using the required amount of numbers for the format mask.

ORA-01898

ORA-01898   too many precision specifiers

Cause: While trying to truncate or round dates, extra data was found in the date format picture.

Action: Check the syntax of the date format picture and retry.

ORA-01899

ORA-01899 **bad precision specifier**

Cause: An invalid precision code was specified in a date.

Action: Enter a valid precision code in the date format. Valid precision types are: century, year, month, day, Julian date, hours, minutes, and seconds.

01900-02039: SQL Parsing

This section lists some of the messages generated when SQL statements are parsed by the Oracle Server.

ORA-01900

ORA-01900 **LOGFILE keyword expected**

Cause: The keyword LOGFILE is required in this context.

Action: Check the syntax, insert the keyword LOGFILE where required, then re-execute the statement.

ORA-01901

ORA-01901   ROLLBACK keyword expected

Cause: The keyword ROLLBACK is required in this context.

Action: Check the syntax, insert the keyword ROLLBACK where required, and re-execute the statement.

ORA-01902

ORA-01902   **SEGMENT keyword expected**

Cause: The keyword SEGMENT is required in this context.

Action: Check the syntax, insert the keyword SEGMENT where required, and re-execute the statement.

ORA-01903

ORA-01903 **EVENTS keyword expected**

Cause: The keyword EVENTS is required in this context.

Action: Check the syntax, insert the keyword EVENTS where required, and re-execute the statement.

ORA-01904

ORA-01904   DATAFILE keyword expected

Cause: The keyword DATAFILE is required in this context.

Action: Check the syntax, insert the keyword DATAFILE where required, and re-execute the statement.

ORA-01905

ORA-01905 **STORAGE keyword expected**

Cause: The keyword STORAGE is required in this context.

Action: Check the syntax, insert the keyword STORAGE where required, and re-execute the statement.

ORA-01906

ORA-01906 BACKUP keyword expected

Cause: The keyword BACKUP is required in this context.

Action: Check the syntax, insert the keyword BACKUP where required, and re-execute the statement.

ORA-01907

ORA-01907   **TABLESPACE keyword expected**

Cause: The keyword TABLESPACE is required in this context.

Action: Check the syntax, insert the keyword TABLESPACE where required, and re-execute the statement.

ORA-01908

ORA-01908 **EXISTS keyword expected**

Cause: The keyword EXISTS is required in this context.

Action: Check the syntax, insert the keyword EXISTS where required, and re-execute the statement.

ORA-01909

ORA-01909 **REUSE keyword expected**

Cause: The keyword REUSE is required in this context.

Action: Check the syntax, insert the keyword REUSE where required, and re-execute the statement.

ORA-01910

ORA-01910   TABLES keyword expected

Cause: The keyword TABLES is required in this context.

Action: Check the syntax, insert the keyword TABLES where required, and re-execute the statement.

ORA-01911

ORA-01911 **CONTENTS keyword expected**

Cause: The keyword CONTENTS is required in this context.

Action: Check the syntax, insert the keyword CONTENTS where required, and re-execute the statement.

ORA-01912

ORA-01912   **ROW keyword expected**

Cause: The keyword ROW is required in this context.

Action: Check the syntax, insert the keyword ROW where required, and re-execute the statement.

ORA-01913

ORA-01913   **EXCLUSIVE keyword expected**

Cause: The keyword EXCLUSIVE is required in this context.

Action: Check the syntax, insert the keyword EXCLUSIVE where required, and re-execute the statement.

ORA-01914

ORA-01914	invalid auditing option for sequence numbers

Cause: AUDIT or NOAUDIT on a sequence number specifies an auditing option that is legal for tables but is not legal for sequence numbers.

Action: The following options may not be used for sequence numbers and should be removed from the statement: COMMENT, DELETE, INDEX, INSERT, LOCK, RENAME, UPDATE, REFERENCES, and EXECUTE.

ORA-01915

ORA-01915 **invalid auditing option for views**

Cause: AUDIT or NOAUDIT on a view specifies an auditing option that is legal for tables but is not legal for views.

Action: The following options may not be used for views and should be removed from the statement: ALTER, INDEX, REFERENCES, and EXECUTE.

ORA-01916

ORA-01916   **ONLINE keyword or OFFLINE keyword expected**

Cause: The keyword ONLINE or OFFLINE is required in this context.

Action: Check the syntax, insert the keyword ONLINE or OFFLINE where required, and re-execute the statement.

ORA-01917

ORA-01917 **user or role *name* does not exist**

Cause: An invalid user or role name was specified.

Action: Check that a valid user or role name is used.

ORA-01918

ORA-01918 **user *name* does not exist**

Cause: The user name specified cannot be found in the system.

Action: Check that a valid user or role name is used.

ORA-01919

ORA-01919   role *name* does not exist

Cause: An invalid role name was specified.

Action: Check that a valid role name is specified.

ORA-01920

ORA-01920   **user name *name* conflicts with another user or role name**

Cause: A user or role by that name already exists.

Action: Choose a different user name.

ORA-01921

ORA-01921   **role name *name* conflicts with another user or role name**

Cause: A user or role by that name already exists.

Action: Choose a different role name.

ORA-01922

ORA-01922 **CASCADE must be specified to drop** *name*

Cause: The user owns objects that need to be dropped along with the user.

Action: Use the CASCADE command.

ORA-01923

ORA-01923   **CASCADE aborted, objects locked by another user**

Cause: A CASCADE operation was aborted because the specified user owns objects that are locked and thus cannot be dropped.

Action: Use the lock monitor to determine who locked the objects. The objects must be unlocked before the specified user can be dropped.

ORA-01924

ORA-01924   **role *name* not granted or does not exist**

Cause: The grantor attempted to use a SET ROLE command to enable a role not granted to the grantor, or to enable a subrole, or the grantor did not have the SET ROLE system privilege.

Action: To use the SET ROLE command, check that the grantor has the SET ROLE system privilege and is granted the role to be passed on.

ORA-01925

ORA-01925   **maximum of *num* enabled roles exceeded**

Cause: The specified number of enabled roles exceeds the value specified in the initialization parameter MAX_ENABLED_ROLES.

Action: Increase MAX_ENABLED_ROLES and restart the instance.

ORA-01926

ORA-01926   **cannot grant WITH GRANT OPTION to a role**

Cause: Privileges with GRANT OPTION cannot be granted to a role.

Action: Perform the grant without the GRANT OPTION.

ORA-01927

ORA-01927   **original grantor must REVOKE privileges**

Cause: Only the original grantor of privileges can REVOKE those privileges.

Action: Ask the database administrator for the grantor of the privileges and ask that person to revoke the privileges.

ORA-01928

ORA-01928 GRANT option not granted for all privileges

Cause: The grantor does not have the GRANT OPTION for some or all of the privileges specified. These privileges are not granted.

Action: Check that grantor has the GRANT OPTION for privileges to be granted to a user or role.

ORA-01929

ORA-01929   **no privileges to GRANT**

Cause: The grantor attempted to grant privileges using the object privilege ALL, but was not granted privileges to do so via the GRANT OPTION.

Action: Check that the grantor has the necessary privileges via the GRANT OPTION and try again.

ORA-01930

ORA-01930   no privileges to REVOKE

Cause: The REVOKE ALL command was issued against a user when there were no privileges to revoke.

Action: This action is unnecessary.

ORA-01931

ORA-01931   cannot grant UNLIMITED TABLESPACE, REFERENCES, or INDEX to a role

Cause: The UNLIMITED TABLESPACE, REFERENCES, or INDEX privileges cannot be granted to a role.

Action: Grant these privileges directly to the user.

ORA-01932

ORA-01932   WITH ADMIN option not granted for role *name*

Cause: Grantor attempted an operation on a role, but does not have the necessary privileges via the ADMIN OPTION.

Action: Obtain the necessary privileges via the ADMIN OPTION and try again.

ORA-01933

ORA-01933   **cannot create a stored object using privileges from a role**

Cause: An attempt was made to create a stored object using privileges from a role. Stored objects cannot use privileges from roles.

Action: Grant the required privileges to the user directly.

ORA-01934

ORA-01934   **circular role grant detected**

Cause: A role, the primary role, cannot be granted to itself or another role that is granted via one or more roles to the primary role. This would cause a circular definition. See *Oracle7 Server SQL Reference* for more detail.

Action: Analyze the role and user structure to be used and check that no attempt is made to grant roles circularly.

ORA-01935

ORA-01935   **missing user or role name**

Cause: The entered statement required a user or role name, and it was not specified.

Action: Specify a user or role name in the statement.

ORA-01936

ORA-01936   cannot specify owner when creating users or roles

Cause: An attempt was made to specify an owner while creating a user or a role. Users and roles do not have owners.

Action: Do not specify an owner when creating a user or a role.

ORA-01937

ORA-01937   **invalid role name**

Cause: An invalid role name was specified.

Action: Choose a different role name.

ORA-01938

ORA-01938   **IDENTIFIED BY must be specified for CREATE USER**

Cause: An attempt was made to create a user without using the IDENTIFIED EXTERNALLY clause or the IDENTIFIED BY password clause.

Action: In the CREATE USER statement, either assign the user a password using the IDENTIFIED BY password clause or use the IDENTIFIED EXTERNALLY clause, which allows the operating system to perform user authentication.

ORA-01939

ORA-01939   **only the ADMIN OPTION can be specified**

Cause: An attempt was made to grant a system privilege or a role to a user using a GRANT statement with the GRANT OPTION (GRANT Form I). System privileges and roles must be granted using the WITH ADMIN OPTION (GRANT Form II).

Action: Use the WITH ADMIN OPTION, rather than the WITH GRANT OPTION, in the GRANT statement.

ORA-01940

ORA-01940   **cannot DROP a user that is currently logged in**

Cause: An attempt was made to drop a user that was currently logged in.

Action: Make sure the user is logged out, then re-execute the command.

ORA-01941

ORA-01941 **SEQUENCE keyword expected**

Cause: The keyword SEQUENCE is required in this context.

Action: Check syntax, insert keyword SEQUENCE as required, and try again.

ORA-01942

ORA-01942   IDENTIFIED BY and EXTERNALLY cannot both be specified

Cause: A user was specified with an Oracle password and identified externally.

Action: Specify only one of these options. For details and syntax, see *Oracle7 Server SQL Reference.*

ORA-01943

ORA-01943   IDENTIFIED BY already specified

Cause: The IDENTIFIED BY 'password' clause was specified more than once in a statement.

Action: Use the clause IDENTIFIED BY only once in the statement. If desired, change the user's password later with an ALTER USER command.

ORA-01944

ORA-01944   **IDENTIFIED EXTERNALLY already specified**

Cause: The keyword IDENTIFIED EXTERNALLY was specified more than once in a statement.

Action: Use the keyword IDENTIFIED EXTERNALLY only once in the statement.

ORA-01945

ORA-01945   DEFAULT ROLE[S] already specified

Cause: The keyword DEFAULT ROLE was used more than once in a CREATE USER or ALTER USER statement.

Action: Use the keyword DEFAULT ROLE only once in the statement. If desired, change the user's role later with an ALTER USER statement.

ORA-01946

ORA-01946   **DEFAULT TABLESPACE already specified**

Cause: The keyword DEFAULT TABLESPACE was specified more than once in a CREATE USER or ALTER USER statement.

Action: Use the keyword DEFAULT TABLESPACE only once in the statement. If desired, change the user's tablespace later with an ALTER USER statement.

ORA-01947

ORA-01947   **TEMPORARY TABLESPACE already specified**

Cause: The keyword TEMPORARY TABLESPACE was specified more than once in a CREATE USER or ALTER USER statement.

Action: Use the keyword TEMPORARY TABLESPACE only once in the statement. If desired, change the user's tablespace later with an ALTER USER statement.

ORA-01948

ORA-01948   **invalid DEFAULT ROLE specification**

Cause: An invalid role name was specified.

Action: Check that the role name is correct.

ORA-01949

ORA-01949	**ROLE keyword expected**

Cause: The context of the command or argument requires the use of the ROLE keyword; for example, DROP [ROLE], DEFAULT [ROLE], and so on.

Action: Check syntax and try again. For syntax rules, see *Oracle7 Server SQL Reference.*

ORA-01950

ORA-01950 **no privileges on tablespace "*name*"**

Cause: The attempt to give the user a tablespace quota failed because the user does not have the necessary system privileges.

Action: Either grant the user the system privileges needed to create objects in the specified tablespace or grant the user a specific space resource in the tablespace.

ORA-01951

ORA-01951   **role not granted for ROLE:** *name*, **user:** *name*

Cause: An attempt was made to revoke a role not granted to the user.

Action: Check that the user has the role.

ORA-01952

ORA-01952 **system privileges not granted to** *name*

Cause: An attempt was made to revoke a system privilege not granted to the user.

Action: Check that the user has the system privilege.

ORA-01953

ORA-01953   command no longer valid, see ALTER USER

Cause: The syntax for assigning quotas on tablespaces has changed. The ALTER USER command is now used to grant quotas on tablespaces.

Action: Use the ALTER USER command.

ORA-01954

ORA-01954 **DEFAULT ROLE clause not valid for CREATE USER**

Cause: An attempt was made to grant a default role to a user in a CREATE USER statement. A DEFAULT ROLE clause cannot be used in a CREATEUSER statement.

Action: Use a GRANT ROLE statement to grant an initial role to the user; then use the ALTER USER command to assign any desired default roles.

ORA-01955

ORA-01955 **user not granted DEFAULT ROLE** *name*

Cause: An attempt was made to set the default role of a user to a role the user was not granted.

Note: Subroles cannot be specified in a DEFAULT ROLE clause.

Action: Check that the user is granted the role before specifying it as the default.

ORA-01956

ORA-01956   **invalid command when OS_ROLES are being used**

Cause: An attempt to grant a role to a user failed because Oracle database roles are disabled and operating system roles are enabled. This occurred because the OS_ROLES parameter in the initialization parameter file is set to TRUE.

Action: Ask the database administrator or security administrator how to set roles via the operating system. See also your Oracle7 operating system-specific Oracle documentation.

ORA-01967

ORA-01967 **invalid option for CREATE CONTROLFILE**

Cause: An invalid CREATE CONTROLFILE option is present.

Action: Specify only valid CREATE CONTROLFILE options.

ORA-01968

ORA-01968   **only specify RESETLOGS or NORESETLOGS once**

Cause: The keyword RESETLOGS or NORESETLOGS has appeared more than once in the CREATE CONTROLFILE statement.

Action: Specify RESETLOGS or NORESETLOGS only once.

ORA-01969

ORA-01969   you must specify RESETLOGS or NORESETLOGS

Cause: The RESETLOGS or NORESETLOGS option is missing from the CREATE CONTROL FILE statement.

Action: Specify RESETLOGS or NORESETLOGS on the command line.

ORA-01970

ORA-01970   you must specify a database name for CREATE CONTROLFILE

Cause: The CREATE CONTROLFILE statement issued is missing a database name.

Action: Issue the CREATE CONTROLFILE statement again with the appropriate database name for the DATABASE keyword.

ORA-01973

ORA-01973 missing change number

Cause: The keyword CHANGE was specified on the command line, but the change number was not specified.

Action: Check the syntax of the command and retry it.

ORA-01974

ORA-01974 **illegal archive option**

Cause: An invalid option to the ALTER SYSTEM ARCHIVE command was specified.

Action: Check the syntax of the command and retry it.

ORA-01975

ORA-01975   **illegal character in change number** *num*

Cause: An invalid change number was supplied.

Action: Resubmit the command with a valid change number.

ORA-01976

ORA-01976   missing change number

Cause: A change number was required, but not supplied.

Action: Check the syntax of the command, supply a change number where needed, then retry the command.

ORA-01977

ORA-01977   **missing thread number**

Cause: The keyword THREAD was specified, but a thread number was not specified.

Action: Check the syntax of the command and retry it.

ORA-01978

ORA-01978   **missing sequence number**

Cause: The keyword SEQUENCE was specified, but a sequence number was not specified.

Action: Check the syntax of the command and retry it.

ORA-01979

ORA-01979 **missing or invalid password for role** *name*

Cause: An attempt was made to enable a role without giving the proper password.

Action: Use the IDENTIFIED BY clause in the SET ROLE command to specify the correct password.

ORA-01980

ORA-01980   error during OS ROLE initialization

Cause: An operating system error occurred while loading a user'soperating system roles.

Action: Check the operating system error and correct it.

ORA-01981

ORA-01981 **CASCADE CONSTRAINTS must be specified to perform this revoke**

Cause: For this revoke, some foreign key constraints must be removed. To perform this automatically, CASCADE CONSTRAINTS can be specified as an option with the REVOKE command.

Action: Remove the constraints manually or specify CASCADE CONSTRAINTS.

ORA-01982

ORA-01982   **invalid auditing option for tables**

Cause: The AUDIT or NOAUDIT command specified either REFERENCES or EXECUTE as an option. Those options are not legal for tables.

Action: Do not specify those auditing options on tables.

ORA-01983

ORA-01983 **invalid auditing option for DEFAULT**

Cause: An illegal option was specified with DEFAULT.

Action: Do not specify the REFERENCES option with DEFAULT.

ORA-01984

ORA-01984   **invalid auditing option for procedures/packages/functions**

Cause: An illegal option was specified with DEFAULT for a procedure, package, or function.

Action: The only legal auditing option with a default for procedures, packages, and functions is EXECUTE. Do not specify any option other than EXECUTE.

ORA-01985

ORA-01985   **cannot create user as LICENSE_MAX_USERS parameter exceeded**

Cause: The maximum number of database users allowed by the database license has been created.

Action: Either drop existing users or contact your Oracle Corporation sales representative to upgrade your site license.

ORA-01986

ORA-01986   invalid option for OPTIMIZER_GOAL

Cause: An invalid OPTIMIZER_GOAL option is present.

Action: Check the syntax for OPTIMIZER_GOAL, specify a valid option, and re-execute the statement.

ORA-01987

ORA-01987 client O/S user name is too long

Cause: A client's O/S user name is too long for the O/S logon to succeed.

Action: Use a shorter O/S user name.

ORA-01988

ORA-01988   remote O/S logon is not allowed

Cause: A remote O/S logon was attempted when it was not allowed.

Action: Use a local client or use the REMOTE_OS_AUTHENT initialization parameter to turn on remote O/S logon.

ORA-01989

ORA-01989   **role *name* not authorized by operating system**

Cause: The specified operating system role does not exist, is not granted to you, or you did not provide the correct password.

Action: Retry the SET ROLE using a valid operating system role and password. See your DBA if you are not granted sufficient privileges.

ORA-01990

ORA-01990   **error opening password file *name***

Cause: An attempt to open the specified password file failed because of one of the following errors:

- An operating system error occurred.
- ORAPWD is already running, so the password file is locked.
- The environment variables ORA_*sid*_PWFILE and ORA_PWFILE are not set.

Action: Check for the above causes and correct as necessary.

ORA-01991

ORA-01991   invalid password file *name*

Cause: The specified password file name is invalid.

Action: Specify a correct password file name.

ORA-01992

ORA-01992   **error closing password file** *name*

Cause: The password file could not be closed because of an operating system error or because an authentication action failed.

Action: Check for the above causes and correct as necessary.

ORA-01993

ORA-01993   **error writing password file *name***

Cause: The password file could not be written to because of an operating system error or because an authentication action failed.

Action: Check for the above causes and correct as necessary.

ORA-01994

ORA-01994   **GRANT failed: cannot add users to public password file**

Cause: A grant failed because a user could not be added to the password file. This is because the value of the REMOTE_LOGIN_PASSWORDFILE initialization parameter is set to PUBLIC.

Action: To add a user to the password file, shutdown the database, change the REMOTE_LOGIN_PASSWORDFILE initialization parameter to PRIVATE, and restart the database.

ORA-01995

ORA-01995 **error reading password file** *name*

Cause: An operating system error occurred when trying to read the password file.

Action: Fix the operating system error.

ORA-01996

ORA-01996 **GRANT failed: password file *name* is full**

Cause: The grant failed when a user could not be added to the password file, because all available slots in it were used already.

Action: Either recreate a larger password file or free up slots in the password file by revoking the SYSOPER and SYSDBA privileges from some of the remote users. You can determine which remote users are valid and their privileges by issuing the following query in SQL*DBA:

SELECT * FROM V$REMOTE_USERS;

ORA-01997

ORA-01997 **GRANT failed: user *name* is identified externally**

Cause: You cannot grant the SYSOPER or SYSDBA privilege to a user created with the IDENTIFIED EXTERNALLY clause.

Action: If you wish to grant the user the SYSOPER or SYSDBA privileges, drop and recreate the user without the IDENTIFIED EXTERNALLY clause.

ORA-01998

ORA-01998 **REVOKE failed: user SYS always SYSOPER and SYSDBA**

Cause: You attempted to revoke one of the necessary privileges, SYSOPER or SYSDBA, from user SYS. These privileges cannot be revoked.

Action: Do not attempt this operation.

ORA-01999

ORA-01999   **password file mode has changed from *mode* to *mode***

Cause: Another instance has started with the FORCE_PASSWORD_FILEMODE initialization parameter, which is incompatible with the startup mode of your instance.

Action: Shut down your instance and start up later when the password file changes are completed for the other instance.

ORA-02000

ORA-02000   **missing keyword** *name*

Cause: The statement entered requires the missing keyword.

Action: Use the specified keyword.

ORA-02002

ORA-02002   user SYS is not permitted to create indexes with freelist groups

Cause: An attempt was made to create an index while connected as SYS.

Action: Connect as another user and retry the command.

ORA-02002

ORA-02002   **error while writing to audit trail**

Cause: The auditing facility is unable to write to the AUDIT_TRAIL table. If this error occurs, SQL statements that are currently being audited may also fail. This error will occur if the SYSTEM tablespace runs out of disk space.

Action: Add space to the SYSTEM tablespace or delete rows from the AUDIT_TRAIL table. If these operations fail or do not eliminate the problem, shut down and restart Oracle with auditing disabled. This is done by setting the initialization parameter AUDIT_TRAIL to FALSE.

ORA-02003

ORA-02003   invalid USERENV parameter

Cause: An invalid parameter was specified for the USERENV function.

Action: Specify one of the allowable parameters TERMINAL, SESSIONID, ENTRYID, or NLS_LANG.

ORA-02004

ORA-02004   security violation

Cause: A user is attempting to perform an operation without the proper permission, and this fact is being recorded in the AUDIT_TRAIL table. This message only appears in the AUDIT_TRAIL table's return code column.

Action: For auditing information only; no action required.

ORA-02005

ORA-02005 **implicit (-1) length not valid for this bind or define datatype**

Cause: A negative length for the define variable was passed to a define function.

Action: An explicit, non-negative, length parameter must be passed.

ORA-02006

ORA-02006   **invalid packed decimal format string**

Cause: A packed decimal datatype was passed to a bind or define function, and the format string parameter was either not present or invalid.

Action: A valid format string parameter must be passed for a packed decimal variable. Check the programming manual for the definition of a packed decimal format string.

ORA-02008

ORA-02008 **non-zero scale specified for non-numeric column**

Cause: A scale factor was specified for a bind or define of a non-numeric variable.

Action: Set the scale factor to zero, then retry the operation.

ORA-02009

ORA-02009   **the file size specified for a file must not be zero**

Cause: A file specification included a SIZE clause with a zero value.

Action: If allowed, leave out the size clause or specify a valid value.

ORA-02010

ORA-02010   **missing host connect string**

Cause: A quoted character string does not follow the keyword USING in the USING clause of a CREATE DATABASE LINK statement.

Action: Specify the host connect string, then retry the operation.

ORA-02011

ORA-02011   duplicate database link name

Cause: The database link name specified in a CREATE DATABASE LINK statement already exists.

Action: Either specify a new name or drop the existing link.

ORA-02012

ORA-02012   missing USING keyword

Cause: The keyword USING does not follow the CONNECT TO clause in a CREATE DATABASE LINK statement.

Action: Specify the USING clause after the CONNECT TO clause.

ORA-02013

ORA-02013   missing CONNECT keyword

Cause: The keyword CONNECT does not follow the database link name in a CREATE DATABASE LINK statement.

Action: Specify the CONNECT clause after the database link name.

ORA-02014

ORA-02014 **cannot select FOR UPDATE from view with DISTINCT, GROUP BY, etc.**

Cause: An attempt was made to SELECT... FOR UPDATE from a view whose defining SELECT statement uses at least one of the following constructs in its outer block: outer join, GROUP BY clause, aggregate functions, SELECT DISTINCT, CONNECT BY clause, or set operation, UNION, INTERSECT, or MINUS.

Action: Do not use SELECT... FOR UPDATE on this view.

ORA-02015

ORA-02015   **cannot select FOR UPDATE from remote table**

Cause: An attempt was made to SELECT... FOR UPDATE from a table in a remote database.

Action: It is not currently possible to SELECT... FOR UPDATE from a table in a remote database. Instead, log in or connect directly to the remote database before entering a SELECT... FOR UPDATE on its tables.

ORA-02016

ORA-02016   **cannot use a subquery in a START WITH on a remote database**

Cause: An attempt was made to use a subquery in a START WITH clause. Such a subquery is not currently permitted if the table in the FROM clause of the tree query is in a remote database.

Action: It is necessary to connect directly to the remote database before using a subquery in the START WITH clause.

ORA-02017

ORA-02017 **integer value required**

Cause: The specified value must be an integer.

Action: Specify an appropriate integer value, then retry the operation.

ORA-02018

ORA-02018   **database link of same name has an open connection**

Cause: The user is currently connected to a PUBLIC database link with the same name as the database link name specified in a CREATE or DROP DATABASE LINK statement.

Action: Close the cursors that use the PUBLIC database link.

ORA-02019

ORA-02019 **connection description for remote database not found**

Cause: The user attempted to connect or log in to a remote database using a connection description that could not be found.

Action: Specify an existing database link. Query the data dictionary to see all existing database links. See your operating system-specific SQL*Net documentation for valid connection descriptors.

ORA-02020

ORA-02020   **too many database links in use**

Cause: The maximum number of active connections to remote databases per user login has been reached.

Action: If the user has no open cursors, the current SQL statement accesses more than the maximum allowed remote databases. Otherwise, the user may free remote database connections by closing all cursors that access the databases.

If this occurs often, consider increasing the value of the initialization parameter OPEN_LINKS, which controls the maximum number of concurrent open connections to remote databases per user process.

ORA-02021

ORA-02021   **DDL operations are not allowed on a remote database**

Cause: An attempt was made to use a DDL operation on a remote database. For example, "CREATE TABLE tablename@remotedbname...".

Action: To alter the remote database structure, you must connect to the remote database with the appropriate privileges.

ORA-02022

ORA-02022   **remote statement has unoptimized view with remote object**

Cause: The local view is unoptimized and contains references to objects at the remote database, and the statement must be executed at the remote database.

Action: Create a similar view on the remote database and modify the violating view in the SQL statement with the new *view@remote.*

ORA-02023

ORA-02023   **START WITH or CONNECT BY predicate cannot be evaluated by remote database**

Cause: The statement contains a tree query on a remote database, and the tree query's START WITH or CONNECT BY predicate contains a term that cannot be evaluated at the remote database. Such terms include calls to user functions, calls to USERENV, and references to ROWID.

Action: Remove the disallowed term, directly connect, or log on to the remote database.

ORA-02024

ORA-02024   **database link not found**

Cause: The database link to be dropped is not found in the dictionary.

Action: Correct the database link name.

ORA-02025

ORA-02025   **all tables in the SQL statement must be at the remote database**

Cause: A SQL statement referenced tables from multiple databases, and one or more of the remote databases are not Oracle7 Servers.

Action: Remote updates can be performed only if all databases in the SQL statement are Oracle7 Servers. Update the earlier version databases in a separate statement.

ORA-02026

ORA-02026   **missing LINK keyword**

Cause: The keyword LINK is required in this context.

Action: Check syntax, insert keyword LINK as required, and try again.

ORA-02027

ORA-02027   **multi-row UPDATE of LONG column is not supported**

Cause: A bind variable whose length is greater than 2000 bytes is being used to update a column, and the UPDATE statement affects more than one row.

Action: Check that each such bind variable updates only a single row.

ORA-02028

ORA-02028   **fetching an exact number of rows is not supported by the server**

Cause: The server does not support UPIALL, so the fetch of an exact number of rows cannot be emulated on the user side.

Action: Connect to a valid server or do not use an exact fetch.

ORA-02029

ORA-02029   missing FILE keyword

Cause: The keyword FILE is required in this context.

Action: Check syntax, insert keyword FILE as required, and try again.

ORA-02030

ORA-02030   **can only select from fixed tables/views**

Cause: An operation other than SELECT on a fixed dynamic performance table or view was attempted. It is only possible to select from fixed tables or views.

Action: Remove the fixed table or view name from the SELECT statement.

ORA-02031

ORA-02031   no ROWID for fixed tables

Cause: An attempt was made to select the ROWID from a fixed table.

Action: Do not select ROWID from a fixed table.

ORA-02032

ORA-02032 **clustered tables cannot be used before the cluster index is built**

Cause: An attempt was made to perform a DML operation on a clustered table for which no cluster index has yet been created.

Action: Create a cluster index before referencing clustered tables in a SQL statement.

ORA-02033

ORA-02033 **a cluster index for this cluster already exists**

Cause: A cluster index already exists for the cluster.

Action: No action required.

ORA-02034

ORA-02034　**speed bind not permitted**

Cause:　　Speed bind not allowed with supplied bind variables.

Action:　　No action required.

ORA-02035

ORA-02035   **illegal bundled operation combination**

Cause: UPI bundled execution call was requested on an illegal combination of operations.

Action: Use a legal combination of operations and retry.

ORA-02036

ORA-02036 **too many variables to describe with automatic cursor open**

Cause: UPI-bundled execution call tried to perform automatic cursor open and close on a describe operation. There were too many SELECT list items or bind variables to do this.

Action: Open and close cursor explicitly.

ORA-02037

ORA-02037   **universalized speed bind storage**

Cause: A user attempted to perform a speed-execution call on a SQL statement containing host variables without first performing a speed-bind call.

Action: Perform a speed-bind call before performing a speed-execution call.

ORA-02038

ORA-02038 define is not allowed for array type

Cause: A user attempted to define a select list variable of type "array". Arrays may serve only as host bind variables.

Action: Correct the offending code.

ORA-02039

ORA-02039   **bind by value is not allowed for array type**

Cause: User attempted to bind an array host variable by value. Arrays may be bound only by reference.

Action: Correct the offending code.

02040-02099: Distributed Transactions

This section lists messages generated during distributed transactions.

ORA-02040

ORA-02040 **remote database *name* does not support two-phase commit**

Cause: A distributed update of more than one database was attempted, but the named database does not support the prepare phase of the two-phase commit, as determined by its logon transaction traits. The transaction was rolled back.

Action: Do not attempt to update the named database, unless it is the only database updated in the transaction. Distributed updates of more than one database in a single transaction can be performed only if all databases support the two-phase commit mechanism.

ORA-02041

ORA-02041   **client database did not begin a transaction**

Cause: An update occurred at a coordinated database without the coordinator beginning a distributed transaction. This may happen if a stored procedure commits and then performs updates, and the stored procedure is invoked remotely. It could also happen if an external transaction monitor violates the XA protocol.

Action: If the cause is the former, check that any commit is not followed by an update.

ORA-02042

ORA-02042   **too many distributed transactions**

Cause: The distributed transaction table is full because too many distributed transactions are active.

Action: Increase the DISTRIBUTED_TRANSACTIONS parameter in the initialization parameter file, shut down and restart Oracle, or run fewer transactions. If it is certain there are not too many concurrent distributed transactions, this may be an internal error. In this case, contact customer support. Shutting down and restarting the instance could be a work-around.

ORA-02043

ORA-02043 **must end current transaction before executing** *command*

Cause: A transaction is in progress and one of the following commands is issued: COMMIT FORCE, ROLLBACK FORCE, or ALTER SYSTEM ENABLE DISTRIBUTED RECOVERY in single process mode.

Action: COMMIT or ROLLBACK the current transaction and retry the command.

ORA-02044

ORA-02044   **transaction manager login denied: transaction in progress**

Cause: A remote transaction manager tried to log in while a distributed transaction is in progress. A protocol error occurred in the remote transaction manager.

Action: End the current transaction.

ORA-02045

ORA-02045   **too many local sessions participating in global transactions**

Cause: There are too many sessions at this site to accommodate this transaction.

Action: Use an existing database link so that another session need not be created at the remote site.

ORA-02046

ORA-02046   **distributed transaction already begun**

Cause: This is an internal error not normally encountered. A server session received a BEGIN_TRAN remote procedure call before finishing with a previous distributed transaction.

Action: Contact customer support.

ORA-02047

ORA-02047   **cannot join the distributed transaction in progress**

Cause: This can happen only when attempting to update an Oracle Version 6 and an Oracle7 database in the same transaction, because either

- A transaction is in progress against a remote database that does not support two-phase commit.
- Updates are pending and an attempt was made to update a different database that does not support two-phase commit.

Action: Complete the current transaction before attempting the action that caused the error.

ORA-02048

ORA-02048   **attempt to begin distributed transaction without logging on**

Cause: The client program did not issue a distributed transaction login.

Action: This is an internal error, contact customer support.

ORA-02049

ORA-02049 **time-out: distributed transaction waiting for lock**

Cause: The time to wait on a lock in a distributed transaction has been exceeded. This time is specified in the initialization parameter DISTRIBUTED_LOCK_TIMEOUT.

Action: This situation is treated as a deadlock and the statement was rolled back. To set the time-out interval to a longer interval, adjust the initialization parameter DISTRIBUTED_LOCK_TIMEOUT, then shut down and restart the instance.

ORA-02050

ORA-02050 **transaction *num* rolled back, some remote DBs may be in-doubt**

Cause: There was a network failure or a remote failure during the two-phase commit.

Action: Notify the database administrator. The remote databases will automatically resynchronize when the failure is repaired.

ORA-02051

ORA-02051   another session in same transaction failed

Cause: A session at the same site with the same global transaction identification failed.

Action: No action is necessary; the transaction will automatically recover.

ORA-02052

ORA-02052 **remote transaction failure at** *name*

Cause: There was some error in a remote transaction at the named database link.

Action: More descriptive messages will follow. Correct the problem specified in the following messages and retry the operation.

ORA-02053

ORA-02053   **transaction *num* committed, some remote DBs may be in-doubt**

Cause: There was a network failure or a remote failure during the two-phase commit.

Action: Notify the database administrator. The remote databases will resynchronize automatically when the failure is repaired.

ORA-02054

ORA-02054 **transaction *num* in-doubt**

Cause: There was a network failure or a remote failure in the two-phase commit.

Action: Notify the database administrator. The remote databases will resynchronize automatically when the failure is repaired. Monitor the DBA_2PC_INDOUBT and DBA_2PC_INCONSISTENT tables for the outcome and resubmit the transaction, if necessary.

ORA-02055

ORA-02055   **distributed update operation failed; rollback required**

Cause: A failure during a distributed update operation may not have rolled back all effects of the operation. Since some sites may be inconsistent with others, the transaction must roll back to a savepoint or roll back entirely.

Action: Roll back to a savepoint or roll back the entire transaction and resubmit the rolled-back statements.

ORA-02056

ORA-02056   **2PC: *str*: bad two-phase command number *num* from *name***

Cause: There was a two-phase commit protocol error.

Action: The transaction may automatically recover. Monitor the DBA_2PC_INDOUBT and DBA_2PC_INCONSISTENT tables to determine the resolution. Enter the transaction again, if necessary. Contact customer support, if necessary.

ORA-02057

ORA-02057   **2PC: *str*: bad two-phase recovery state number *num* from *name***

Cause: An internal error in the two-phase recovery protocol occurred.

Action: Contact customer support.

ORA-02058

ORA-02058   no prepared transaction found with ID *num*

Cause: A COMMIT FORCE was attempted on a transaction, but the transaction with LOCAL_TRAN_ID or GLOBAL_TRAN_ID was not found in the DBA_2PC_INDOUBT table in prepared state.

Action: Check the DBA_2PC_INDOUBT table to ensure the proper transaction ID is used and attempt the commit again.

ORA-02059

ORA-02059   **ORA-CRASH-TEST-n in commit comment**

Cause: This is a special comment used to test the two-phase commit.

Action: Do not use this special comment unless working with customer support.

ORA-02060

ORA-02060   **SELECT FOR UPDATE specified a join of non-collocated tables**

Cause: An attempt was made to join tables in different remote databases in a SELECT... FOR UPDATE statement.

Action: Joined tables in a SELECT statement with the FOR UPDATE clause must reside at the same database. Simplify the SELECT... FOR UPDATE statement so that it joins only tables in the same database.

ORA-02061

ORA-02061   **lock table specified list of non-collocated tables**

Cause: An attempt was made to lock more than one table in different remote databases in a LOCK TABLE statement. Tables in a LOCK TABLE statement must reside at the same database.

Action: Issue multiple LOCK TABLE commands.

ORA-02062

ORA-02062   **distributed recovery received DBID *num*, expected *num***

Cause: A database link at the coordinator point no longer points to the expected database. The database link may have been redefined, or a different database may have been mounted.

Action: Check the database link at the coordinator point to see if it was redefined. If so, contact the database administrator for the remote database to check whether the link still references a valid database. If the link changed, recreate it to reference a valid database at the remote site.

ORA-02063

ORA-02063 **preceding *str* from *name***

Cause: An Oracle error message was received from a remote database link.

Action: Refer to the preceding messages. Correct the problem mentioned in the preceding messages and try the operation again.

ORA-02064

ORA-02064   iterated or long remote update with subquery not supported

Cause: One of the following unsupported operations was attempted:
-            an array execute of a remote update with a subquery that references a database link
-            an update of a long column with bind variable and an update of a second column with a subquery that references both a database link and abind variable
-            a commit in a coordinated session issued from a remote procedural call with OUT parameters

Action: Simplify the remote update statement.

ORA-02065

ORA-02065   illegal option for ALTER SYSTEM

Cause: The option specified for ALTER SYSTEM is not supported.

Action: Refer to *Oracle7 Server SQL Reference* for the proper syntax of the ALTER SYSTEM command.

ORA-02066

ORA-02066   **missing or invalid MTS_DISPATCHERS text**

Cause: A character string literal was expected, but not found, in the ALTER SYSTEM SET MTS_DISPATCHERS command.

Action: Place the string literal containing the dispatcher's specification in the ALTER SYSTEM SET MTS_DISPATCHERS command.

ORA-02067

ORA-02067   **transaction or savepoint rollback required**

Cause: A failure occurred, probably in a trigger or stored procedure with multiple remote updates. Some sites in the transaction may be inconsistent, so the previous Oracle call cannot be executed, and the transaction cannot be committed.

Action: Roll back to a previous savepoint or to the beginning of the transaction. Then resubmit the rolled-back statements.

ORA-02068

ORA-02068   **following severe error from *name***

Cause: A severe error, a disconnect or fatal Oracle error, was received from the given database link. See the accompanying messages for a more detailed description of the problem.

Action: Contact the remote system administrator to determine the cause of the remote problem and the estimated time of the problem's resolution.

ORA-02069

ORA-02069 **global_names parameter must be set to TRUE for this operation**

Cause: A remote mapping of the statement is required but cannot be done because the parameter GLOBAL_NAMES is not set to TRUE.

Action: Issue the statement ALTER SESSION SET GLOBAL_NAMES = TRUE,if possible.

ORA-02070

ORA-02070   **database *"name"* does not support capabilit*y name* in this context**

Cause: The remote database does not support the named capability in the context in which it was used.

Action: Simplify the SQL statement.

ORA-02071

ORA-02071   **error initializing capabilities for remote database name**

Cause: Oracle could not load the capability table from the named remote database.

Action: Contact customer support concerning the remote SQL*Connect product.

ORA-02073

ORA-02073 **sequence numbers not supported in remote updates**

Cause: Sequence numbers may not be used in INSERTS, UPDATES, or DELETES on remote tables.

Action: Perform the operation without specifying a sequence number.

ORA-02074

ORA-02074   cannot COMMIT or ROLLBACK in a distributed transaction

Cause: A COMMIT or ROLLBACK was attempted from a session other than the parent of a distributed transaction.

Action: Only the parent session can issue a COMMIT or ROLLBACK.

ORA-02076

ORA-02076   **sequence not co-located with updated or long column**

Cause: All referenced sequences must be located on the same node as the updated table or the long column.

Action: Correct the problem and then reissue the statement.

ORA-02077

ORA-02077 **selects of long columns must be from co-located tables**

Cause: All tables returning values of long columns in a SELECT statement must be located at the same node.

Action: Correct the problem, then reissue the statement.

ORA-02078

ORA-02078   **invalid setting for ALTER SYSTEM FIXED_DATE**

Cause: An invalid value was used.

Action: Use a correct value.

ORA-02079

ORA-02079 **no new sessions may join a committing distributed transaction**

Cause: A call to UPI2BG was issued in a session for a transaction that has begun to commit in a different branch. That is, a call to UPI2EN was issued for a branch of the same transaction in another session. This can only happen when using an external transaction manager.

Action: Contact customer support.

ORA-02080

ORA-02080   database link is in use

Cause: A transaction is active or a cursor is open on the database link.

Action: Before closing a database link, you must first close all cursors and then end all transactions that use it.

ORA-02081

ORA-02081 **database link is not open**

Cause: An attempt was made to use a database link that is not currently open.

Action: Resolve network problems or contact the remote database administrator to determine if the remote database's problem has been fixed. Contact the local database administrator to determine if network problems have been resolved. Wait until the link is open and try the operation again.

ORA-02082

ORA-02082   a loop-back database link must have a connection qualifier

Cause: An attempt was made to create a database link with the same name as the current database.

Action: This database link needs a trailing qualifier to make the name unique. These qualifiers are operating system-specific. See your operating system-specific Oracle documentation for more information about creating database links.

ORA-02083

ORA-02083   **database name has illegal character *str***

Cause: The database name supplied contains an invalid character. For example, there can be no periods, semicolons, single quotes, double quotes, blanks, or non-printable characters in the database name. See *Oracle7 Server Reference* for naming rules.

Action: Specify a valid database name.

ORA-02084

ORA-02084   **database name is missing a component**

Cause: The complete database name was not given

Action: Specify the full database name.

ORA-02085

ORA-02085 **database link** *name* **connects to** *name*

Cause: The database link attempted to connect to a database with a different name. The name of the database link must be the same name as the name of the database.

Action: Create a database link with the same name as the database to which it connects.

ORA-02086

ORA-02086   **database link name is too long**

Cause: The name of the database or database link is too long. Database and database link names can be no longer than 128 characters.

Action: Specify the correct database or database link name or specify a name with fewer characters.

ORA-02087

ORA-02087   **object locked by another process in same transaction**

Cause: In a Parallel Server environment, a database link that loops back to the same instance is being used, and one session is trying to convert a lock that was obtained by the other session.

Action: Obtain a more restrictive lock first. For example if session 1 needs a SHARE lock, and session 2 needs an EXCLUSIVE lock, make sure that the EXCLUSIVE lock is obtained first. Otherwise, use the same session to access the object.

ORA-02088

ORA-02088   distributed database option not installed

Cause: The distributed database option is not installed at this site, so the requested operation is not possible.

Action: Contact Oracle Corporation about obtaining the distributed database option.

ORA-02089

ORA-02089 **COMMIT is not allowed in a subordinate session**

Cause: COMMIT was issued in a session that is not the two-phase commit global coordinator.

Action: Issue commits only at the global coordinator.

ORA-02091

ORA-02091 **transaction rolled back**

Cause: The distributed transaction is assigned to the system rollback segment and is trying to get into the prepared state, but the required number of non-PREPARED slots are not available, so the transaction is rolled back.

If the transaction aborted at a remote site, only ORA-02091 will be seen. If the transaction aborted at the host site, then both ORA-02091 and ORA-02092 will be seen.

Action: Complete the current transaction and then do the operation required.

ORA-02092

ORA-02092 **out of transaction table slots for distributed transaction**

Cause: The distributed transaction is assigned to the system rollback segment and is trying to get into the prepared state, but the required number of non-PREPARED slots are not available, so the transaction is rolled back.

If the transaction aborted at a remote site, only ORA-02091 will be seen. If the transaction aborted at the host site, then both ORA-2091 and ORA-2092 willbe seen.

Cause: Add a rollback segment and do the operation again.

ORA-02093

ORA-02093 **TRANSACTIONS_PER_ROLLBACK_SEGMENT** *name* **more than maximum possible.**

Cause: The value of the initialization parameter specified is greater than allowed on this port.

Action: Use the default value or reduce it to less than the maximum.

ORA-02094

ORA-02094   **replication option not installed**

Cause: The replication option was not installed at this site. Updatable snapshots, deferred remote procedure calls, and other replication features were, therefore, unavailable.

Action: Install the replication option. The replication option is not part of the Oracle7 Server product and must be purchased separately. Contact an Oracle sales representative if the replication option needs to be purchased.

02100-02139: Precompiler Runtime SQLLIB

The messages in this range have the same cause and action as listed in Chapter 4. For example, ORA-2100 is the same as SQL-2100.

02140-02299: SQL Parsing

This section lists some of the messages generated when SQL statements are parsed by the Oracle Server.

ORA-02140

ORA-02140 **invalid tablespace name**

Cause: A valid tablespace name does not follow ALTER TABLESPACE.

Action: Specify a valid tablespace name following ALTER TABLESPACE.

ORA-02141

ORA-02141 invalid OFFLINE option

Cause: An option other than NORMAL or IMMEDIATE follows OFFLINE in an ALTER TABLESPACE statement.

Action: The user must either specify no option following OFFLINE or one of these options: NORMAL or IMMEDIATE.

ORA-02142

ORA-02142 **missing or invalid ALTER TABLESPACE option**

Cause: An invalid option was specified after ALTER TABLESPACE.

Action: Use one of the valid options, such as ADD DATAFILE, RENAME DATAFILE, DEFAULT STORAGE, ONLINE, OFFLINE, BEGIN BACKUP, or END BACKUP.

ORA-02143

ORA-02143   **invalid STORAGE option**

Cause: An option other than INITIAL, NEXT, MINEXTENTS, MAXEXTENTS, FREELISTS, FREELIST GROUPS, or PCTINCREASE was specified in an ALTER INDEX statement or in the USING INDEX clause of an ALTER SNAPSHOT statement.

Action: Check syntax, specify only valid options, and try again.

ORA-02144

ORA-02144   no option specified for ALTER CLUSTER

Cause: No valid ALTER CLUSTER options were specified.

Action: Specify one or more valid options, such as PCTFREE, PCTUSED, SIZE, INITRANS, MAXTRANS, or STORAGE.

ORA-02145

ORA-02145   missing STORAGE option

Cause: No STORAGE options were specified following the keyword STORAGE.

Action: Check syntax, specify one or more valid STORAGE options, and try again.

ORA-02146

ORA-02146 SHARED specified multiple times

Cause: The SHARED option was specified more than once in a CREATEDATABASE statement.

Action: Only specify the SHARED option once.

ORA-02147

ORA-02147 conflicting SHARED/EXCLUSIVE options

Cause: Both the SHARED and EXCLUSIVE options were specified in a CREATE DATABASE statement.

Action: Specify SHARED or EXCLUSIVE, but not both.

ORA-02148

ORA-02148   **EXCLUSIVE specified multiple times**

Cause: The EXCLUSIVE option was specified in a CREATE DATABASE statement more than once.

Action: Only specify the EXCLUSIVE option once.

ORA-02155

ORA-02155   **invalid DEFAULT tablespace identifier**

Cause: A valid tablespace name does not follow DEFAULT TABLESPACE.

Action: Place a valid tablespace name after DEFAULT TABLESPACE and try again.

ORA-02156

ORA-02156   invalid TEMPORARY tablespace identifier

Cause: A valid tablespace name does not follow TEMPORARY TABLESPACE.

Action: Place a valid tablespace name after TEMPORARY TABLESPACE and try again.

ORA-02157

ORA-02157   no options specified for ALTER USER

Cause: No options were specified.

Action: Specify at least one ALTER USER option and try again.

ORA-02158

ORA-02158   invalid CREATE INDEX option

Cause: An option other than PCTFREE, INITRANS, MAXTRANS, NOSORT, STORAGE, or TABLESPACE was specified.

Action: Specify one of the valid CREATE INDEX options and try again.

ORA-02161

ORA-02161 **invalid value for MAXLOGFILES**

Cause: A number does not follow MAXLOGFILES.

Action: Specify a number after MAXLOGFILES and try again.

ORA-02162

ORA-02162   invalid value for MAXDATAFILES

Cause: A number does not follow MAXDATAFILES.

Action: Specify a number after MAXDATAFILES and try again.

ORA-02163

ORA-02163   invalid value for FREELIST GROUPS

Cause: A number does not follow FREELIST GROUPS.

Action: Specify a number after FREELIST GROUPS and try again.

ORA-02164

ORA-02164 **DATAFILE clause specified more than once**

Cause: The CREATE DATABASE statement contains more than one DATAFILE clause.

Action: Specify at most one DATAFILE clause and try again.

ORA-02165

ORA-02165   **invalid option for CREATE DATABASE**

Cause: An invalid CREATE DATABASE option is present.

Action: Specify only valid CREATE DATABASE options and try again.

ORA-02166

ORA-02166 **ARCHIVELOG and NOARCHIVELOG specified**

Cause: Both ARCHIVELOG and NOARCHIVELOG are specified in a CREATE DATABASE statement.

Action: Specify either of these two options, but not both, and try again.

ORA-02167

ORA-02167 **LOGFILE clause specified more than once**

Cause: The CREATE DATABASE statement contains more than one LOGFILE clause.

Action: Specify at most one LOGFILE clause and try again.

ORA-02168

ORA-02168   invalid value for FREELISTS

Cause: A number does not follow the FREELISTS storage option of the CREATE TABLE or CREATE INDEX statement.

Action: Specify a number after FREELISTS and try again.

ORA-02169

ORA-02169 **FREELISTS storage option not allowed**

Cause: An attempt was made to specify the FREELISTS storage option when performing an operation other than CREATE TABLE or CREATE INDEX.

Action: Check the syntax of the statement, remove this option, then re-execute the statement.

ORA-02170

ORA-02170 FREELIST GROUPS storage option not allowed

Cause: An attempt was made to specify the FREELIST GROUPS storage option. This option may be specified only when using the CREATE TABLE command and when the ALLOW_FREELIST_GROUPS parameter is specified in the initialization parameter file.

Action: Retry the statement without the FREELIST GROUPS storage option or set the ALLOW_FREELIST_GROUPS parameter in the initialization parameter file.

ORA-02171

ORA-02171   **invalid value for MAXLOGHISTORY**

Cause: A number does not follow MAXLOGHISTORY.

Action: Specify a number after MAXLOGHISTORY.

ORA-02172

ORA-02172   **the PUBLIC keyword is not appropriate for a disable thread**

Cause: The PUBLIC keyword was specified when disabling a thread.

Action: Check the syntax of the command and retry.

ORA-02173

ORA-02173 **invalid option for DROP TABLESPACE**

Cause: Something other than INCLUDING CONTENTS was found following the tablespace name, or text other than CASCADE CONSTRAINTS was found following INCLUDING CONTENTS.

Action: Place either nothing or only INCLUDING CONTENTS after the tablespace name, or place either nothing or only CASCADE CONSTRAINTS after INCLUDING CONTENTS and try again.

ORA-02174

ORA-02174   missing required thread number

Cause: A thread number must be specified after THREAD keyword.

Action: Use correct syntax.

ORA-02175

ORA-02175   invalid rollback segment name

Cause: In the CREATE or DROP ROLLBACK SEGMENT statements, a valid rollback segment name does not follow ROLLBACK SEGMENT.

Action: Place a valid rollback segment name after ROLLBACK SEGMENT and try again.

ORA-02176

ORA-02176   invalid option for CREATE ROLLBACK SEGMENT

Cause: An invalid option was specified in a CREATE ROLLBACKSEGMENT statement.

Action: Specify one of the valid options, TABLESPACE or STORAGE, and try again.

ORA-02177

ORA-02177 **missing required group number**

Cause: A group number must be specified after GROUP keyword.

Action: Check the syntax of the command, specify a group number after GROUP, and retry it.

ORA-02178

ORA-02178   correct syntax is: SET TRANSACTION READ { ONLY | WRITE }

Cause: There is a syntax error in the user's statement.

Action: Correct the syntax as indicated and try again.

ORA-02180

ORA-02180   **invalid option for CREATE TABLESPACE**

Cause: An invalid option follows CREATE TABLESPACE.

Action: Specify one of the valid options: DATAFILE, DEFAULT STORAGE, ONLINE, or OFFLINE, and try again.

ORA-02181

ORA-02181   invalid option to ROLLBACK WORK

Cause: A word other than TO follows ROLLBACK [WORK].

Action: Place either nothing or TO SAVEPOINT after ROLLBACK [WORK] and try again.

ORA-02182

ORA-02182   **savepoint name expected**

Cause: A valid savepoint name does not follow ROLLBACK [WORK] TO [SAVEPOINT].

Action: Place a savepoint name following TO [SAVEPOINT] and try again.

ORA-02184

ORA-02184 **resource quotas are not allowed in REVOKE**

Cause: In a revoke statement, a resource quota was specified.

Action: Specify the resource privilege without the quota and try again.

ORA-02185

ORA-02185   a token other than WORK follows COMMIT

Cause: A token other than WORK follows COMMIT.

Action: Place either nothing or WORK after COMMIT and try again.

ORA-02186

ORA-02186   **tablespace resource privilege may not appear with other privileges**

Cause: An attempt was made to grant or revoke a resource quota in the same statement in which other privileges are granted or revoked.

Action: Use a separate GRANT or REVOKE statement for the resource quota privilege and try again.

ORA-02187

ORA-02187   **invalid quota specification**

Cause: In a GRANT or REVOKE statement, an attempt was made to grant a tablespace quota above the upper limit.

Action: Grant a smaller tablespace quota and try again.

ORA-02189

ORA-02189   **required clause: ON <tablespace>**

Cause: The GRANT or REVOKE statement specifying a tablespace quota requires the ON tablespace clause.

Action: Correct the syntax and try again.

ORA-02190

ORA-02190   **keyword TABLES expected**

Cause: The keyword TABLES is expected following DROP CLUSTER *cluster name* INCLUDING.

Action: Place the keyword TABLES after INCLUDING and try again.

ORA-02191

ORA-02191   correct syntax is: SET TRANSACTION USE ROLLBACK SEGMENT name

Cause: There is a syntax error in the SET TRANSACTION statement.

Action: Correct the syntax error and retry the statement.

ORA-02192

ORA-02192 **PCTINCREASE not allowed for rollback segment storage clauses**

Cause: The storage parameter PCTINCREASE was specified in a CREATE or ALTER ROLLBACK SEGMENT command.

Action: Retry the command without the PCTINCREASE clause.

ORA-02194

ORA-02194   **event specification syntax error** *num* **minor error** *num* **near** *name*

Cause: There is a syntax error in an event specification.

Action: Check syntax and spelling and try again.

ORA-02197

ORA-02197   **file list already specified**

Cause: In a CREATE TABLESPACE statement, more than one DATAFILE clause was specified.

Action: Merge the DATAFILE clauses into a single clause specifying multiple files and try again.

ORA-02198

ORA-02198 ONLINE/OFFLINE option already specified

Cause: In a CREATE TABLESPACE statement, the ONLINE and/or OFFLINE options were specified more than once.

Action: Specify either ONLINE or OFFLINE, but not both, and try again.

ORA-02199

ORA-02199   missing DATAFILE clause

Cause: A CREATE TABLESPACE statement has no DATAFILE clause.

Action: Specify the missing DATAFILE clause and try again.

ORA-02200

ORA-02200   WITH GRANT OPTION not allowed for PUBLIC

Cause: An attempt was made to GRANT to PUBLIC WITH GRANT OPTION.

Action: Remove either the PUBLIC or the WITH GRANT OPTION clause and try again.

ORA-02201

ORA-02201 **sequence not allowed here**

Cause: An attempt was made to reference a sequence in a FROM list. This is not permitted. A sequence can only be referenced in a SELECT list.

Action: Check the statement syntax, correct the reference, and try again.

ORA-02202

ORA-02202   **no more tables permitted in this cluster**

Cause: An attempt was made to create a table in a cluster that already contains 32 tables. No more than 32 tables may be stored in a cluster.

Action: Do not try to cluster more than 32 tables.

ORA-02203

ORA-02203   **INITIAL storage options not allowed**

Cause: The user attempted to alter the INITIAL storage option of a table, cluster, index, or rollback segment. This option may be specified only when the object is created.

Action: Remove the INITIAL option and try again.

ORA-02204

ORA-02204   ALTER, INDEX, REFERENCE, and EXECUTE not allowed for views

Cause: An attempt was made to grant or revoke an invalid privilege on a view.

Action: Do not attempt to grant or revoke the privileges ALTER, INDEX, or REFERENCES on views.

ORA-02205

ORA-02205   only SELECT and ALTER privileges are valid for sequences

Cause: An attempt was made to grant or revoke an invalid privilege on a sequence.

Action: Do not attempt to grant or revoke the privileges DELETE, INDEX, INSERT, UPDATE, REFERENCES, or EXECUTE on sequences.

ORA-02206

ORA-02206   **duplicate INITRANS option specification**

Cause: INITRANS is specified more than once.

Action: Specify INITRANS only once and try again.

ORA-02207

ORA-02207   invalid INITRANS option value

Cause: The INITRANS value is not an integer between 2 and 255, or it is not less than or equal to the MAXTRANS value.

Action: Specify a valid INITRANS value and try again.

ORA-02208

ORA-02208   duplicate MAXTRANS option specification

Cause: The MAXTRANS option is specified more than once.

Action: Specify MAXTRANS only once and try again.

ORA-02209

ORA-02209   invalid MAXTRANS option value

Cause: The MAXTRANS value is not an integer between 2 and 255, or it is not greater than or equal to the INITRANS value.

Action: Specify a valid MAXTRANS value and try again.

ORA-02210

ORA-02210   no options specified for ALTER TABLE

Cause: No ALTER TABLE option was specified.

Action: Specify at least one ALTER TABLE option and try again.

ORA-02211

ORA-02211   **invalid value for PCTFREE or PCTUSED**

Cause: The specified value for PCTFREE or PCTUSED is not an integer between 0 and 100.

Action: Specify an appropriate value for the option and try again.

ORA-02212

ORA-02212   duplicate PCTFREE option specification

Cause: The PCTFREE option was specified more than once.

Action: Specify the PCTFREE option only once and try again.

ORA-02213

ORA-02213   **duplicate PCTUSED option specification**

Cause: The PCTUSED option was specified more than once.

Action: Specify the PCTUSED option only once and try again.

ORA-02214

ORA-02214 duplicate BACKUP option specification

Cause: The BACKUP option to ALTER TABLE was specified more than once.

Action: Specify the BACKUP option only once and try again.

ORA-02215

ORA-02215 **duplicate tablespace name clause**

Cause: There is more than one TABLESPACE clause in one of the following statements: GRANT/REVOKE RESOURCE, CREATE TABLE, CREATE INDEX, CREATE/DROP TABLESPACE, CREATE CLUSTER, and CREATE ROLLBACK SEGMENT.

Action: Check the statement syntax, specify only one TABLESPACE clause, and try again.

ORA-02216

ORA-02216	tablespace name expected

Cause: A valid tablespace name is not present where required in one of the following statements: CREATE TABLE, CREATE INDEX, CREATE/DROP TABLESPACE, CREATE CLUSTER, and CREATE ROLLBACK SEGMENT.

Action: Check the statement syntax, specify a valid tablespace name where required, and try again.

ORA-02217

ORA-02217 **duplicate storage option specification**

Cause: A storage option, such as INITIAL, NEXT, MINEXTENTS, MAXEXTENTS, or PCTINCREASE, was specified more than once.

Action: Specify storage options only once and try again.

ORA-02218

ORA-02218   **invalid INITIAL storage option value**

Cause: The specified value must be an integer.

Action: Specify an appropriate integer value and try again.

ORA-02219

ORA-02219 **invalid NEXT storage option value**

Cause: The specified value must be an integer.

Action: Specify an appropriate integer value and try again.

ORA-02220

ORA-02220   **invalid MINEXTENTS storage option value**

Cause: The specified value must be a positive integer that is less than or equal to MAXEXTENTS.

Action: Specify an appropriate integer value and try again.

ORA-02221

ORA-02221   **invalid MAXEXTENTS storage option value**

Cause: The specified value must be a positive integer greater than or equal to MINEXTENTS.

Action: Specify an appropriate integer value and try again.

ORA-02222

ORA-02222 **invalid PCTINCREASE storage option value**

Cause: The specified value must be a positive integer.

Action: Specify an appropriate integer value and try again.

ORA-02223

ORA-02223 **invalid OPTIMAL storage option value**

Cause: The specified value must be an integer.

Action: Choose an appropriate integer value for the OPTIMAL storage option.

ORA-02224

ORA-02224 **EXECUTE privilege not allowed for tables**

Cause: An attempt was made to GRANT or REVOKE an invalid privilege on a table.

Action: Do not attempt to GRANT or REVOKE the EXECUTE privilege on tables.

ORA-02225

ORA-02225 **only EXECUTE privilege is valid for procedures**

Cause: An attempt was made to GRANT or REVOKE an invalid privilege on a procedure, function, or package.

Action: Do not attempt to GRANT or REVOKE any privilege other than EXECUTE on procedures, functions, or packages.

ORA-02226

ORA-02226 **invalid MAXEXTENTS value max allowed:** *num*

Cause: The value specified for MAXEXTENTS is too large for the database block.

Action: Specify a value for MAXEXTENTS that is smaller than the number given in the message.

ORA-02227

ORA-02227   **invalid cluster name**

Cause: A cluster name was not properly formed.

Action: Check the rules for forming object names and enter an appropriate cluster name.

ORA-02228

ORA-02228 duplicate SIZE specification

Cause: The SIZE option is specified more than once.

Action: Specify the SIZE option only once and try again.

ORA-02229

ORA-02229   **invalid SIZE option value**

Cause: The specified value must be an integer number of bytes.

Action: Specify an appropriate integer value and try again.

ORA-02230

ORA-02230   **invalid ALTER CLUSTER option**

Cause: An option other than PCTFREE, PCTUSED, INITRANS, MAXTRANS, STORAGE, or SIZE was specified in an ALTER CLUSTER statement.

Action: Specify only valid options in the ALTER CLUSTER statement and try again.

ORA-02231

ORA-02231   missing or invalid option to ALTER DATABASE

Cause: An invalid option was specified in the statement.

Action: Check the syntax of the ALTER DATABASE command. Specify only valid options in the ALTER DATABASE statement and try again.

ORA-02232

ORA-02232   invalid MOUNT mode

Cause: A mode other than SHARED or EXCLUSIVE follows the keyword MOUNT in an ALTER DATABASE statement.

Action: Specify either SHARED, EXCLUSIVE, or nothing following the keyword MOUNT in the ALTER DATABASE statement and try again.

ORA-02233

ORA-02233 invalid CLOSE mode

Cause: A mode other than NORMAL or IMMEDIATE follows the keyword CLOSE in an ALTER DATABASE statement.

Action: Specify either NORMAL, IMMEDIATE, or nothing following the keyword CLOSE in the ALTER DATABASE statement and try again.

ORA-02236

ORA-02236 invalid filename

Cause: In a LOGFILE, DATAFILE, or RENAME clause, a character string literal was expected, but not found, in a filename list.

Action: Specify filenames using character string literals and try again.

ORA-02237

ORA-02237   **invalid file size**

Cause: An integer file size does not follow SIZE in a LOGFILE or DATAFILE file list.

Action: Specify an appropriate file size following the keyword SIZE and try again.

ORA-02238

ORA-02238   **filename lists have different numbers of files**

Cause: In a RENAME clause in ALTER DATABASE or TABLESPACE, the number of existing filenames does not equal the number of new filenames.

Action: Make sure a new filename corresponds to each existing filename and try again.

ORA-02239

ORA-02239   **there are objects that reference this sequence**

Cause: The sequence to be dropped is still referenced by other objects.

Action: Make sure the sequence name is correct or drop the constraint or object that references the sequence.

ORA-02240

ORA-02240   invalid value for OBJNO or TABNO

Cause: A number does not follow OBJNO or TABNO.

Action: Specify a number after OBJNO or TABNO.

ORA-02241

ORA-02241 **must be of form EXTENTS FILE *num* BLOCK *num* SIZE *num*, ...**

Cause: There was an error in the extent storage clause.

Action: Respecify the storage clause using the correct syntax and retry the command.

ORA-02242

ORA-02242   no options specified for ALTER INDEX

Cause: No options were specified after ALTER INDEX.

Action: Specify the INITRANS, MAXTRANS, or STORAGE option in the ALTER INDEX statement and try again.

ORA-02243

ORA-02243   **invalid ALTER INDEX or ALTER SNAPSHOT option**

Cause: An option other than INITRANS, MAXTRANS, or STORAGE was specified in an ALTER INDEX statement or in the USING INDEX clause of an ALTER SNAPSHOT statement.

Action: Specify only valid options and try again.

ORA-02244

ORA-02244   invalid ALTER ROLLBACK SEGMENT option

Cause: An option other than STORAGE was found.

Action: Specify the STORAGE option and try again.

ORA-02245

ORA-02245   **invalid ROLLBACK SEGMENT name**

Cause: A rollback segment name was expected, but not found, following ALTER [PUBLIC] ROLLBACK SEGMENT.

Action: Place a rollback segment name after SEGMENT and try again.

ORA-02246

ORA-02246   missing EVENTS text

Cause: A character string literal was expected, but not found, following the ALTER SESSION SET EVENTS command.

Action: Place the string literal containing the EVENTS text after the keyword EVENTS and try again.

ORA-02247

ORA-02247   **no option specified for ALTER SESSION**

Cause: The option SET EVENTS was expected, but not found, following the ALTER SESSION command.

Action: Place the SET EVENTS option after the ALTER SESSION command and try again.

ORA-02248

ORA-02248   invalid option for ALTER SESSION

Cause: An option other than SET EVENTS was found following the ALTER SESSION command.

Action: Specify the SET EVENTS option after the ALTER SESSION command and try again.

ORA-02249

ORA-02249   **missing or invalid value for MAXLOGMEMBERS**

Cause: A valid number does not follow MAXLOGMEMBERS. The value specified must be between 1 and the operating system-specific maximum number of log file members.

Action: Specify a valid number after MAXLOGMEMBERS.

ORA-02250

ORA-02250   **missing or invalid constraint name**

Cause: The constraint name is missing or invalid.

Action: Specify a valid constraint name and try again.

ORA-02251

ORA-02251   **subquery not allowed here**

Cause: A subquery is not allowed at this point in the statement.

Action: Remove or relocate the subquery and try again.

ORA-02252

ORA-02252   check constraint condition not properly ended

Cause: The specified search condition for the check constraint is not properly ended.

Action: Check and correct the search condition's syntax. Then retry the operation.

ORA-02253

ORA-02253   constraint specification not allowed here

Cause: A constraint specification is not allowed at this point in the statement.

Action: Remove or relocate the constraint specification and try again.

ORA-02254

ORA-02254 **DEFAULT <expression> not allowed here**

Cause: A default-value expression is not allowed for the column at this point in the statement.

Action: Remove or relocate the default-value expression and try again.

ORA-02255

ORA-02255 **NOT NULL not allowed after DEFAULT NULL**

Cause: A NOT NULL specification conflicts with the NULL default value.

Action: Remove either the NOT NULL or the DEFAULT NULL specification and try again.

ORA-02256

ORA-02256   **number, type, and size of referencing columns must matchreferenced columns**

Cause: The number of columns in the foreign-key referencing list is not equal to the number of columns in the referenced list, or the datatype or size of at least one referencing column is not equal to the datatype or size of its referenced column.

Action: Ensure that the referencing columns match the referenced columns.

ORA-02257

ORA-02257   maximum number of columns exceeded

Cause: The number of columns in the key list exceeds the maximum number.

Action: Reduce the number of columns in the list.

ORA-02258

ORA-02258   duplicate or conflicting NULL and/or NOT NULL specifications

Cause: A duplicate or conflicting NULL and/or NOT NULL was specified.

Action: Remove one of the conflicting specifications and try again.

ORA-02259

ORA-02259   duplicate UNIQUE/PRIMARY KEY specifications

Cause: A duplicate unique or primary key was specified.

Action: Remove the duplicate specification and try again.

ORA-02260

ORA-02260   **table can have only one primary key**

Cause: Two or more primary keys were specified for the same table.

Action: Remove the extra primary keys and try again.

ORA-02261

ORA-02261   **such unique or primary key already exists in the table**

Cause: A unique or primary key was specified that already exists for the table.

Action: Remove the extra key and try again.

ORA-02262

ORA-02262 **ORA-*num* occurs while type-checking column default value expression**

Cause: An attempt to alter a column's datatype caused a type-checking error because the new datatype conflicted with the existing column's default value expression.

Action: Remove the default value expression or do not alter the column's datatype, and try again.

ORA-02263

ORA-02263   **need to specify the datatype for this column**

Cause: The required datatype for the column is missing.

Action: Specify the required datatype and try again.

ORA-02264

ORA-02264   *name* **already used by an existing constraint**

Cause: The specified constraint name is not unique.

Action: Specify a unique constraint name for the constraint and try again.

ORA-02265

ORA-02265   **cannot derive the datatype of the referencing column**

Cause: The datatype of the referenced column is not defined as yet.

Action: Make sure that the datatype of the referenced column is defined before referencing it.

ORA-02266

ORA-02266   **unique/primary keys in table referenced by enabled foreign keys**

Cause: An attempt was made to drop or truncate a table with unique or primary keys referenced by foreign keys enabled in another table.

Action: Before dropping or truncating the table, disable the foreign key constraints in other tables. You can see what constraints are referencing a table by issuing the following command:

SELECT * FROM USER_CONSTRAINTS WHERE TABLE_NAME = "tabnam";

ORA-02267

ORA-02267   **column type incompatible with referenced column type**

Cause: An attempt was made to perform an operation on a pair of columns with incompatible datatypes.

Action: Specify a compatible datatype for the referencing column.

ORA-02268

ORA-02268   **referenced table does not have a primary key**

Cause: The referenced table does not have a primary key.

Action: Do not attempt to reference the table using a unique key, or create a unique key for the table.

ORA-02269

ORA-02269   **key column cannot be of LONG datatype**

Cause: An attempt was made to define a key column of datatype LONG. This is not allowed.

Action: Change the datatype of the column or remove the LONG column from the key, and try again.

ORA-02270

ORA-02270   **no matching unique or primary key for this column-list**

Cause: An attempt was made to reference a unique or primary key in a table with a CREATE or ALTER TABLE statement when no such key exists in the referenced table.

Action: Add the unique or primary key to the table or find the correct names of the columns with the primary or unique key, and try again.

ORA-02271

ORA-02271   table does not have such a constraint

Cause: An attempt was made to reference a table using a constraint that does not exist.

Action: Check the spelling of the constraint name or add the constraint to the table, and try again.

ORA-02272

ORA-02272   **constrained column cannot be of LONG datatype**

Cause: A constrained column cannot be defined as datatype LONG. This is not allowed.

Action: Change the datatype of the column or remove the constraint on the column, and try again.

ORA-02273

ORA-02273   **this unique/primary key is referenced by some foreign keys**

Cause: A unique or primary key referenced by foreign keys cannot be dropped.

Action: Remove all references to the key before dropping it.

ORA-02274

ORA-02274   duplicate referential constraint specifications

Cause: A referential constraint was specified more than once. This is not allowed.

Action: Remove the duplicate specification.

ORA-02275

ORA-02275   **such a referential constraint already exists in the table**

Cause: An attempt was made to specify a referential constraint that already exists. This would result in duplicate specifications and so is not allowed.

Action: Be sure to specify a constraint only once.

ORA-02276

ORA-02276   **default value type incompatible with column type**

Cause: The type of the evaluated default expression is incompatible with the datatype of the column.

Action: Modify the default expression or change the column's datatype, and try again.

ORA-02277

ORA-02277   **invalid sequence name**

Cause: An invalid sequence name was specified.

Action: Check syntax and spelling, specify a valid sequence name, and try again.

ORA-02278

ORA-02278   duplicate or conflicting MAXVALUE/NOMAXVALUE specifications

Cause: Duplicate or conflicting MAXVALUE and/or NOMAXVALUE specifications.

Action: Remove one of the conflicting specifications and try again.

ORA-02279

ORA-02279   duplicate or conflicting MINVALUE/NOMINVALUE specifications

Cause: Duplicate or conflicting MINVALUE and/or NOMINVALUE clauses were specified.

Action: Remove one of the conflicting specifications and try again.

ORA-02280

ORA-02280   **duplicate or conflicting CYCLE/NOCYCLE specifications**

Cause: Duplicate or conflicting CYCLE and/or NOCYCLE clauses were specified.

Action: Remove one of the conflicting specifications and try again.

ORA-02281

ORA-02281 **duplicate or conflicting CACHE/NOCACHE specifications**

Cause: Duplicate or conflicting CACHE and/or NOCACHE clauses were specified.

Action: Remove one of the conflicting specifications and try again.

ORA-02282

ORA-02282   duplicate or conflicting ORDER/NOORDER specifications

Cause: Duplicate or conflicting ORDER and/or NOORDER clauses were specified.

Action: Remove one of the conflicting specifications and try again.

ORA-02283

ORA-02283   cannot alter starting sequence number

Cause: An attempt was made to alter a starting sequence number. This is not allowed.

Action: Do not try to alter a starting sequence number.

ORA-02284

ORA-02284   duplicate INCREMENT BY specifications

Cause: A duplicate INCREMENT BY clause was specified.

Action: Remove the duplicate specification and try again.

ORA-02285

ORA-02285 **duplicate START WITH specifications**

Cause: A duplicate START WITH clause was specified.

Action: Remove the duplicate specification and try again.

ORA-02286

ORA-02286   **no options specified for ALTER SEQUENCE**

Cause: No ALTER SEQUENCE option was specified.

Action: Check the syntax. Then specify at least one ALTER SEQUENCE option.

ORA-02287

ORA-02287   sequence number not allowed here

Cause: The specified sequence number reference, CURRVAL or NEXTVAL, is inappropriate at this point in the statement.

Action: Check the syntax. Then remove or relocate the sequence number.

ORA-02288

ORA-02288   **invalid OPEN mode**

Cause: A mode other than RESETLOGS was specified in an ALTER DATABASE OPEN statement. RESETLOGS is the only valid OPEN mode.

Action: Remove the invalid mode from the statement or replace it with the keyword RESETLOGS, and try again.

ORA-02289

ORA-02289   **sequence does not exist**

Cause: The specified sequence does not exist, or access privilege is required for this operation. Also, this error can occur if attempting to access a remote sequence through an invalid or non-existent database link.

Action: Check spelling of the sequence name or obtain the required access privilege. If necessary, create the sequence. All remote sequences accessed in a distributed transaction must be on the same node.

ORA-02290

ORA-02290   **check constraint *str.name* violated**

Cause: The value or values attempted to be entered in a field or fields violate a defined check constraint.

Action: Enter values that satisfy the constraint.

ORA-02291

ORA-02291 integrity constraint *str.name* violated - parent key not found

Cause: An attempt was made to INSERT or UPDATE a foreign key value. The result was a value that is not in the parent key.

Action: UPDATE to or INSERT a value that is in the parent key.

ORA-02292

ORA-02292   **integrity constraint *str.name* violated - child record found**

Cause: An attempt was made to delete a row that is referenced by a foreign key.

Action: It is necessary to DELETE or UPDATE the foreign key before changing this row.

ORA-02293

ORA-02293   **cannot enable *name* - check constraint violated**

Cause: An attempt was made via an ALTERTABLE statement to add a check constraint to a populated table that had no complying values.

Action: Retry the ALTER TABLE statement, specifying a check constraint on a table containing complying values.

ORA-02294

ORA-02294   **cannot add referential constraint - parent keys not found**

Cause: An attempt was made via an ALTERTABLE statement to add a referential constraint to a populated table that had no matching parent values.

Action: Retry the ALTER TABLE statement, specifying a valid primary key in a parent table.

ORA-02295

ORA-02295   **found more than one enable/disable clause for constraint**

Cause: An attempt was made via a CREATE or ALTER TABLE statement to specify more than one ENABLE and/or DISABLE clause for a given constraint.

Action: Only one ENABLE or DISABLE clause may be specified for a given constraint.

ORA-02296

ORA-02296   **cannot enable *name* - null values found**

Cause: An ALTER TABLE command with an ENABLE CONSTRAINT clause failed because the table contains values that do not satisfy the constraint.

Action: Make sure that all values in the table satisfy the constraint before issuing an ALTER TABLE command with an ENABLE CONSTRAINT clause.

ORA-02297

ORA-02297 **cannot disable constraint *name.name* - dependencies exist**

Cause: An alter table disable constraint failed because the table has foreign keys that are dependent on the constraint.

Action: Either disable the foreign key constraints or use a DISABLECASCADE command.

ORA-02298

ORA-02298 **cannot enable *name* - parent keys not found**

Cause: An ALTER TABLE ENABLE CONSTRAINT command failed because the table has orphaned child records.

Action: Make sure that the table has no orphaned child records before issuing an ALTER TABLE ENABLE CONSTRAINT command.

ORA-02299

ORA-02299   **cannot enable *name* - duplicate keys found**

Cause: An ALTER TABLE ENABLE CONSTRAINT command failed because the table has duplicate key values.

Action: Make sure that the table has no duplicate key values before issuing an ALTER TABLE ENABLE CONSTRAINT command.

02351-02375: SQL*Loader in Direct Path Mode

This section lists messages generated when the direct path mode is used to load data into the database. See also range ORA-02476 to ORA-02479 for parallel direct path loader messages.

ORA-02351

ORA-02351 record *num*: Rejected - Error on table *name*, column *name*

Cause: The indicated record caused an error on insert.

Action: Determine the error and correct the record.

ORA-02352

ORA-02352   direct path connection must be homogeneous

Cause: The user-side to kernel-side connection is heterogeneous.

Action: Check that the connection is between identical hardware and operating systems.

ORA-02353

ORA-02353   **multi-byte character error**

Cause: The text specified by POSITION or terminated by new lines is invalid.

Action: Edit the control file or the datafile to fix the alignment of the column.

ORA-02354

ORA-02354   conversion initialization error occurred on field *name*

Cause: An error occurred during conversion initialization for a field.

Action: This is an internal error. Contact customer support.

ORA-02355

ORA-02355   **conversion error occurred on CONSTANT field** *str*

Cause: A CONSTANT field caused an error. All rows will be rejected.

Action: Determine the Oracle error and correct the record.

ORA-02356

ORA-02356   **the database is out of space. The load cannot continue**

Cause: The load was discontinued due to space exhaustion in the database.

Action: Add space for the specified table.

ORA-02357

ORA-02357   **packed decimal conversion error**

Cause: The column could not be converted from packed decimal to character.

Action: Check the column and make it conform to packed decimal format.

ORA-02358

ORA-02358 **zoned decimal conversion error**

Cause: The column could not be converted from packed decimal to character.

Action: Check the column and make it conform to packed decimal format.

ORA-02359

ORA-02359   **field in datafile exceeded maximum specified length**

Cause: The field is too long for the specified column.

Action: Shorten the field so that the length of the field equals or is less than the column length.

ORA-02360

ORA-02360   column not found before end of logical record (use TRAILING NULLCOLS)

Cause: A column started after the end of the logical record had been reached.

Action: Use the TRAILING NULLCOLS option or supply data for the column.

ORA-02361

ORA-02361   **initial enclosing character not found**

Cause: The initial enclosing character of an enclosed field was not found.

Action: Supply the character before the start of the field.

ORA-02362

ORA-02362   **logical record ended - second enclosing character not present**

Cause: The second enclosing character of an enclosed field was not found.

Action: Supply the character after the end of the field.

ORA-02363

ORA-02363   no terminator found after TERMINATED and ENCLOSED field

Cause: No terminator was found after a terminated and enclosed field.

Action: Supply the terminator after the end of the field.

ORA-02364

ORA-02364   record *num*: Discarded - failed all WHEN clauses

Cause: The record was not loaded because it failed the WHEN clauses of all the tables.

Action: No action required.

ORA-02365

ORA-02365   **index *name* was left in Direct Load State due to**

Cause: The index was not loaded. The cause of the error is stated after the message.

Action: Correct the cause of the error, then reload.

ORA-02366

ORA-02366 **the following index or indexes on table *name* were processed**

Cause: The indexes listed for the named table were loaded.

Action: No action required.

ORA-02367

ORA-02367   **index *name* was loaded**

Cause: The index specified in the message was loaded.

Action: No action required.

ORA-02368

ORA-02368   **record *num*: Discarded - all columns null.**

Cause: The record was not loaded because all of the columns in the table were null for that record.

Action: No action required.

ORA-02369

ORA-02369   warning: Variable-length field was truncated

Cause: A variable-length field was truncated by the end of the logical record.

Action: Although this is a warning, a variable length field has the length embedded in the data for each row, so check the data.

ORA-02370

ORA-02370 **record** *num* **- Warning on table** *name***, column** *num*

Cause: The record indicated caused a warning.

Action: Determine the warning and correct the record, if necessary.

ORA-02371

ORA-02371   loader must be at least version *num* for direct path

Cause: The version of SQL*Loader being used is incompatible with this version of Oracle.

Action: Upgrade the version of SQL*Loader to at least the specified version or use the conventional path.

ORA-02372

ORA-02372	**relative start position > absolute field end position**

Cause: The relative start of a field specified as POSITION(*+n:y) occurred after the absolute position y.

Action: Check that the values of n and y are correct and that the statement was entered correctly. Also check that all opening delimiters have matching closing delimiters.

ORA-02373

ORA-02373   **Error parsing insert statement for table** *name*

Cause: A parsing error occurred.

Action: Check the message following this parsing error message.

ORA-02374

ORA-02374   **No more slots for read buffer queue**

Cause: There are no free read buffers.

Action: Use the READBUFFERS keyword in the loader to increase the number of read buffers.

02376-02399: Oracle Resources

This section lists messages generated when resources within the Oracle Server are altered.

ORA-02376

ORA-02376 **invalid or redundant resource**

Cause: A resource that is not defined or that is specified twice appears in the CREATE or ALTER PROFILE statement.

Action: Define the resource before issuing a CREATE or ALTER PROFILE command. Also check the statement to see that each resource is listed only once.

ORA-02377

ORA-02377   **invalid resource limit**

Cause: An invalid resource limit of 0 was specified.

Action: Specify a resource limit greater than 0.

ORA-02378

ORA-02378   **duplicate resource name** *name*

Cause: The same resource was specified twice in a CREATE or ALTER PROFILE statement.

Action: Issue the statement again, carefully checking that each resource is used only once.

ORA-02379

ORA-02379 **profile** *name* **already exists**

Cause: An attempt was made to create a profile that already exists.

Action: Create the profile with a name not already used by another profile.

ORA-02380

ORA-02380	**profile *name* does not exist**

Cause: An attempt was made to assign a user to a non-existent profile.

Action: Assign the user to an existing profile.

ORA-02381

ORA-02381   cannot drop PUBLIC_DEFAULT profile

Cause: An attempt was made to drop the PUBLIC_DEFAULT profile.

Action: The PUBLIC_DEFAULT profile cannot be dropped.

ORA-02382

ORA-02382   **profile *name* has users assigned, cannot drop without CASCADE**

Cause: An attempt was made to drop a profile that had users assigned to it without using the CASCADE option of the DROP PROFILE statement. A profile that has users assigned to it cannot be dropped, unless the CASCADE option is specified. CASCADE reassigns users to the PUBLIC_DEFAULT profile before dropping the profile.

Action: To reassign all users to the PUBLIC_DEFAULT profile, issue the DROP PROFILE statement with the CASCADE option.

ORA-02383

ORA-02383   **illegal cost factor**

Cause: A negative value or UNLIMITED cost was specified for this resource in the ALTER RESOURCE COST statement.

Action: Only positive integer values can be specified for resources in the ALTER RESOURCE COST statement. Issue the statement again, using a positive integer value for each resource specified.

ORA-02390

ORA-02390   **exceeded COMPOSITE_LIMIT, logoff in progress**

Cause: The COMPOSITE_LIMIT for the profile is exceeded. That is, the weighted sum of the connection time, logical reads per session, CPU usage per session, and private SGA space used during the session exceeded the limit set by the COMPOSITE_LIMIT clause set in the user profile.

Action: If this happens often, ask the database administrator to raise the COMPOSITE_LIMIT of the user profile, or determine which resource is used the most and raise the limit on that resource.

ORA-02391

ORA-02391   **exceeded simultaneous SESSIONS_PER_USER limit, maximum *num***

Cause: An attempt was made to exceed the maximum number of concurrent sessions allowed by the SESSIONS_PER_USER clause of the user profile.

Action: End one or more concurrent sessions or ask the database administrator to increase the SESSIONS_PER_USER limit of the user profile.

ORA-02392

ORA-02392   **exceeded session limit on CPU usage, logging off**

Cause: An attempt was made to exceed the maximum CPU usage allowed by the CPU_PER_SESSION clause of the user profile.

Action: If this happens often, ask the database administrator to increase the CPU_PER_SESSION limit of the user profile.

ORA-02393

ORA-02393   **exceeded call limit on CPU usage**

Cause: An attempt was made to exceed the maximum CPU time for a call, a parse, execute, or fetch, allowed by the CPU_PER_CALL clause of the user profile.

Action: If this happens often, ask the database administrator to increase the CPU_PER_CALL limit of the user profile.

ORA-02394

ORA-02394   **exceeded session limit on I/O usage, logging off**

Cause: An attempt was made to exceed the maximum I/O allowed by the LOGICAL_READS_PER_SESSION clause of the user profile.

Action: If this happens often, ask the database administrator to increase the LOGICAL_READS_PER_SESSION limit of the user profile.

ORA-02395

ORA-02395   **exceeded call limit on I/O usage**

Cause: An attempt was made to exceed the maximum I/O for a call, a parse, execute, or fetch, allowed by the LOGICAL_READS_PER_CALL clause of the user profile.

Action: If this happens often, ask the database administrator to increase the LOGICAL_READS_PER_CALL limit of the user profile.

ORA-02396

ORA-02396   exceeded max Idle Time, please connect again

Cause: A user has exceeded the maximum time allowed to remain idle.

Action: The user must reconnect to the database.

ORA-02397

ORA-02397   exceeded PRIVATE_SGA Limit, logging off

Cause: This error occurs only when using a multi-threaded server.

Action: Contact the database administrator to expand the PRIVATE_SGA limit.

ORA-02398

ORA-02398   **exceeded procedure space usage**

Cause: Stored procedures used up too much space in the SYSTEM tablespace.

Action: Use fewer stored procedures or add an additional file to the system tablespace.

ORA-02399

ORA-02399   **exceeded maximum connect time, logging off**

Cause: A user has exceeded the maximum time allowed to be connected tothe database.

Action: The user must reconnect to the database.

02400-02419: EXPLAIN PLAN Command

This sections lists messages generated when using the SQL EXPLAINPLAN command.

ORA-02401

ORA-02401   **cannot EXPLAIN view owned by another user**

Cause: The view specified in the user's SQL statement belongs to another user and cannot be explained.

Action: Create a view that has the same definition but belongs to the current user.

ORA-02402

ORA-02402 **PLAN_TABLE not found**

Cause: The PLAN_TABLE used by EXPLAIN to describe the SQL statement does not exist in the current schema.

Action: Create a PLAN_TABLE in the current schema or use the INTO clause to store the EXPLAIN results in a plan table with a different name.

ORA-02403

ORA-02403   **plan table does not have correct format**

Cause: The plan table does not have the appropriate columns.

Action: Redefine the plan table with the appropriate columns.

ORA-02404

ORA-02404   **specified plan table not found**

Cause: The specified plan table cannot be found.

Action: Create the specified plan table or use an existing one.

02420-02429: Schemas

This section lists messages generated when commands are used that alter schemas or schema objects.

ORA-02420

ORA-02420   missing schema authorization clause

Cause: The schema AUTHORIZATION clause is missing in a CREATE SCHEMA statement.

Action: Check the syntax of the statement, provide a valid schema AUTHORIZATION clause, and retry the statement.

ORA-02421

ORA-02421   **missing or invalid schema authorization identifier**

Cause: Either:

- The schema authorization identifier is not recognized.
- An attempt was made to use a schema authorization identifier for a user while not connected as that user.

Action: If the cause is

- an unrecognized identifier, then use an existing identifier.
- connected with a different user name, then disconnect and connect to the correct user name.

ORA-02422

ORA-02422 **missing or invalid schema element**

Cause: A statement other than a CREATE TABLE, CREATE VIEW, or GRANT appears in a CREATE SCHEMA statement.

Action: See *Oracle7 Server SQL Reference* for the valid elements of the CREATE SCHEMA statement.

ORA-02423

ORA-02423   **schema name does not match schema authorization identifier**

Cause: A table definition with a schema name prepended to the table name does not match the schema name provided in the AUTHORIZATION clause of a CREATE SCHEMA STATEMENT.

Action: Check that the schema names match the one given in the CREATE SCHEMA statement.

ORA-02424

ORA-02424   potential circular view references or unknown referenced tables

Cause: The CREATE SCHEMA statement contains a view that depends on other views contained in the CREATE SCHEMA statement, or they contain references to non-existing tables.

Action: Create the dependent views in a separate CREATE SCHEMA statement and ensure that all referenced tables are either defined in the CREATE SCHEMA statement or exist outside the statement.

ORA-02425

ORA-02425   **create table failed**

Cause: A CREATE TABLE statement failed in the CREATE SCHEMA statement.

Action: The cause for failure will be presented below this message. Follow the appropriate actions as suggested by the subsequent messages.

ORA-02426

ORA-02426   privilege grant failed

Cause: A GRANT statement failed in the CREATE SCHEMA statement.

Action: The cause for failure will be presented below this message. Follow appropriate actions, as suggested by the subsequent messages.

ORA-02427

ORA-02427   **create view failed**

Cause: A CREATE VIEW statement failed in the CREATE SCHEMA statement.

Action: The cause for failure will be presented below this message. Follow appropriate actions, as suggested by the subsequent messages.

ORA-02428

ORA-02428   **could not add foreign key reference**

Cause: Oracle could not add a foreign key reference because of an error in the declaration. Either the referenced table does not exist or the table does not have a unique key.

Action: Check that the referenced table exists and/or has a unique key.

ORA-02429

ORA-02429 **cannot drop index used for enforcement of unique/primary key**

Cause: An attempt was made to drop an index that is being used as the enforcement mechanism for a unique or primary key.

Action: Drop the integrity constraint instead of dropping the index.

02430-02449: Constraint Enabling & Disabling

This section lists messages generated when commands are used that affect constraints on a table.

ORA-02430

ORA-02430 **cannot enable constraint *name* - no such constraint**

Cause: The named constraint does not exist for this table.

Action: Check that a constraint exists before trying to enable it.

ORA-02431

ORA-02431   **cannot disable constraint** *name* **- no such constraint**

Cause: The named constraint does not exist for this table.

Action: Check that a constraint exists before trying to disable it.

ORA-02432

ORA-02432 **cannot enable primary key - primary key not defined for table**

Cause: An attempt was made to enable a primary key that is not defined for the table.

Action: Add a primary key definition for the table.

ORA-02433

ORA-02433   **cannot disable primary key - primary key not defined for table**

Cause: An attempt was made to disable a primary key that is not defined for the table.

Action: Check that a primary key exists before trying to disable it.

ORA-02434

ORA-02434   **cannot enable unique(*str*) - unique key not defined for table**

Cause: An attempt was made to enable a unique key that is not defined for the table.

Action: Check that a unique key exists before trying to enable it.

ORA-02435

ORA-02435 **cannot disable unique *str* - unique key not defined for table**

Cause: An attempt was made to disable a unique key that is not defined for the table.

Action: Check that a unique key exists before trying to disable it.

ORA-02436

ORA-02436   **date or system variable wrongly specified in CHECK constraint**

Cause: An attempt was made to use a date constant or system variable, such as USER, in a check constraint that was not completely specified in a CREATE TABLE or ALTER TABLE statement. For example, a date was specified without the century.

Action: Completely specify the date constant or system variable.

ORA-02437

ORA-02437 **cannot enable** *name* **- primary key violated**

Cause: An attempt was made to enable a primary key constraint on a column that contains either duplicate values or null.

Action: Remove the duplicate values or null before enabling the primary key constraint.

ORA-02438

ORA-02438   **column check constraint cannot reference other columns**

Cause: An attempt was made to define a column check constraint that references another column.

Action: This is not permitted for column check constraints. Create this constraint as a table check constraint.

ORA-02439

ORA-02439   **non-unique index exists on unique/primary key constraint**

Cause: An attempt was made to enable a primary key or unique constraint on a column that has an existing non-unique index.

Action: Drop the non-unique index on the column or do not attempt to enable this constraint.

ORA-02440

ORA-02440   **create as select with referential constraints not allowed**

Cause: Integrity constraints on a table cannot be created when creating the table using the CREATE TABLE... AS SELECT... form.

Action: First, create the table, and then alter the table to add the desired integrity constraints.

ORA-02441

ORA-02441   **cannot drop non-existent primary key**

Cause: An attempt was made to drop the primary key constraint on a table, but there is no primary key constraint on this table.

Action: Ensure the correct table name is entered in the statement. Otherwise, no action required.

ORA-02442

ORA-02442 **cannot drop non-existent unique key**

Cause: An attempt was made to drop a unique key constraint for a column that does not have a unique key constraint.

Action: Make certain the correct column name was entered. Otherwise, no action required.

ORA-02443

ORA-02443   **cannot drop constraint - non-existent constraint**

Cause: An attempt was made to drop a constraint that does not exist.

Action: Make certain the constraint and table name are correct and attempt the procedure again.

ORA-02444

ORA-02444   **cannot resolve referenced object in referential constraints**

Cause: An attempt was made to define a foreign key with a reference to a schema object that cannot be resolved to a base table reference.

Action: Define referential constraints only with schema objects resolving to a base table.

ORA-02445

ORA-02445   **exceptions table not found**

Cause: The explicitly or implicitly declared exceptions table does not exist.

Action: If the correct exceptions table name was used, then create the table and retry the enable command.

ORA-02446

ORA-02446   CREATE TABLE ... AS SELECT failed - check constraint violated

Cause: An attempt was made to use a CREATE TABLE... AS SELECT statement when some rows violated one or more CHECK constraints.

Action: Do not select rows that violate constraints.

02450-02479: Hash Cluster Commands

This section lists messages that occur when commands are used that affect hash clusters.

ORA-02450

ORA-02450   **invalid hash option - missing keyword IS**

Cause: The IS keyword is missing from the CREATE CLUSTER command.

Action: Check the syntax of the command and retry the statement.

ORA-02451

ORA-02451   **duplicate HASHKEYS specification**

Cause: The HASHKEYS option of the CREATE CLUSTER command is specified more than once.

Action: Check the syntax of the command, make certain to specify the HASHKEYS option only once, then retry the command.

ORA-02452

ORA-02452	invalid HASHKEYS option value

Cause: The value specified for HASHKEYS must be an integer.

Action: Check the syntax of the command, make certain to specify an integer for the HASHKEYS option, then retry the command.

ORA-02453

ORA-02453 **duplicate HASH IS specification**

Cause: The HASH IS option is specified more than once.

Action: Check the syntax of the command, make certain to specify the HASH IS option only once, then retry the command.

ORA-02454

ORA-02454   **number of hash keys per block *num* exceeds maximum of *num***

Cause: The SIZE option specified for this hashed cluster is too small.

Action: Retry the statement using a larger value for the SIZE option.

ORA-02455

ORA-02455 **the number of cluster key column must be 1**

Cause: An attempt was made to create a cluster using the HASH IS option with a number of key columns other than 1. Specify only one key column when using the HASH IS option.

Action: Check the syntax of the command. Either specify only one key column or do not specify the HASH IS option, then retry the command.

ORA-02456

ORA-02456 **the HASH IS column specification must be NUMBER(*,0)**

Cause: The column specification in the command must specify an integer.

Action: Check the syntax of the command, make certain to specify an integer for the column specification, then retry the command.

ORA-02457

ORA-02457   **the HASH IS option must specify a valid column**

Cause: An attempt was made to create a cluster with the HASH IS option without specifying a valid column name.

Action: Check the syntax of the command. Specify a valid column name in the statement, then retry the statement.

ORA-02458

ORA-02458 HASHKEYS must be specified for a HASH CLUSTER

Cause: An attempt was made to create a hash cluster without specifying the HASHKEYS option in the statement.

Action: Check the syntax of the command, specify the HASHKEYS option, and retry the command.

ORA-02459

ORA-02459   **hashkey value must be a positive integer**

Cause: The value specified for HASHKEYS was not a positive integer.

Action: Check the syntax of the command, specify only positive integers for the HASHKEYS option, then retry the command.

ORA-02460

ORA-02460   **inappropriate index operation on a hash cluster**

Cause: An attempt was made to create a cluster index on a hash cluster.

Action: Creation of cluster indexes on hash clusters is not allowed. Do not attempt to create this index.

ORA-02461

ORA-02461   **inappropriate use of the INDEX option**

Cause: The INDEX option cannot be specified for hash clusters.

Action: Check the syntax of the command. Do not specify the INDEX option with hash clusters.

ORA-02462

ORA-02462   duplicate INDEX option specified

Cause: The INDEX option is specified more than once.

Action: Check the syntax of the command, correct the problem, then retry the statement.

ORA-02463

ORA-02463   duplicate HASH IS option specified

Cause: The HASH IS option is specified more than once.

Action: Check the syntax of the command, correct the problem, then retry the statement.

ORA-02464

ORA-02464   cluster definition cannot be both HASH and INDEX

Cause: Both the HASH IS and INDEX option were specified. Clusters can be hash or indexed, but not both.

Action: Decide on which type of cluster is to be created, check the syntax of the command, then retry the statement.

ORA-02465

ORA-02465   **inappropriate use of the HASH IS option**

Cause: The HASH IS option is valid only for clusters.

Action: Check the syntax of the command, use the HASH IS option only for clusters, then retry the command.

ORA-02466

ORA-02466   **the SIZE option is not allowed to be altered for HASH CLUSTERS**

Cause: An attempt was made to change the SIZE option of a hash cluster after the cluster was created.

Action: The SIZE option can be specified only when creating the hash cluster. Do not attempt to alter the SIZE of the cluster.

ORA-02467

ORA-02467   **column referenced in hash expression not present in cluster**

Cause: A column referenced in a hash expression is not present in the cluster definition.

Action: Recreate the cluster and correct the error in the hash expression.

ORA-02468

ORA-02468   **constant or system variable wrongly specified in expression**

Cause: A constant or system variable is specified in the hash expression.

Action: Recreate the cluster and correct the error in the hash expression.

ORA-02469

ORA-02469   **hash expression does not return an Oracle Number**

Cause: The result of evaluating a hash expression is not an Oracle Number.

Action: Recreate the cluster and correct the error in the hash expression.

ORA-02470

ORA-02470   TO_DATE, USERENV, or SYSDATE incorrectly used in hash expression

Cause: TO_DATE, USERENV, and SYSDATE are not allowed in hash expressions.

Action: Recreate the cluster and correct the error in the hash expression.

ORA-02471

ORA-02471 **SYSDATE, UID, USER, ROWNUM, OR LEVEL incorrectly used in hash expression**

Cause: SYSDATE, UID, USER, ROWNUM, OR LEVEL are not allowed in hash expression(s).

Action: Recreate the cluster and correct the error in the hash expression.

ORA-02472

ORA-02472 **PL/SQL functions not allowed in hash expressions**

Cause: A PL/SQL function is used in a hash expression.

Action: Recreate the cluster and remove the PL/SQL function.

ORA-02473

ORA-02473   **error while evaluating the cluster's hash expressions**

Cause: An error occurred while evaluating the cluster's hash expression.

Action: Correct the query and retry.

02476-02479: Parallel Direct Loader

This section lists messages generated when the parallel direct path mode is used to load data into the database. See also range ORA-02351 to ORA-02375 for direct path loader messages.

ORA-02476

ORA-02476   **cannot create index due to parallel direct load on table**

Cause: A parallel direct load is occurring to the specified table.

Action: Retry the statement after the load is complete.

ORA-02477

ORA-02477   **cannot perform parallel direct load on object "name"**

Cause: A parallel direct load is not possible because an index is being created on the table.

Action: Retry the load after the index creation is complete.

ORA-02478

ORA-02478   **merge into base segment would overflow MAXEXTENTS limit**

Cause: Merge of temporary segment into base segment failed because MAXEXTENTS was larger than the total in the temporary and base segments.

Action: Use a larger value for MAXEXTENTS on the base segment or make the extents in the temporary segments larger.

ORA-02479

ORA-02479   **error while translating file name for parallel load**

Cause: The specified name of the database file to load with data was invalid.

Action: Specify a valid database filename.

ORA-02495

ORA-02495   **cannot resize file *name*, tablespace *name* is read only**

Cause: An attempt was made to resize a datafile in a tablespace that was read only.

Action: Change the tablespace to read/write and retry the resize operation.

02500-02699: CTOS

See your operating system-specific Oracle documentation for information on error messages in this range. For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications" in the Preface.

02700-02874: UNIX Two Task

See the *Oracle7 Server for UNIX Administrator's Reference Guide*. For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications" in the Preface.

02875-02899: IBM RS/6000

See the Oracle7 for IBM RISC System/6000 Installation and Configuration Guide, Oracle7 for MVS Messages Guide, and *Oracle7 for VM Messages Guide.* For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications" in the Preface.

03000-03099: Features Not Implemented

This section lists messages generated when a user tries to access a feature that is not implemented in Oracle.

ORA-03001

ORA-03001   unimplemented feature

Cause: An attempt was made to use a feature that is not currently implemented.

Action: Do not attempt to use the feature at this time.

ORA-03002

ORA-03002   operator not implemented

Cause: An attempt was made to use an operator that is not currently implemented.

Action: Do not attempt to use the operator at this time.

ORA-03007

ORA-03007 **obsolete feature**

Cause: An attempt was made to use a feature that is no longer supported.

Action: No action required.

ORA-03008

ORA-03008 **parameter COMPATIBLE >= *val* needed for *feature_name***

Cause: An attempt was made to use a feature for a later Oracle version than the setting of the initialization parameter, COMPATIBLE.

Action: Set COMPATIBLE to the value in the message, or higher, and retry the command, but be aware that this will limit your downgrade options.

03100-03199: Two-Task Interface

This sections lists messages generated when the two-task interface to Oracle is used.

ORA-03100

ORA-03100 **communication area cannot be allocated; insufficient memory**

Cause: An attempt to allocate stack space for communication purposes failed. This is an internal error message not usually issued.

Action: Contact customer support.

ORA-03106

ORA-03106   **fatal two-task communication protocol error**

Cause: The communication path between Oracle and the user task has stopped. This is an internal error message not usually issued.

Action: Contact customer support.

ORA-03107

ORA-03107   oranet buffer underflow

Cause: This is an internal error message not usually issued.

Action: Contact customer support.

ORA-03108

ORA-03108   **oranet: Oracle does not support this interface version**

Cause: The version of Oracle connected to via SQL*Net does not support this version of the SQL*Net protocol.

Action: Upgrade the older version of Oracle and try again.

ORA-03109

ORA-03109   **oranet buffer overflow**

Cause: Data being transferred between the user program and Oracle overflowed the buffer space available. This is an internal error message not usually issued.

Action: Contact customer support.

ORA-03110

ORA-03110   **oranet: Oracle does not support this SQL version**

Cause: The version of Oracle connected to via SQL*Net does not support the version of the SQL*Net protocol.

Action: Upgrade the older version of Oracle and try again.

ORA-03111

ORA-03111   **break received on communication channel**

Cause: A break was processed on the communications channel, but was not properly handled by SQL*Net software. This is an internal error message not usually issued.

Action: Contact customer support.

ORA-03112

ORA-03112   host string syntax error

Cause: In a single-task operation, the internally generated host string was invalid. This is an internal error message not usually issued.

Action: Contact customer support.

ORA-03113

ORA-03113 **end-of-file on communication channel**

Cause: An unexpected end-of-file was processed on the communication channel. The problem could not be handled by the SQL*Net, two task, software. This message could occur if the shadow two-task process associated with a SQL*Net connect has terminated abnormally, or if there is a physical failure of the interprocess communication vehicle, that is, the network or server machine went down.

Action: If this message occurs during a connection attempt, check the setup files for the appropriate SQL*Net driver and confirm SQL*Net software is correctly installed on the server. If the message occurs after a connection is well established, and the error is not due to a physical failure, check if a trace file was generated on the server at failure time. Existence of a trace file may suggest an Oracle internal error that requires the assistance of customer support.

ORA-03114

ORA-03114 **not connected to Oracle**

Cause: A call to Oracle was attempted when no connection was established. Usually this happens because a user-written program has not logged on. It may also happen if communication trouble causes a disconnection.

Action: Try again. If the message recurs and the program is user-written, check the program.

ORA-03115

ORA-03115   **unsupported network datatype or representation**

Cause: A user bind or define, or an Oracle function, is not supported by this heterogeneous SQL*Net connection.

Action: Upgrade the older version of Oracle and try again.

ORA-03116

ORA-03116 **invalid buffer length passed to a conversion routine**

Cause: This is an internal error message not usually issued.

Action: Contact customer support.

ORA-03117

ORA-03117   **two-task save area overflow**

Cause: This is an internal error message not usually issued.

Action: Contact customer support.

ORA-03118

ORA-03118 **two-task coroutine has invalid state**

Cause: This is an internal error message not usually issued.

Action: Contact customer support.

ORA-03120

ORA-03120 **two-task conversion routine: integer overflow**

Cause: An integer value in an internal Oracle structure overflowed when being sent or received over a heterogeneous connection. This can happen when an invalid buffer length or too great a row count is specified. It usually indicates a bug in the user application.

Action: Check parameters to Oracle calls. If the problem recurs, reduce all integer parameters, column values *not* included, to less than 32767.

ORA-03121

ORA-03121   **no interface driver connected - function not performed**

Cause: This message occurs usually because the SQL*Net driver is not loaded into memory or there is a mismatch of the version of the driver. A user program linked with an obsolete library may also cause this message. Only programs provided by Oracle Corporation should use this interface.

Action: If the message occurred during use of a program provided by Oracle Corporation, contact customer support. If a user program caused the message, relink the program with current versions of the Oracle libraries.

ORA-03122

ORA-03122   **attempt to close Oracle-side window on user side**

Cause: This is an internal error message not usually issued.

Action: Contact customer support.

ORA-03123

ORA-03123 **operation would block**

Cause: The attempted operation cannot complete now.

Action: Retry the operation later.

ORA-03124

ORA-03124   **two-task internal error**

Cause: Internal error.

Action: Contact customer support.

ORA-03125

ORA-03125 **client-server protocol violation**

Cause: The application received a bad escape sequence from the server which may indicate a problem with the client application user code.

Action: Contact your customer support representative.

ORA-03126

ORA-03126 **network drive does not support non-blocking operations**

Cause: A non-block operation was attempted and the network driver does not support non-block operations.

Action: Use the default blocking operations or use a driver supporting non-blocking operations.

ORA-03127

ORA-03127   **no new operations allowed until active non-blocking operation ends**

Cause: An attempt was made to execute a new operation before the active non-blocking operation completed.

Action: Execute the new operation after the non-blocking operation completes.

ORA-03128

ORA-03128   connection is in blocking mode

Cause: The OCI test for non-blocking mode on a connection indicates that the connection is in blocking mode.

Action: If non-blocking mode is required, use appropriate OCI calls to change the mode.

03200-03289: Extent Allocation and Other Space Management Errors

This section lists messages generated during extent allocation.

ORA-03200

ORA-03200   **the segment type specification is invalid**

Cause: The segment type is not TABLE, INDEX, or CLUSTER.

Action: Use a correct segment type.

ORA-03201

ORA-03201   **the group number specification is invalid**

Cause: The freelist group number is either negative or larger than the number of freelist groups in the segment.

Action: Use a correct group number.

ORA-03202

ORA-03202   **the scan limit specification is invalid**

Cause: The scan limit does not have a positive integer value greater than the number of freelist groups in the segment.

Action: Use a correct scan limit.

ORA-03203

ORA-03203   **concurrent update activity makes space analysis impossible**

Cause: High volume of user updates interfere with the space analysis.

Action: Retry the command or lock the underlying objects.

ORA-03276

ORA-03276 **duplicate ALLOCATE EXTENT option specification**

Cause: The ALLOCATE EXTENT option to ALTER TABLE is specified more than once.

Action: Specify the option at most once.

ORA-03277

ORA-03277	**invalid SIZE specified**

Cause: The value specified for the SIZE option of the ALTER TABLE... ALLOCATE EXTENT statement must be a positive integer.

Action: Choose an appropriate integer value.

ORA-03278

ORA-03278 **duplicate option specified for ALLOCATE EXTENT**

Cause: An option, datafile, SIZE, or INSTANCE, was specified more than once in the ALTER TABLE... ALLOCATE EXTENT statement.

Action: Specify each option at most once.

ORA-03279

ORA-03279   invalid INSTANCE specified

Cause: The value specified for the INSTANCE option of the ALTER TABLE... ALLOCATE EXTENT statement is not valid.

Action: Choose an appropriate INSTANCE specification.

ORA-03280

ORA-03280   **invalid datafile filename specified**

Cause: A character-string literal is expected but not found for the DATAFILE option of the ALTER TABLE... ALLOCATE EXTENT statement.

Action: Specify the filename using a character-string literal enclosed in single quotes. Query the data dictionary view DBA_DATA_FILES to find the correct names of the datafiles.

ORA-03281

ORA-03281   **invalid ALLOCATE EXTENT option**

Cause: An option other than DATAFILE, SIZE, or INSTANCE was specified in the ALTER TABLE... ALLOCATE EXTENT statement.

Action: Remove the invalid option and retry the command.

ORA-03282

ORA-03282   missing ALLOCATE EXTENT option

Cause: No ALLOCATE EXTENT options were specified inside the parentheses of the ALTER TABLE... ALLOCATE EXTENT statement.

Action: If no options are required, do not use parentheses. Otherwise, specify one or more of the following options within parentheses: datafile, SIZE,or INSTANCE.

ORA-03283

ORA-03283   **specified datafile *name* does not exist**

Cause: The datafile specified for the DATAFILE option in the ALTER TABLE... ALLOCATE EXTENT statement does not exist.

Action: Retry the option with the correct filename for the database file. Query the data dictionary view DBA_DATA_FILES to find the correct names of the datafiles.

ORA-03284

ORA-03284   **specified datafile *name* is not a member of tablespace *name***

Cause: The datafile specified in the ALTER TABLE... ALLOCATE EXTENT statement does not belong to the tablespace in which the table resides.

Action: Retry the option with the correct datafile or the correct table. Query the data dictionary view DBA_DATA_FILES to find the correct names of the datafiles in that tablespace or query DBA_TABLES to find the correct table name.

ORA-03286

ORA-03286   ALLOCATE EXTENT not valid for HASH CLUSTERS

Cause: The cluster cannot use the allocate extent option because it is a hash cluster.

Action: Do not use this command on a hash cluster.

ORA-03287

ORA-03287   invalid FREELIST GROUP specified

Cause: The specified FREELIST GROUP number is invalid.

Action: Choose a number between 1 and the number of freelist groups for this object.

ORA-03288

ORA-03288 **both FREELIST GROUP and INSTANCE parameters may not be specified**

Cause: Both FREELIST GROUP and INSTANCE were specified in clause.

Action: Remove one of the two parameters.

03290-03295: TRUNCATE Command

This section lists messages generated when the TRUNCATE command is used.

ORA-03290

ORA-03290   invalid truncate command - missing CLUSTER or TABLE keyword

Cause: An invalid object specification was given for the TRUNCATE command.

Action: Check the syntax of the command and retry the operation specifying either TRUNCATE CLUSTER or TRUNCATE TABLE.

ORA-03291

ORA-03291   **invalid truncate option - missing STORAGE keyword**

Cause: The STORAGE keyword was not specified with the DROP or REUSE options of the TRUNCATE command.

Action: Check the syntax of the command and retry the operation.

ORA-03292

ORA-03292   table to be truncated is a member of a cluster

Cause: An attempt was made to truncate a table that is a member of a cluster.

Action: Either truncate the entire cluster or drop and re-create the individual table.

ORA-03293

ORA-03293   **cluster to be truncated is a HASH CLUSTER**

Cause: An attempt was made to truncate a hash cluster. This is not permitted; only index clusters can be truncated.

Action: Drop and re-create the cluster instead of using the TRUNCATE command.

03296-03299: Resizeable Datafiles

ORA-03297

ORA-03297   **file contains *num* blocks of data beyond requested RESIZE value**

Cause: Some portion of the file in the region to be trimmed is currently in use by a database segment.

Action: Drop or move segments containing extents in this region prior to resizing the file, or choose a resize value such that only free space is in the trimmed region.

ORA-03299

ORA-03299   **cannot create dictionary table** *name*

Cause: A dictionary table is created upon the first execution of the command ALTER DATABASE AUTOEXTEND ON for a database. This operation did not succeed. The most probable cause for this is insufficient space in the system tablespace.

Action: See the accompanying messages and take appropriate action.

03300-03499: Meiko

For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications" in the Preface.

03500-03699: Macintosh

See *Oracle7 Server for Macintosh: Error Messages and Codes.* For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications" in the Preface.

03700-03999: AOS/VS

For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications"operating in the Preface.

04000-04019: Command Parameters

This section lists some of the messages generated when inconsistent values are used in a command.

ORA-04000

ORA-04000   **the sum of PCTUSED and PCTFREE cannot exceed 100**

Cause: The sum of the PCTUSED and PCTFREE values specified for a cluster or table exceeds 100.

Action: Create the table or cluster specifying values for PCTUSED and PCTFREE. The sum of these values cannot exceed 100.

ORA-04001

ORA-04001   **sequence parameter *str* must be an integer**

Cause: The specified sequence parameter was not an integer.

Action: Create the sequence, giving the specified parameter an integer value.

ORA-04002

ORA-04002   **INCREMENT must be a non-zero integer**

Cause: A sequence increment was specified as zero.

Action: Specify the increment as a non-zero value.

ORA-04003

ORA-04003 **sequence parameter *num* exceeds the maximum size allowed *num* digits**

Cause: The specified sequence parameter has too many digits.

Action: Specify the parameter with the allowed number of digits and try again.

ORA-04004

ORA-04004   **MINVALUE must be less than MAXVALUE**

Cause: MINVALUE was specified to be greater than or equal to MAXVALUE.

Action: Specify a MINVALUE that is less than MAXVALUE and try again.

ORA-04005

ORA-04005 **INCREMENT must be less than MAXVALUE minus MINVALUE**

Cause: INCREMENT was specified as equal to or greater than MAXVALUE minus MINVALUE. This is not permitted.

Action: Specify INCREMENT as less than MAXVALUE minus MINVALUE and try again.

ORA-04006

ORA-04006 **START WITH cannot be less than MINVALUE**

Cause: The given starting value is less than MINVALUE.

Action: Check that the starting value is not less than MINVALUE and try again.

ORA-04007

ORA-04007   **MINVALUE cannot be made to exceed the current value**

Cause: The given MINVALUE is greater than the current value.

Action: Check that MINVALUE is not greater than the current value and try again.

ORA-04008

ORA-04008   **START WITH cannot be more than MAXVALUE**

Cause: The starting value is larger than MAXVALUE.

Action: Check that the starting value is less than MAXVALUE and try again.

ORA-04009

ORA-04009   **MAXVALUE cannot be made to be less than the current value**

Cause: The current value exceeds the given MAXVALUE.

Action: Check that the new MAXVALUE is larger than the current value and try again.

ORA-04010

ORA-04010   **the number of values to CACHE must be greater than 1**

Cause: The value specified in the CACHE clause of a CREATE SEQUENCE statement is 1. The default value is 20.

Action: Specify NOCACHE or a value greater than 1 and try again.

ORA-04011

ORA-04011   **sequence** *name* **must range between** *num* **and** *num*

Cause: The value specified for one of the sequence parameters exceeds limits.

Action: Specify a parameter within these limits.

ORA-04012

ORA-04012   **object is not a sequence**

Cause: The object specified cannot have sequence operations performed on it.

Action: Re-enter the statement, making sure to spell the sequence name correctly.

ORA-04013

ORA-04013   **number to CACHE must be less than one cycle**

Cause: The number to CACHE specified in a CREATE SEQUENCE statement exceeds the number of values in a cycle.

Action: Enlarge the cycle or cache fewer values, and try again.

ORA-04014

ORA-04014 **descending sequences that CYCLE must specify MIN VALUE**

Cause: A value was not specified in the CREATE SEQUENCE statement. Sequences that cycle must specify a value at which to begin again.

Action: Re-create the sequence, specifying the value at which it is to begin.

ORA-04015

ORA-04015   **ascending sequences that CYCLE must specify MAX VALUE**

Cause: A value was not specified in the CREATE SEQUENCE statement. Sequences that cycle must specify a value at which to begin again.

Action: Recreate the sequence, specifying the value at which it is to begin.

ORA-04016

ORA-04016   **sequence** *name* **no longer exists**

Cause: Sequence was dropped while processing its next value.

Action: Recreate the sequence

04020-04039: Library Object Locks

This section lists messages generated when a library object is locked and cannot be accessed.

ORA-04020

ORA-04020 deadlock detected while trying to lock object *name*

Cause: While trying to lock a library object, a deadlock is detected.

Action: Retry the operation later.

ORA-04021

ORA-04021   **time-out occurred while waiting to lock object** *name*

Cause: While trying to lock a library object, a time-out occurred.

Action: Retry the operation later.

ORA-04029

ORA-04029   **error ORA-*num* occurred when querying *name***

Cause: The table or view being queried might be missing. Look up the ORA-*num* indicated in the message.

Action: Fix the error.

04030-04039: System Memory

This section lists messages generated by an Oracle Server when the operating system cannot allocate sufficient memory during an Oracle session.

ORA-04030

ORA-04030   **out of process memory when trying to allocate *num* bytes *str*,*str***

Cause: Operating system process private memory has been exhausted.

Action: See the database administrator or operating system administrator to increase process memory quota. There may be a bug in the application that causes excessive allocations of process memory space.

ORA-04031

ORA-04031   **unable to allocate *num* bytes of shared memory *num*, *num, num***

Cause: More shared memory is needed than was allocated in the operating system process. SGA private memory has been exhausted

Action: Either use the DBMS_SHARED_POOL package to pin large packages, reduce your use of shared memory, or increase the amount of available shared memory by increasing the value set for the SHARED_POOL_SIZE initialization parameter.

04040-04069: Stored Procedures

This section lists some of the messages generated when stored procedure are accessed.

ORA-04041

ORA-04041   **package specification must be created first before creating package body**

Cause: An attempt was made to create a package body before creating its package specification.

Action: Create the package specification before trying to create the package body.

ORA-04042

ORA-04042   **procedure, function, package, or package body does not exist**

Cause: An attempt was made to access a procedure, function, package, or package body that does not exist.

Action: Ensure the name specified is correct.

ORA-04043

ORA-04043   **object *name* does not exist**

Cause: An object name was specified that was not recognized by the system. There are several possible causes:

- An invalid name for a table, view, sequence, procedure, function, package, or package body was entered. Since the system did not recognize the invalid name, it responded with the message that the named object did not exist.

- An attempt was made to rename an index or a cluster or some other object that cannot be renamed.

Action: Check the spelling of the renamed object and rerun the code. Valid names of tables, views, functions, and so forth can be listed by querying the data dictionary.

ORA-04044

ORA-04044   **procedure, function, or package is not allowed here**

Cause: An attempt was made to specify a procedure, function, or package in an inappropriate place in a statement.

Action: Refer to *Oracle7 Server SQL Reference* for the correct placement of procedures, functions, and packages in statements.

ORA-04045

ORA-04045 **errors during recompilation/revalidation of** *name.name*

Cause: Errors occurred during the implicit recompilation/revalidation of the object named in the message.

When Trusted Oracle7 is configured in OS MAC mode, you may get this error if you attempt to select from a view that needs to be recompiled. In particular, if the view is on a lower database and a higher database selects from this view, this error can occur.

Action: More descriptive messages follow this one. Check the causes mentioned in the messages that follow and take the appropriate actions.

For Trusted Oracle7 users, if the cause is that the view needs to be recompiled, ensure that it is recompiled before it is selected.

ORA-04046

ORA-04046	**results of compilation are too large to support**

Cause: An attempt to compile and store a large stored procedure resulted in compilation data that is too large for the system to support or store.

Action: Reduce the size of the store procedure by splitting it into smallerstored procedures.

ORA-04050

ORA-04050   **invalid or missing procedure, function, or package name**

Cause: The required procedure, function, or package name is invalid or missing.

Action: Specify a valid name.

ORA-04051

ORA-04051 **user *name* cannot use database link *name***

Cause: An attempt was made to access a non-existent database link, or a link not owned by the user logging in or PUBLIC, while attempting to access a remote object.

Action: Change the database link structure, so all indirect remote access requests are done from the same user originating the request or PUBLIC.

ORA-04052

ORA-04052 **error occurred when looking up remote object** *name*

Cause: An error occurred looking up a remote object.

Action: Fix the error. Check that the remote database system has run the script to create necessary views used for querying/looking up objects stored in the database. See the *Oracle7 Server Administrator's Guide*.

ORA-04053

ORA-04053   **error occurred when validating remote object** *name*

Cause: An error occurred trying to validate a remote object.

Action: Fix the error. Check that the remote database system has run the script to create necessary views used for querying/looking up objects stored in the database. See the *Oracle7 Server Administrator's Guide*.

ORA-04054

ORA-04054 **database link** *name* **does not exist**

Cause: During compilation of a PL/SQL block, an attempt was made to use a non-existent database link.

Action: Use a different database link or create the database link.

ORA-04055

ORA-04055 **package STANDARD@*name* is incompatible to local package STANDARD**

Cause: An attempt was made to compile a remote procedure call, but the remote procedure was compiled using a package STANDARD that has a different time-stamp from the one of the local package STANDARD. Therefore, those package STANDARDs are incompatible with each other and PL/SQLpackages, procedures, and functions compiled against the remote package STANDARD are also incompatible with those compiled against the local package STANDARD.

Action: Upgrade either the local database or the remote database, whichever is older, to the same version.

ORA-04060

ORA-04060   **insufficient privileges to execute *name***

Cause: An attempt was made to execute a stored procedure without sufficient privileges.

Action: Obtain the necessary privileges through direct grants of the privileges, not through roles.

ORA-04061

ORA-04061   **existing state of *name* has been invalidated**

Cause: An attempt was made to resume the execution of a stored procedure using the existing state, which has become invalid or inconsistent because the stored procedure has been altered or dropped.

Action: Attempt the action again. This action should cause the existing state of all packages to be re-initialized.

ORA-04062

ORA-04062 **time-stamp of** *name* **has been changed**

Cause: An attempt was made to execute a stored procedure to service a remote procedure call stub that specifies a time-stamp that is different from the current time-stamp of the procedure.

Action: Recompile the caller to obtain the new time-stamp.

ORA-04063

ORA-04063 *name* **has errors**

Cause: An attempt was made to execute a stored procedure that has errors. For stored procedures, the problem could be syntax or references to other, non-existent procedures. For views, the problem could be a reference in the view's defining query to a non-existent table.

Action: Fix the errors and create referenced objects as necessary.

ORA-04064

ORA-04064   not executed, invalidated *name*

Cause: An attempt was made to execute a stored procedure that has been invalidated.

Action: Recompile the procedure and its dependents.

ORA-04065

ORA-04065   **not executed, altered, or dropped** *name*

Cause: An attempt was made to execute a stored procedure that has been altered or dropped.

Action: Recompile the procedure's dependents.

ORA-04066

ORA-04066 **non-executable object,** *name*

Cause: An attempt was made to execute an object that is not a package, procedure, or function.

Action: Check that a correct name is used.

ORA-04067

ORA-04067 **not executed, *name* does not exist**

Cause: An attempt was made to execute a package, procedure, or function that does not exist.

Action: Check the name in the command.

ORA-04068

ORA-04068 **existing state of packages has been discarded**

Cause: An error occurred when attempting to execute a stored procedure.

Action: Refer to the accompanying messages and fix the error mentioned. Try the procedure again after proper re-initialization of any application's state.

04070-04099: Triggers

This section lists messages generated when triggers are accessed.

ORA-04070

ORA-04070   **invalid trigger name**

Cause: An invalid trigger name was specified.

Action: Check that the trigger name is not a reserved keyword.

ORA-04071

ORA-04071   missing BEFORE or AFTER keyword

Cause: The trigger statement is missing the BEFORE/AFTER clause.

Action: Specify either BEFORE or AFTER.

ORA-04072

ORA-04072   invalid trigger type

Cause: An invalid trigger type was given.

Action: Specify either INSERT, UPDATE, or DELETE as the trigger type.

ORA-04073

ORA-04073   **column list not valid for this trigger type**

Cause: A column list was specified for a trigger type other than UPDATE.

Action: Remove the column list from the trigger.

ORA-04074

ORA-04074   invalid REFERENCING name

Cause: An invalid name was given in the REFERENCING clause.

Action: Check that the REFERENCING name is not a reserved word.

ORA-04075

ORA-04075   **invalid trigger action**

Cause: A statement was given for the trigger action.

Action: Specify another trigger action that is not a statement.

ORA-04076

ORA-04076   **invalid NEW or OLD specification**

Cause: An invalid NEW or OLD specification was given for a column.

Action: Respecify the column using a correct NEW or OLD specification.

ORA-04077

ORA-04077   **WHEN clause cannot be used with table level triggers**

Cause: The WHEN clause can only be specified for row-level triggers.

Action: Remove the WHEN clause or specify it for each row.

ORA-04078

ORA-04078   **OLD and NEW values cannot be identical**

Cause: The REFERENCING clause specifies identical values for NEW and OLD.

Action: Specify the REFERENCING clause again with a different value for either NEW or OLD.

ORA-04079

ORA-04079   invalid trigger specification

Cause: The CREATE TRIGGER statement is invalid.

Action: Refer to *Oracle7 Server SQL Reference* for the correct syntax of the CREATE TRIGGER statement.

ORA-04080

ORA-04080   **trigger *name* does not exist**

Cause: The trigger name specified is invalid, or the trigger does not exist.

Action: Check the trigger name.

ORA-04081

ORA-04081   **trigger *name* already exists**

Cause: The trigger name or type already exists.

Action: Use a different trigger name or drop the trigger that is of the same type.

ORA-04082

ORA-04082 **NEW or OLD references not allowed in table level triggers**

Cause: The trigger is accessing NEW or OLD values in a table trigger.

Action: Remove any new or old references.

ORA-04083

ORA-04083   **invalid trigger variable** *name*

Cause: The variable referenced in the trigger body is invalid.

Action: See *Oracle7 Server SQL Reference* for valid trigger variable types.

ORA-04084

ORA-04084 **cannot change trigger NEW values in after row triggers**

Cause: New trigger variables can only be changed in before-row triggers.

Action: Change trigger type or remove the variable reference.

ORA-04085

ORA-04085   **cannot change the value of an OLD reference variable**

Cause: OLD values can only be read and not changed.

Action: Do not attempt to change an OLD variable.

ORA-04086

ORA-04086 **trigger description too long, move comments into triggering code**

Cause: The trigger description is limited to 2000 characters, for dictionary storage reasons. The description does not include the text of the "when" clause or the text of the PL/SQL code executed for the trigger.

Action: If the trigger description contains a large comment, move that comment into the PL/SQL code for the trigger.

ORA-04087

ORA-04087   **cannot change the value of ROWID reference variable**

Cause: ROWIDs can only be read and not changed.

Action: Do not attempt to change a ROWID value.

ORA-04088

ORA-04088   **error during execution of trigger *name.name***

Cause: A runtime error occurred during execution of a trigger.

Action: Check the triggers that were involved in the operation.

ORA-04089

ORA-04089   **cannot create triggers on objects owned by SYS**

Cause: An attempt was made to create a trigger on an object owned by SYS.

Action: Do not create triggers on objects owned by SYS.

ORA-04090

ORA-04090 *name* **specifies same table, event and trigger time as** *name*

Cause: The named trigger has a duplicate event and trigger time as another trigger.

Action: Combine both triggers into one trigger.

ORA-04091

ORA-04091   **table *name* is mutating, trigger/function may not see it**

Cause: A trigger or a user-defined PL/SQL function that is referenced in the statement attempted to query or modify a table that was in the middle of being modified by the statement that fired the trigger.

Action: Rewrite the trigger or function so it does not read the table.

ORA-04092

ORA-04092   **cannot COMMIT or ROLLBACK in a trigger**

Cause: A trigger attempted to COMMIT or ROLLBACK. This is not permitted.

Action: Rewrite the trigger so that COMMIT or ROLLBACK statements are not used.

ORA-04093

ORA-04093   **references to columns of type LONG are not allowed in triggers**

Cause: A trigger attempted to reference a LONG column in the triggering table.

Action: Do not reference the LONG column.

ORA-04094

ORA-04094   **table *name* is constraining, trigger may not modify it**

Cause: A trigger attempted to modify a table that was constraining for some referential constraint of a parent SQL statement.

Action: Rewrite the trigger so that it does not modify that table.

ORA-04095

ORA-04095   **trigger *name* already exists on another table, cannot replace**

Cause: An attempt was made to replace a trigger that exists on another table.

Action: Re-create the trigger on the other table using the CREATE OR REPLACE TRIGGER statement.

ORA-04096

ORA-04096 **trigger** *name* **has a WHEN clause that is larger than 2K**

Cause: A trigger's WHEN clause is limited to 2K for dictionary storage reasons.

Action: Use a smaller WHEN clause. Note, the trigger body could perform the same limiting action as the WHEN clause.

ORA-04097

ORA-04097 **error while trying to keep trigger n***ame*

Cause: An error occurred while trying to keep a trigger pinned in the shared pool.

Action: Check the trigger for possible syntax errors.

04100-04499: MVS

See your operating system-specific Oracle documentation. For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications" in the Preface.

04500-04899: CMS

See your operating system-specific Oracle documentation. For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications" in the Preface.

04900-04929: ICL DRS6000

See your operating system-specific Oracle documentation. For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications" in the Preface.

05000-05899: Original Equipment Manufacturers

See your operating system-specific Oracle documentation. For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications" in the Preface.

06000-06429: SQL*Net

See *Oracle Network Products Messages Manual* for information on these error messages. For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications" in the Preface.

06430-06499: NCR System 3000

See *Oracle7 Server for NCR System 3000 Installation and User's Guide.* For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications" in the Preface.

06500-06599: PL/SQL

This section lists messages generated when Oracle detects a problem with PL/SQL code. These ORA-*nnnnn* messages are usually followed by PLS-*nnnnn* messages, which are listed in Chapter 3.

ORA-06500

ORA-06500   PL/SQL: storage error

Cause: This is a rare internal error message. Memory has been exhausted or corrupted.

Action: Contact customer support.

ORA-06501

ORA-06501   PL/SQL: internal error *num*, arguments [*num*], [*num*], [*num*], [*num*],[*num*], [*num*]

Cause: This is an internal error message. An error has been detected in a PL/SQL program.

Action: Contact customer support.

ORA-06502

ORA-06502 **PL/SQL: numeric or value error**

Cause: An arithmetic, numeric, string, conversion, or constraint error occurred. For example, this error occurs if an attempt is made to assign the value NULL to a variable declared NOT NULL, or if an attempt is made to assign an integer larger than 99 to a variable declared NUMBER(2).

Action: Change the data, how it is manipulated, or how it is declared so that values do not violate constraints.

ORA-06503

ORA-06503 **PL/SQL: Function returned without value**

Cause: A call to a PL/SQL function completed, but no RETURN statement was executed.

Action: Rewrite the PL/SQL function, making sure that it always returns a value of a proper type.

ORA-06504

ORA-06504   **PL/SQL: Return types of Result Set variables or query do not match**

Cause: Number and/or types of columns in a query do not match the declared return type of a Result Set variable, or the declared types of two Result Set variables do not match.

Action: Change the program statement or declaration. Identify the query to which the variable, during execution, actually refers.

ORA-06508

ORA-06508   **PL/SQL: could not find program unit being called**

Cause: An attempt was made to call a stored program that could not be found. The program may have been dropped or incompatibly modified, or have compiled with errors.

Action: Check that all referenced programs, including their package bodies, exist and are compatible.

ORA-06509

ORA-06509   PL/SQL: ICD vector missing for this package

Cause: This indicates a version clash between some package distributed with an Oracle product and the product executable.

Action: Contact customer support.

ORA-06510

ORA-06510   **PL/SQL: unhandled user-defined exception *str***

Cause: A user-defined exception was raised by PL/SQL code, but not handled.

Action: Fix the problem causing the exception or write an exception handler for this condition. It may be necessary to contact the application ordatabase administrator.

ORA-06511

ORA-06511   PL/SQL: cursor already open

Cause: An attempt was made to open a cursor that was already open.

Action: Close the cursor before attempting to reopen it.

ORA-06512

ORA-06512 **at** *str* **line** *num*

Cause: This is usually the last of a message stack and indicates where a problem occurred in the PL/SQL code.

Action: Fix the problem causing the exception or write an exception handler for this condition. It may be necessary to contact the application or database administrator.

ORA-06513

ORA-06513 **PL/SQL: index for PL/SQL table out of range for host language array**

Cause: An attempt was made to copy a PL/SQL table to a host language array, but an index in the table is either less than one or greater than the maximum size of the host language array. When copying PL/SQL tables to host language arrays, the table entry at index 1 is placed in the first element of the array, the entry at index 2 is placed in the second element of the array, and so on. If a table entry was not assigned, then the corresponding element in the host language array is set to NULL.

Action: Increase the size of the host language arrays or decrease the size of the PL/SQL table. Also make sure that you do not use index values less than 1.

ORA-06550

ORA-06550   **line *num*, column *num*: *str***

Cause: A PL/SQL compilation error has occurred. The numbers given for line and column are the location in the PL/SQL block where the error occurred.

Action: Refer to the following PL/SQL messages for more information about the error.

ORA-06554

ORA-06554   package DBMS_STANDARD must be created before using PL/SQL

Cause: The data dictionary script required by PL/SQL has not yet been run.

Action: See the *Oracle7 Server Administrator's Guide* and your Oracle7 operating system-specific documentation for more information.

ORA-06555

ORA-06555 **this name is currently reserved for use by user SYS**

Cause: An attempt was made to create a user-owned package named STANDARD, DBMS_STANDARD, or DBMS_OUTPUT. This is not allowed.

Action: Choose another name for the package.

ORA-06556

ORA-06556   **the pipe is empty, cannot fulfill the UNPACK_MESSAGE request**

Cause: There are no more items in the pipe.

Action: Check that the sender and receiver agree on the number and types of items placed on the pipe.

ORA-06557

ORA-06557   null values are not allowed for any parameters to pipe ICD's

Cause: Internal error from the DBMS_PIPE package.

Action: Contact customer support.

ORA-06558

ORA-06558   buffer in DBMS_PIPE package is full. No more items allowed

Cause: The pipe buffer size has been exceeded.

ORA-06559

ORA-06559 **wrong datatype requested, *type*, actual datatype is *type***

Cause: The sender put a different datatype on the pipe than that being requested, package DBMS_PIPE. The recognized datatypes and the corresponding numbers are: 6 - number, 9 - char, 12 - date.

Action: Check that the sender and receiver agree on the number and types of items placed on the pipe.

ORA-06562

ORA-06562 **type of OUT argument must match type of column or bind variable**

Cause: An attempt was made to get the value of a column or a bind variable by calling procedure COLUMN_VALUE or VARIABLE_VALUE of package DBMS_SQL. However, the type of the given out argument was different from the type of the column or bind variable that was previously defined by calling procedure DEFINE_COLUMN, for defining a column, or BIND_VARIABLE, for binding a bind variable, of package DBMS_SQL.

Action: Pass in an OUT argument of the correct type when calling procedure COLUMN_VALUE or VARIABLE_VALUE. The right type is the type that was provided when defining the column or binding the bind variable.

ORA-06560

ORA-06560   pos, *num*, is negative or larger than the buffer size, *num*

Cause: Internal error from the DBMS_PIPE package.

Action: Contact customer support.

ORA-06563

ORA-06563 **top level procedure or function specified, cannot have subparts**

Cause: The name to be resolved was specified with three parts, a.b.c, but the a.b part resolves to a top level procedure or function that does not have nested procedures. This can also happen with a two-part name, a.b, where a is a synonym for a top level package or procedure.

Action: Specify a procedure or function within a package or a top level procedure or function.

ORA-06564

ORA-06564 **object *name* does not exist**

Cause:　　　The named object could not be found. Either it does not exist or you do not have permission to access it.

Action:　　　Create the object or get permission to access it.

ORA-06565

ORA-06565   **cannot execute *name* from within stored procedure**

Cause: The named procedure cannot be executed from within a stored procedure, function, or package. This function can only be used from PL/SQL anonymous blocks.

Action: Remove the procedure from the calling stored procedure.

ORA-06566

ORA-06566   **invalid number of rows specified**

Cause: An invalid number of rows was specified in a call to the procedures DEFINE_COLUMN in the package DBMS_SQL. For a given parsed statement in a given cursor, all columns must be defined to have the same number of rows, so all the calls to DEFINE_COLUMN must specify the same numberof rows.

Action: Specify a number that matches the number for previously defined columns.

ORA-06567

ORA-06567   **invalid number of values specified**

Cause: An invalid number of values to be bound was specified in a call to the procedure BIND_VARIABLE in the package DBMS_SQL. For a given parsed statement in a given cursor, the same number of values must be bound for all bind variables, so all the calls to BIND_VARIABLE must specify the same number of values.

Action: Make sure that the same number of values are given for each of the bind variables.

ORA-06568

ORA-06568	obsolete ICD procedure called

Cause: An obsolete ICD procedure was called by a PL/SQL program. The PL/SQL program was probably written for an earlier release of Oracle.

Action: Ensure that all PL/SQL packages are upgraded to the latest release of Oracle by following the upgrade instructions noted in the README document or by running the CATPROC.SQL script.

ORA-06570

ORA-06570 **shared pool object does not exist, cannot be pinned**

Cause: The specified shared pool shared cursor could not be found. Therefore, it cannot be pinned.

Action: Make sure that a correct shared cursor name is given. Names are a string of the form 'HHHHHHHH,SDDDDDDDDDD' where the Hs are an 8-digit hex number from the 'address' column of V$SQLAREA, and the Ds are a 1- to 10-digit decimal number with an optional leading sign from the'hash_value' column.

ORA-06571

ORA-06571   function *name* does not guarantee not to update database

Cause: There are two possible causes for this message:
- A SQL statement references a packaged PL/SQL function that does not contain a pragma that prevents the database from being updated.
- A SQL statement references a stand-alone, PL/SQL function that contains an instruction to update the database.

Action: If the referenced function is a packaged PL/SQL function: Recreate the PL/SQL function with the required pragma; be certain to include the 'Write No Database State' (WNDS) argument in the argument list of the pragma.

If the referenced function is a stand-alone PL/SQL function: Do not use the function.

ORA-06572

ORA-06572 **function** *name* **has out arguments**

Cause: A SQL statement references either a packaged or a stand-alone PL/SQL function that contains an OUT parameter in its argument list. PL/SQL functions referenced by SQL statements must not contain the OUT parameter.

Action: Recreate the PL/SQL function without the OUT parameter in the argument list.

ORA-06573

ORA-06573 **function *name* modifies package state, cannot be used here**

Cause: There are two possible causes for this message:

- A SQL statement references a packaged PL/SQL function that does not contain a pragma containing the 'Write no PackageState' (WNPS).
- A SQL statement references a stand-alone PL/SQL function that modifies a package state. A stand-alone PL/SQL function referenced by a SQL statement cannot modify a package state.

Action: If the function is a packaged PL/SQL function: recreate the function and include a pragma containing the 'Write no Package State' (WNPS).

If the function is a stand-alone PL/SQL function: delete the function from the SQL statement.

ORA-06574

ORA-06574   function *name* references package state, cannot execute remotely

Cause: There are two possible causes for this message:

▪   A remote packaged function or a remote-mapped, local packaged function that does not contain a pragma with the 'Write no Package State' (WNPS) and 'Read no Package State' (RNPS) arguments references a package state.

▪   A remote stand-alone function or a remote-mapped, local stand-alone function contains a reference to a package state that reads or writes a package variable.

Only local functions that are referenced in a SELECT list, VALUES clause of an INSERT statement, or SET clause of an UPDATE statement can modify a package state.

Action: If the function is a packaged function: Recreate the function and include a pragma containing the 'Write no Package State' (WNPS) and 'Read no Package State' (RNPS) arguments.

If the function is a stand-alone function: Do not call the function.

ORA-06575

ORA-06575 **function *name* is in an invalid state**

Cause: A SQL statement references a PL/SQL function that is in an invalid state. Oracle attempted to compile the function, but detected errors.

Action: Check the SQL statement and the PL/SQL function for syntax errors or incorrectly assigned, or missing, privileges for a referenced object.

06581-06591: European OEM Ports

See your operating system-specific Oracle documentation. For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications" in the Preface.

06600-06699: SQL*Net

See your operating system-specific Oracle documentation. For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications" in the Preface.

06700-06899: SQL*Net TLI

See your operating system-specific Oracle documentation. For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications" in the Preface.

06900-06939: CMX

See your operating system-specific Oracle documentation. For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications" in the Preface.

06950-06999: SQL*Net AppleTalk

See your operating system-specific Oracle documentation. For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications" in the Preface.

07000-07099: SQL*Connect for DB2

See your operating system-specific Oracle documentation. For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications" in the Preface.

07100-07199: SQL*Connect for IMS

See your operating system-specific Oracle documentation. For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications" in the Preface.

07200-07499: UNIX

See your operating system-specific Oracle documentation. For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications" in the Preface.

07500-07999: VAX/VMS

See your operating system-specific Oracle documentation. For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications" in the Preface.

08000-08174: Accessing Data

This section lists messages generated when Oracle is accessing data or database objects.

ORA-08000

ORA-08000   **maximum number of session sequence lists exceeded**

Cause: This is an internal error message not normally issued.

Action: Contact customer support.

ORA-08001

ORA-08001   **maximum number of sequences per session exceeded**

Cause: An operation requested a resource that was unavailable. The maximum number of user sequences per session is specified by the initialization parameter DC_SEQUENCES. When this maximum is reached, no more requests are processed.

Action: Try the operation again in a few minutes. If this error occurs often, shut down Oracle, increase the DC_SEQUENCES parameter in the initialization parameter file, and restart Oracle.

ORA-08002

ORA-08002 *name***.CURRVAL is not yet defined in this session**

Cause: Sequence CURRVAL was selected before sequence NEXTVAL was referenced.

Action: Select NEXTVAL from the sequence before selecting CURRVAL.

ORA-08003

ORA-08003 **sequence *name*.NEXTVAL exceeds internal limits**

Cause: The sequence was created with unsafe values for some of the parameters. The calculation of NEXTVAL cannot be made because it exceeds the legal representation size.

Action: Alter or recreate the sequence number with legal limits.

ORA-08004

ORA-08004   *name.*NEXTVAL *str* [MIN][MAX]VALUE and cannot be instantiated

Cause: The requested value of NEXTVAL is greater than MAXVALUE or less than MINVALUE.

Action: Alter the sequence so that a new value can be requested and try again.

ORA-08005

ORA-08005   **specified row does not exist**

Cause: A row with the given ROWID does not exist in any of the specified tables.

Action: Check the query for a misspelled ROWID or table name and try again.

ORA-08006

ORA-08006 **specified row no longer exists**

Cause: The row has been deleted by another user since the operation began.

Action: No action required.

ORA-08008

ORA-08008   another instance is mounted with USE_ROW_ENQUEUES = *num*

Cause: The shared instance being started does not have the same value for the USE_ROW_ENQUEUES initialization parameter as already-running instances.

Action: Make sure all instances' initialization parameter files specify the same value for the USE_ROW_ENQUEUES parameter and try again.

ORA-08100

ORA-08100   index is not valid - see trace file for diagnostics

Cause: There is an error in the index validated by VALIDATE INDEX.

Action: Check the relevant trace file. If the problem is not resolvable, contact customer support.

ORA-08101

ORA-08101 **index key does not exist root dba *num*, dba *num num***

Cause: The system detected a possible inconsistency in an index. This is an internal error message that is not normally issued.

Action: Send the trace file to customer support, along with a description of the circumstances that produced the error. Customer support will need the numbers given in the message.

ORA-08102

ORA-08102 **index key not found, obj#** *num***, dba** *num num*

Cause: The system detected a possible inconsistency in an index. This is an internal error message that is not normally issued.

Action: Send the trace file to customer support, along with a description of the circumstances that produced the error. Customer support will need the numbers given in the message.

ORA-08103

ORA-08103   **object no longer exists**

Cause: The object has been deleted by another user since the operation began.

Action: Remove references to the object.

08175-08190: Discrete Transactions

ORA-08175

ORA-08175   **discrete transaction restriction violated** *name*

Cause: An attempt was made to perform an action that is not currently supported in a discrete transaction.

Action: Rollback the transaction and retry it as a normal transaction.

ORA-08176

ORA-08176   **cannot continue consistent read for the table/index - no undo records**

Cause: Oracle encountered an operation that does not generate undo records. For example, the operation might have been an attempt to create an index by performing a direct load or executing a discrete mode transaction.

Action: Retry the operation with a different snapshot time. If using an index, retry the operation without using the index.

08200-08399: nCUBE

For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications" in the Preface.

08400-08499: PL/SQL Utility Packages for Procedural Gateway

ORA-08401

ORA-08401   invalid compiler name: *name*

Cause: An invalid compiler name was passed to a UTL_RAW conversion routine.

Action: Correct the compiler name parameter in the PL/SQL code that called the conversion routine.

ORA-08402

ORA-08402   invalid compiler name in MAKE_RAW_TO_NUMBER_FORMAT

Cause: An invalid compiler name was passed to the MAKE_RAW_TO_NUMBER_FORMAT routine.

Action: Correct the compiler name parameter in the PL/SQL code that calledthe routine.

ORA-08403

ORA-08403   invalid compiler name in MAKE_NUMBER_TO_RAW_FORMAT

Cause: An invalid compiler name was passed to the MAKE_NUMBER_TO_RAW_FORMAT routine.

Action: Correct the compiler name parameter in the PL/SQL code that called the routine.

ORA-08404

ORA-08404   **invalid compiler name in RAW_TO_NUMBER_FORMAT**

Cause: An invalid compiler name was passed to theRAW_TO_NUMBER_FORMAT routine.

Action: Correct the compiler name parameter in the PL/SQL code that called the routine.

ORA-08405

ORA-08405   **invalid compiler name in NUMBER_TO_RAW_FORMAT**

Cause: An invalid compiler name was passed to theNUMBER_TO_RAW_FORMAT routine.

Action: Correct the compiler name parameter in the PL/SQL code that called the routine.

ORA-08406

ORA-08406   **error encountered in MAKE_RAW_TO_NUMBER_FORMAT**

Cause: The MAKE_RAW_TO_NUMBER_FORMAT routine returned an error.

Action: This message will be preceded by messages providing details about the error. Check those messages to determine what action to take.

ORA-08407

ORA-08407   error encountered in MAKE_NUMBER_TO_RAW_FORMAT

Cause: The MAKE_NUMBER_TO_RAW_FORMAT routine returned an error.

Action: This message will be preceded by messages providing details about the error. Check those messages to determine what action to take.

ORA-08408

ORA-08408 error encountered in RAW_TO_NUMBER

Cause: The RAW_TO_NUMBER routine returned an error.

Action: This message will be preceded by messages providing details about the error. Check those messages to determine what action to take.

ORA-08409

ORA-08409 **error encountered in NUMBER_TO_RAW**

Cause: The NUMBER_TO_RAW routine returned an error.

Action: This message will be preceded by messages providing details about the error. Check those messages to determine what action to take.

ORA-08410

ORA-08410   **error encountered in RAW_TO_NUMBER_FORMAT**

Cause: The RAW_TO_NUMBER_FORMAT routine returned an error.

Action: This message will be preceded by messages providing details about the error. Check those messages to determine what action to take.

ORA-08411

ORA-08411   **error encountered in NUMBER_TO_RAW_FORMAT**

Cause: The NUMBER_TO_RAW_FORMAT routine returned an error.

Action: This message will be preceded by messages providing details about the error. Check those messages to determine what action to take.

ORA-08412

ORA-08412 **error encountered in WMSGBSIZ, size for WMSGBLK is not big enough for warning message**

Cause: The WMSGBSIZ is the maximum size for warning message block, it is recommended to be 1024 bytes to 8 kbytes.

Action: Defined WMSGBLK of size between 1k to 8k bytes and update the WMSGBSIZ to the size of WMSGBLK.

ORA-08413

ORA-08413 **invalid compiler type in FORMAT parameter at** *name*

Cause: An invalid compiler type was defined in the format control block. The format control block is invalid.

Action: Check to be sure that the format parameter was built by MAKE_RAW_TO_NUMBER_FORMAT or MAKE_NUMBER_TO_RAW_FORMAT and that it was not accidentally overwritten or modified by the PL/SQL procedure.

ORA-08414

ORA-08414 **error encountered in** *name*

Cause: The function <routine> returned an error. Where <routine> may be:RAW_TO_NUMBERNUMBER_TO_RAWRAW_TO_NUMBER_FORMATNUMBER_TO_RAW_FORMATMAKE_NUMBER_TO_RAW_FORMATMAKE_RAW_TO_NUMBER_FORMAT

Action: This message will be preceded by messages providing details about the error. Check those messages to determine what action to take.

ORA-08421

ORA-08421   **mask option USAGE IS POINTER is not supported**

Cause: The USAGE IS POINTER clause was included in the mask options passed to a UTL_RAW conversion routine.

Action: Remove the USAGE IS POINTER clause from the mask options parameter in the PL/SQL code that called the conversion routine. The POINTER data type is not allowed.

ORA-08422

ORA-08422   mask option USAGE IS INDEX is not supported

Cause: The USAGE IS INDEX clause was included in the mask options passed to a UTL_RAW conversion routine.

Action: Remove the USAGE IS INDEX clause from the mask options parameter inthe PL/SQL code that called the conversion routine. The INDEX data type is not allowed.

ORA-08423

ORA-08423   **mask option USAGE IS COMP-1 is not supported**

Cause: The USAGE IS COMP-1 clause was included in the mask options passed to a UTL_RAW conversion routine.

Action: Remove the USAGE IS COMP-1 clause from the mask options parameter inthe PL/SQL code that called the conversion routine. The COMP-1 data type is not allowed.

ORA-08424

ORA-08424   mask option USAGE IS COMP-2 is not supported

Cause: The USAGE IS COMP-2 clause was included in the mask options passed to a UTL_RAW conversion routine.

Action: Remove the USAGE IS COMP-2 clause from the mask options parameter inthe PL/SQL code that called the conversion routine. The COMP-2 data type is not allowed.

ORA-08425

ORA-08425   **raw buffer is too short for 8 bytes binary**

Cause: The input raw buffer passed to a UTL_RAW RAW_TO_NUMBER conversion routine was less than 8 bytes long, but the picture mask parameter specified that 8 bytes of input data were to be converted.

Action: Either the input data is incorrect, or the picture mask is incorrect. Correct the appropriate item.

ORA-08426

ORA-08426 **raw buffer is too short for 4 bytes binary**

Cause: The input raw buffer passed to a UTL_RAW RAW_TO_NUMBER conversion routine was less than 4 bytes long, but the picture mask parameter specified that 4 bytes of input data were to be converted.

Action: Either the input data is incorrect, or the picture mask is incorrect. Correct the appropriate item.

ORA-08427

ORA-08427   raw buffer is too short for 2 bytes binary

Cause: The input raw buffer passed to a UTL_RAW RAW_TO_NUMBER conversion routine was less than 2 bytes long, but the picture mask parameter specified that 2 bytes of input data were to be converted.

Action: Either the input data is incorrect, or the picture mask is incorrect. Correct the appropriate item.

ORA-08428

ORA-08428 raw buffer is too short for display data

Cause: The input raw buffer passed to a UTL_RAW RAW_TO_NUMBER conversion routine was too short to contain sufficient data for the specified picturemask parameter.

Action: Either the input data is incorrect, or the picture mask is incorrect. Correct the appropriate item.

ORA-08429

ORA-08429 **raw data has invalid digit in display type data**

Cause: The input raw buffer passed to a UTL_RAW RAW_TO_NUMBER conversion routine contained invalid data. The picture mask parameter specified a digit, but the corresponding input from the raw data did not contain a valid digit.

Action: Either the input data is incorrect, or the picture mask is incorrect. Correct the appropriate item.

ORA-08430

ORA-08430 raw data missing leading sign

Cause: The input raw buffer passed to a UTL_RAW RAW_TO_NUMBER conversion routine had no leading sign, but the mask options parameter specified a leading sign.

Action: Correct the input raw data or the mask options so that they match.

ORA-08431

ORA-08431 **raw data missing zero as defined in picture**

Cause: The picture mask parameter passed to a UTL_RAW RAW_TO_NUMBER conversion routine contained a zero, but the corresponding input from the raw data was not a zero.

Action: Either the input data is incorrect, or the picture mask is incorrect. Correct the appropriate item.

ORA-08432

ORA-08432   **raw data has invalid floating point data**

Cause: The input raw data passed to a UTL_RAW RAW_TO_NUMBER conversion routine contained invalid floating point data.

Action: Correct the input raw data.

ORA-08433

ORA-08433 **invalid picture type in convert raw to number**

Cause: The picture mask parameter passed to a UTL_RAW RAW_TO_NUMBER conversion routine contained non-numeric characters, but the conversion was to a numeric data type.

Action: Correct the picture mask parameter.

ORA-08434

ORA-08434 **raw data has invalid trailing sign**

Cause: The input raw buffer passed to a UTL_RAW RAW_TO_NUMBER conversion routine had no trailing sign, but the mask options parameter specified atrailing sign.

Action: Correct the input raw data or the mask options so that they match.

ORA-08435

ORA-08435   raw data missing the leading sign

Cause: The input raw buffer passed to a UTL_RAW RAW_TO_NUMBER conversion routine had no leading sign, but the mask options parameter specified aleading sign.

Action: Correct the input raw data or the mask options so that they match.

ORA-08436

ORA-08436   raw data has invalid sign digit

Cause: The input raw buffer passed to a UTL_RAW RAW_TO_NUMBER conversion routine had an invalid sign digit in the position where the picture mask specified a sign.

Action: Correct the input raw data or the picture mask so that they match.

ORA-08437

ORA-08437   **invalid picture type in picture mask**

Cause: The picture mask parameter passed to a UTL_RAW NUMBER_TO_RAW conversion routine contained non-numeric characters, but the conversion was to a numeric data type.

Action: Correct the picture mask parameter.

ORA-08438

ORA-08438   **error converting Oracle number to character data**

Cause: An error occurred when converting an Oracle number to a character variable. The Oracle number was not in the correct format.

Action: Correct the call to the conversion routine. The input must be a valid Oracle number variable.

ORA-08439

ORA-08439   **error converting Oracle number to COBOL COMP-3**

Cause: An error occurred when converting an Oracle number to a COBOL COMP-3 variable. The Oracle number was not in the correct format.

Action: Correct the call to the conversion routine. The input must be a valid Oracle number variable.

ORA-08440

ORA-08440 **raw buffer is too short to hold converted data**

Cause: The output raw buffer passed to a UTL_RAW NUMBER_TO_RAW conversion routine was not large enough to contain the results of the conversion based on the picture mask.

Action: Increase the raw buffer size to the size necessary to hold the entire result of the conversion.

ORA-08441

ORA-08441   **closed parenthesis missing in picture mask**

Cause: A closed parenthesis was missing from the picture mask passed to a UTL_RAW conversion routine.

Action: Correct the picture mask.

ORA-08442

ORA-08442 **error converting Oracle number to COBOL DISPLAY**

Cause: An error occurred when converting an Oracle number to a COBOL DISPLAY variable. The Oracle number was not in the correct format.

Action: Correct the call to the conversion routine. The input must be a valid Oracle number variable.

ORA-08443

ORA-08443 **syntax error in BLANK WHEN ZERO clause in mask options**

Cause: A syntax error was found in the BLANK WHEN ZERO clause in the mask options parameter passed to a UTL_RAW conversion routine. Valid specifications are: BLANK ZERO, BLANK ZEROS, BLANK ZEROES, BLANK WHEN ZERO, BLANK WHEN ZEROS, and BLANK WHEN ZEROES.

Action: Correct the mask options parameter.

ORA-08444

ORA-08444   **syntax error in JUSTIFIED clause in mask options**

Cause: A syntax error was found in the JUSTIFIED clause in the mask options parameter passed to a UTL_RAW conversion routine. Valid specifications are: JUST, JUST RIGHT, JUSTIFIED, and JUSTIFIED RIGHT.

Action: Correct the mask options parameter.

ORA-08445

ORA-08445   **syntax error in SIGN clause in mask options**

Cause: A syntax error was found in the SIGN clause in the mask options parameter passed to a UTL_RAW conversion routine. Valid specifications are: SIGN, LEADING SIGN, LEADING SEPARATE SIGN, LEADING SEPARATE CHARACTER SIGN, TRAILING SIGN, TRAILING SEPARATE SIGN, TRAILING SEPARATE CHARACTER SIGN IS LEADING, SIGN IS LEADING, SEPARATE SIGN IS LEADING, SEPARATE CHARACTER SIGN IS TRAILING, SIGN IS TRAILING, SEPARATE SIGN IS TRAILING, and SEPARATE CHARACTER.

Action: Correct the mask options parameter.

ORA-08446

ORA-08446 **syntax error in SYNCHRONIZED clause in mask options**

Cause: A syntax error was found in the SYNCHRONIZED clause in the mask options parameter passed to a UTL_RAW conversion routine. Valid specifications are: SYNC, SYNC LEFT, SYNC RIGHT, SYNCHRONIZED, SYNCHRONIZED LEFT, and SYNCHRONIZED RIGHT.

Action: Correct the mask options parameter.

ORA-08447

ORA-08447   **syntax error in USAGE clause in mask options**

Cause: A syntax error was found in the USAGE clause in the mask options parameter passed to a UTL_RAW conversion routine. Valid specifications are: USAGE, DISPLAY USAGE, COMP USAGE, COMP-3 USAGE, COMP-4 USAGE, COMPUTATIONAL USAGE, COMPUTATIONAL-3 USAGE, COMPUTATIONAL-4 USAGE IS, DISPLAY USAGE IS, COMP USAGE IS, COMP-3 USAGE IS, COMP-4 USAGE IS, COMPUTATIONAL USAGE IS, COMPUTATIONAL-3 USAGE IS, and COMPUTATIONAL-4.

Action: Correct the mask options parameter.

ORA-08448

ORA-08448 **syntax error in DECIMAL-POINT environment clause**

Cause: A syntax error was found in the DECIMAL-POINT environment clause parameter passed to a UTL_RAW conversion routine. Valid specifications are: DECIMAL-POINT IS COMMA.

Action: Correct the environment clause parameter.

ORA-08449

ORA-08449   invalid numeric symbol found in picture mask

Cause: An invalid numeric symbol was found in the picture mask parameter passed to a UTL_RAW conversion routine.

Action: Correct the picture mask parameter.

ORA-08450

ORA-08450   **invalid specification of CR in picture mask**

Cause: The CR suffix was incorrectly specified in the picture mask parameter passed to a UTL_RAW conversion routine. The CR suffix can only appear at the end of a picture mask.

Action: Correct the picture mask parameter.

ORA-08451

ORA-08451   **invalid specification of DB in picture mask**

Cause: The DB suffix was incorrectly specified in the picture mask parameter passed to a UTL_RAW conversion routine. The DB suffix can only appear at the end of a picture mask.

Action: Correct the picture mask parameter.

ORA-08452

ORA-08452   **specification of E in picture mask is unsupported**

Cause: The floating point exponent symbol "E" was specified in the picture mask parameter passed to a UTL_RAW conversion routine. The floating point data type is currently not supported by the UTL_RAW conversion routines.

Action: Correct the picture mask parameter, and the data, if necessary.

ORA-08453

ORA-08453   **more than one V symbol specified in picture mask**

Cause: The picture mask passed to a UTL_RAW conversion routine contained more than one decimal point indicator "V". Only one decimal point indicator is allowed in the picture mask.

Action: Correct the picture mask parameter.

ORA-08454

ORA-08454   **more than one S symbol specified in picture mask**

Cause: The picture mask passed to a UTL_RAW conversion routine contained more than one operational sign indicator "S". Only one operational sign indicator is allowed in the picture mask.

Action: Correct the picture mask parameter.

ORA-08455

ORA-08455 **syntax error in CURRENCY SIGN environment clause**

Cause: A syntax error was found in the CURRENCY SIGN environment clause parameter passed to a UTL_RAW conversion routine. Valid specifications are: CURRENCY SIGN IS x, where x is a valid currency sign.

Action: Correct the environment clause parameter.

ORA-08456

ORA-08456   **no sign in picture mask but SIGN clause in mask options**

Cause: The picture mask parameter passed to a UTL_RAW conversion routine contained no sign symbol, "S", "+", or "-", but the mask options parameter contained a SIGN clause. A sign symbol is required in the picture mask parameter when the mask options parameter contains a SIGN clause.

Action: Correct the picture mask parameter or the mask options parameter.

ORA-08457

ORA-08457   **syntax error in SEPARATE CHARACTER option of SIGN clause**

Cause: A syntax error was found in the SEPARATE CHARACTER option of the SIGN clause in the mask options parameter passed to a UTL_RAW conversion routine. Valid specifications are: SEPARATE and SEPARATE CHARACTER.

Action: Correct the mask options parameter.

ORA-08458

ORA-08458   invalid format parameter

Cause: The format parameter passed to a UTL_RAW conversion routine was invalid. The format parameter should have been built by a prior call to either MAKE_RAW_TO_NUMBER_FORMAT or MAKE_NUMBER_TO_RAW_FORMAT.

Action: Check to be sure that the format parameter was built by MAKE_RAW_TO_NUMBER_FORMAT or MAKE_NUMBER_TO_RAW_FORMAT and that it was not accidentally overwritten or modified by the PL/SQL procedure.

ORA-08459

ORA-08459 **invalid format parameter length**

Cause: The format parameter passed to a UTL_RAW conversion routine was not the correct length. Format parameters must be 2048 bytes in length.

Action: Check to be sure that the format parameter was built by MAKE_RAW_TO_NUMBER_FORMAT or MAKE_NUMBER_TO_RAW_FORMAT and that it was not accidentally overwritten or modified by the PL/SQL procedure.

ORA-08460

ORA-08460   **invalid environment clause in environment parameter**

Cause: The environment parameter passed to a UTL_RAW conversion routine contained an unsupported or invalid environment clause. Only the CURRENCY SIGN and the DECIMAL-POINT IS COMMA environment clauses are supported.

Action: Correct the environment parameter.

ORA-08461

ORA-08461   **raw buffer is too short for decimal data**

Cause: The input raw buffer passed to a UTL_RAW RAW_TO_NUMBER conversion routine was too short to contain sufficient data to satisfy the specified picture mask parameter.

Action: Either the input data is incorrect, or the picture mask is incorrect. Correct the appropriate item.

ORA-08462

ORA-08462   raw buffer contains invalid decimal data

Cause: The input raw buffer passed to a UTL_RAW RAW_TO_NUMBER conversion routine contains invalid decimal data.

Action: Correct the input data.

ORA-08463

ORA-08463   **overflow converting decimal number to Oracle number**

Cause: The output variable passed to a UTL_RAW RAW_TO_NUMBER was not large enough to hold the Oracle number resulting from the input decimal number.

Action: Be sure that the input decimal number is valid, and be sure that the output variable is large enough to hold the Oracle number value.

ORA-08464

ORA-08464   **input raw decimal data contains more than 42 digits**

Cause: The input raw buffer passed to a UTL_RAW RAW_TO_NUMBER conversion routine contained more than 42 digits. This exceeds the maximum size of an Oracle number.

Action: Correct the raw input buffer.

ORA-08465

ORA-08465   **input mask contains more than 32 characters**

Cause: The input mask passed to UTL_PG numeric conversionroutine contained more the 32 characters.

Action: Correct the mask input buffer.

ORA-08466

ORA-08466 **raw buffer length *num* is too short for *name***

Cause: The input raw buffer passed to a UTL_PG   RAW_TO_NUMBERconversion routine was less than *num* bytes long, but thepicture mask parameter specified that *num* bytes of input data were to be converted.

Action: Either the input data is incorrect, or the picture mask is incorrect. Correct the appropriate item.

ORA-08467

ORA-08467   **error converting Oracle number to** *num*

Cause: An error occurred when converting an Oracle number to aCOBOL <type> of:DISPLAYCOMP-3or character variable.The Oracle number was not in the correct format.

Action: Correct the call to the conversion routine. The input must be a valid Oracle number variable.

ORA-08468

ORA-08468   **mask option** *name* **is not supported**

Cause: The mask option <clause> was passed to a UTL_PGconversion routine, but is not supported by UTL_PG.The <clause>   can be:USAGE IS POINTERUSAGE IS INDEXUSAGE IS COMP-1USAGE IS COMP-2POINTER

Action: Remove the <clause> from the mask options parameter in the PL/SQL call to UTL_PG.

ORA-08498

ORA-08498 **Warning: picture mask *name* overrides picture mask option USAGE IS *name* to USAGE IS DISPLAY**

Cause: Picture mask option is overridden by Picture mask.

Action: No action need, just for information.

ORA-08499

ORA-08499   **Warning: picture mask options *name* ignored by UTL_RAW**

Cause: Picture mask options such as OCCUR, SYNC and others are not processedby the UTL_RAW numeric conversion routines and will put in above warning message.

Action: No action need, just for information.

08500-09099: SQL*Connect

For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications" in the Preface.

09100-09199: Oracle Gateways

This section lists messages that are returned to an Oracle7 client application because a problem occurred while accessing a non-Oracle system, known as a target system, using an Oracle gateway.

ORA-09100

ORA-09100   **target system returned following message:**

Cause: A target system being accessed by an Oracle gateway returned an error that could not be handled by the gateway server. This message is followed by a second message generated by the target system or the gateway.

Action: See the documentation for the target system for an explanation of the second message.

ORA-09101

ORA-09101   **target system communication error**

Cause: The gateway encountered an error when communicating with the target system. This message will generally be followed by a second message generated either by the target system or the gateway.

Action: See the documentation for the target system for an explanation of the second message.

ORA-09102

ORA-09102 **gateway internal communication error**

Cause: The Oracle gateway encountered an internal error during communication with a module that is part of the gateway.

Action: Contact customer support.

ORA-09103

ORA-09103 **gateway internal protocol error**

Cause: The Oracle gateway encountered an internal error during communication with an internal module.

Action: Contact customer support.

ORA-09104

ORA-09104   **gateway internal cross-communication error**

Cause: The Oracle gateway encountered an internal error during communication with a module that is part of the gateway but located in a different process.

Action: Contact customer support.

ORA-09105

ORA-09105   failed to acquire storage space in target system

Cause: For an Oracle gateway, the amount of storage available within the target system was reached. This message is followed by a second message generated by the target system.

Action: See the documentation for the target system for an explanation of the second message.

ORA-09106

ORA-09106 **target system could not obtain IO resource**

Cause: The target system was unable to open a file, dataset, or other resource needed to perform the I/O. This message is followed by a second message generated by the target system.

Action: See the documentation for the target system for an explanation of the second message.

ORA-09107

ORA-09107   **definition of a target system object may have changed**

Cause: In an Oracle transparent gateway, an attempt was made to execute a SQL statement whose definition in the GDDL source may be out of date. The timestamp in the gateway data dictionary is different from the timestamp in the target system.

Action: Use the Transparent Gateway Administrator utility to replace the table using the revised GDDL source.

ORA-09108

ORA-09108 **invalid record in target system**

Cause: The Oracle gateway or the target system noticed that the record is not valid according to its definition. This message is followed by a second message generated by the gateway or the target system depending on the error.

Action: See the gateway documentation or the documentation for the target system for an explanation of the second error message.

ORA-09109

ORA-09109   view "*view_name*" cannot be used in a subquery

Cause: The view definition contains a set operator, which makes it impossible to use that view definition in a subquery. This message can occur only in SQL-based gateways.

Action: Do not use the view in a subquery.

ORA-09110

ORA-09110   **view "*view_name*" defines operation not supported on the target system**

Cause: In a user-developed gateway, the implementor defined a view that defines an Oracle table in terms of other data source tables, and this definition contains SQL elements not supported by that system.

Action: Contact the implementor to correct the view definition error.

ORA-09111

ORA-09111   **data manipulation operation not legal on this object**

Cause: In an Oracle transparent gateway, the target system rejected an insert, delete, or update operation. This message is followed by a second message generated by the target system.

Action: See the documentation for the target system for an explanation of the second message.

ORA-09112

ORA-09112   **target system check constraint violated**

Cause: The values being inserted or updated do not satisfy the check constraint in the transparent gateway. This message is followed by a second message generated by the target system or the gateway.

Action: See the documentation for the target system or your Oracle7 operating system-specific documentation for an explanation of the second message.

ORA-09113

ORA-09113 **target system integrity constraint violated**

Cause: The values being inserted, updated, or deleted do not satisfy the integrity constraints defined in the target system in a transparent gateway. This message is followed by a second message generated by the target system or the gateway.

Action: See the documentation for the target system or your Oracle7 operating system-specific documentation for an explanation of the second message.

ORA-09114

ORA-09114　**target system is out of memory**

Cause:　A target system being accessed by an Oracle gateway ran out of memory. This message is followed by a second message generated by the target system.

Action:　See the documentation for the target system for an explanation of the second message.

ORA-09115

ORA-09115 **target system has timed out**

Cause: A target system being accessed by an Oracle gateway timed out. This error occurs when system resource usage is too high for the request to be served in a reasonable time or when a deadlock, or some other wait condition, occurred in the target system. This message is followed by a second message generated by the target system or the gateway.

Action: See the documentation for the target system or your Oracle7 operating system-specific documentation for an explanation of the second message.

ORA-09116

ORA-09116   **definition of target system data object is inconsistent**

Cause: In an Oracle transparent gateway, The target system was unable to perform the initial operations necessary to access the mapped table. The GDDL source specifies certain fields or options that are not supported for this gateway configuration.

Action: Correct the GDDL source and issue the Transparent Gateway Administrator command to CREATE OR REPLACE or DROP the table.

ORA-09117

ORA-09117   **bind-value cannot be translated into SQL text**

Cause: For the SQL-based transparent gateway, the target system does not support binds, or binds are disabled using the RESOLVE_BINDS gateway initialization parameter.

Action: Enable binds or do not bind using datatypes not supported in SQL text.

ORA-09118

ORA-09118   maximum number of savepoints exceeded

Cause: For the SQL-based transparent gateway, the target system was unable to set the savepoint because the maximum number of active savepoints was reached.

Action: Re-organize the transaction so that this limit is not reached or increase the maximum in the target system if that is possible.

ORA-09119

ORA-09119   **initialization file contains error**

Cause: An Oracle gateway server encountered an error in the gateway server initialization file. This message is followed by a second message. If the second message did not follow, it can be found in the gateway log file.

Action: Refer to the second message to fix the initialization file.

ORA-09120

ORA-09120   **gateway has timed out access to target system**

Cause: The Oracle Gateway request was not serviced within the specified time by the target system.

Action: See your Oracle7 operating system-specific documentation for an explanation of when this error can occur.

ORA-09121

ORA-09121   **fetch exceeds maximum number of records defined for table**

Cause: The transparent gateway detected that a query will retrieve more rows than the maximum defined in the GDDL for mapped table.

Action: See your Oracle7 operating system-specific documentation for an explanation of when this error can occur.

ORA-09122

ORA-09122 **access to object "*name*" denied**

Cause: An Oracle transparent gateway denied access to the specified object.

Action: Log in as a user with access to the object or change the access permission of the object.

ORA-09123

ORA-09123   **ROWID not found in gateway rowid cache**

Cause: For the transparent gateway, the ROWID cache holds no entry that corresponds to the specified ROWID. Possibly, the ROWID entry was overwritten in the gateway ROWID cache.

Cause: Increase the size of the gateway ROWID cache by setting the value of the ROWID_CACHE_SIZE gateway initialization parameter to a higher value and restart the gateway server.

ORA-09124

ORA-09124 **in** *name* **argument** *name***: Index out of range**

Cause: The transparent gateway server detected that the index for a mapped table is bigger than allowed by the definition in the GDDL source.

Action: Modify the query or the GDDL source so that the index is in range.

09200-09499: DOS, OS/2, and Novell

See your operating system-specific Oracle documentation. For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications"related_pubs in the Preface.

09500-09699: MPE/XL

See your operating system-specific Oracle documentation. For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications"related_pubs in the Preface.

09700-09999: UNIX

See your operating system-specific Oracle documentation. For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications"related_pubs in the Preface.

10000-10999: Internal

11000-11999: SQL*Net MVS

See your operating system-specific Oracle documentation. For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications"related_pubs in the Preface.

12000-12014: Table Snapshots

This section lists messages generated when table snapshots are accessed.

ORA-12000

ORA-12000   **a snapshot log already exists on table** *name*

Cause: An attempt was made to create a snapshot log on a table that already has a snapshot log. Each master table can have only one snapshot log.

Action: All snapshots on a table can use the same snapshot log for that table. To make changes to the existing log, use the ALTER SNAPSHOT LOG command or drop the snapshot log and create it again.

ORA-12001

ORA-12001   **cannot create log: table *name* already has a trigger**

Cause: An attempt was made to create a snapshot log on a table when the table already had an AFTER ROW trigger. Snapshot logs are filled by an AFTER ROW trigger on the master table, so the trigger cannot be created.

Action: If it is necessary to create a snapshot log, drop the current trigger on the master table.

ORA-12002

ORA-12002	**there is no snapshot log on table *name***

Cause: An attempt was made to drop a snapshot that does not exist.

Action: Check spelling, syntax, and use an existing snapshot name.

ORA-12003

ORA-12003   **snapshot *name* does not exist**

Cause: An attempt was made to reference a snapshot that does not exist.

Action: Check the name of the snapshot in the statement for typing errors. Retry the statement with an existing snapshot name.

ORA-12004

ORA-12004 **REFRESH FAST cannot be used**

Cause: An attempt to fast refresh a snapshot was made when the master table for the snapshot has no snapshot log, or the snapshot log cannot be used.

Action: To REFRESH FAST, first create a snapshot log on the master table. Otherwise, use the REFRESH COMPLETE option of the ALTER SNAPSHOT command.If these options are omitted, Oracle decides the appropriate method for refreshing.

ORA-12005

ORA-12005   **may not schedule automatic refresh for times in the past**

Cause: An attempt was made to schedule an automated snapshot refresh for a time in the past.

Action: Choose a time in the future for the automated snapshot refresh.

ORA-12006

ORA-12006 **a snapshot with the same *user.name* already exists**

Cause: An attempt was made to create a snapshot with the name of an existing snapshot.

Action: Create the snapshot using a different name or drop the existing snapshot.

ORA-12007

ORA-12007   **snapshot reuse parameters are inconsistent**

Cause: During Import or Export, the CREATE SNAPSHOT or CREATE SNAPSHOT LOG.. REUSE command was given inconsistent parameters immediately after the REUSE.

Action: Other messages will accompany this message. Examine the other messages for a more complete description of the problem.

ORA-12008

ORA-12008   error in snapshot refresh path

Cause: Some problem is preventing the snapshot from reading the view on the master table. A number of events could cause this problem, among them the following:

- The master table no longer exists.
- A column was added to the master table, and the view was not altered.
- The master is at a remote site and inaccessible.

Action: Other messages will accompany this message. Examine the other messages for a more complete description of the problem. The most likely solution to the problem will be to recreate the snapshot.

ORA-12009

ORA-12009   **snapshots may not contain long columns**

Cause: An attempt was made to create a snapshot with a long column.

Action: Do not attempt to create snapshot logs with long columns.

ORA-12010

ORA-12010   **cannot create snapshot log on table owned by SYS**

Cause: CREATE SNAPSHOT LOG attempts to create a trigger on the table, but triggers cannot be created on SYS tables.

Action: Do not create snapshot logs on SYS tables.

ORA-12011

ORA-12011   **execution of *num* jobs failed**

Cause: Some kind of error was caught in DBMS_IJOB.RUN. One or more jobs that were due to be run produced errors that they could not handle.

Action: Look at the alert log for details on which jobs failed and why.

ORA-12012

ORA-12012 **error on auto execute of job *num***

Cause: Some kind of error was caught while doing an automatic execute of a job.

Action: Look at the accompanying errors for details on why the execute failed.

ORA-12013

ORA-12013 **updatable snapshots must be simple enough to do fast refresh**

Cause: The updatable snapshot query statement is specified with a join, subquery, union, CONNECT BY, ORDER BY, or GROUP BY.

Cause: Specify a simpler snapshot statement. If a join is needed, create multiple simple snapshots, and then create a view on top of the multiple snapshots.

12100-12299: SQL*Net

See your operating system-specific Oracle documentation. For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications" in the Preface.

12300-12499: Trusted Oracle7

This section lists messages generated by Trusted Oracle7.

ORA-12315

ORA-12315   **database link type is invalid for the ALTER DATABASE statement**

Cause: The database link name specified on the ALTER DATABASE statement is not an ROM: link. Specify an ROM: link when using the ALTER DATABASE statement to mount or open a secondary database.

Action: Retry the ALTER DATABASE statement using a valid ROM: link to the database to be mounted or opened. If a valid ROM: link does not exist, create one using the CREATE DATABASE LINK command. See the *Trusted Oracle7 Server Administrator's Guide, Release 7.1* for more information about creating database links using the ROM: link type.

ORA-12316

ORA-12316 syntax error in database link's connect string

Cause: The connect string in the CREATE DATABASE LINK statement has a syntactical error.

Action: Drop the database link and recreate it using valid syntax. See *Oracle7 Server SQL Reference* for more information about the connect string portion of the CREATE DATABASE LINK statement.

ORA-12317

ORA-12317   logon to database (link name *name*) denied

Cause: Any of the following:
- Either the user name or password was misspelled.
- The user does not exist in the secondary database.
- The user in the secondary database was not granted the CREATE SESSION system privilege.
- The username/password in the database link definition is invalid for any of the above reasons.
- If database verification of login information is active, then the username and password do not match in the primary and secondary databases.
- If operating system verification of login information is active, then the username does not match in the primary and secondary databases.

Action: Enter the correct information.
- Ask the database administrator for the secondary database to create the user.
- Ask the database administrator for the secondary database to grant the CREATE SESSION system privilege to the user.
- Correct the database link definition.
- Check that username and password match in the primary and secondary databases.
- Check that the username matches in the primary and secondary databases.

Note: Always use operating system authentication in Trusted Oracle7 For more information about the advantages of O/S authentication, see the *Trusted Oracle7 Server Administrator's Guide, Release 7.1.*

ORA-12318

ORA-12318 **database (link name *name*) is already mounted**

Cause: An attempt was made to mount a secondary database already mounted by the instance.

Action: No additional action is required to mount the database. To establish access, use the ALTER DATABASE command to open the database.

ORA-12319

ORA-12319 **database (link name *name*) is already open**

Cause: An attempt was made to open a secondary database that is already open.

Action: No additional action is required to establish access.

ORA-12321

ORA-12321   **database (link name *name*) is not open and AUTO_MOUNTING=FALSE**

Cause: The instance attempted to mount the secondary database, but it is not open and it is not enabled for automatic mounting.

Action: Manually mount and open the secondary database using the ALTER DATABASE command with the OPEN and MOUNT options. To allow the instance to mount and open secondary databases automatically, set the AUTO_MOUNTING parameter in the initialization parameter file to TRUE.

ORA-12322

ORA-12322 **unable to mount database (link name *name*)**

Cause: This message should be accompanied by additional messages that indicate the cause of the problem.

Action: Follow the steps outlined in the accompanying messages to resolve the problem.

ORA-12323

ORA-12323   **unable to open database (link name *name*)**

Cause: This message should be accompanied by additional messages that indicate the cause of the problem.

Action: Follow the steps outlined in the accompanying messages to resolve the problem.

ORA-12324

ORA-12324   **cannot use the ROM: link type on a private database link**

Cause: You can only specify the ROM: link type on a public, not on a private, database link.

Action: If there is an existing public database link to the secondary database, use that link. If not, and public access to the secondary database is desired, create a public database link to the secondary database using the CREATE DATABASE LINK command.

ORA-12326

ORA-12326   **database *name* is closing immediately; no operations are permitted**

Cause: The operation attempted was terminated because the database is closing.

Action: Wait until the database reopens or contact the database administrator.

ORA-12329

ORA-12329   **database *name* is closed; no operations are permitted**

Cause: The operation attempted was terminated because the database is closing.

Action: Wait until the database reopens or contact the database administrator.

ORA-12333

ORA-12333 **database (link name *name*) is not mounted**

Cause: An attempt was made to open a database that is not mounted.

Action: Mount the database with the ALTER DATABASE command, then reattempt to open the database.

ORA-12334

ORA-12334 **database (link name *name*) is still open**

Cause: An attempt was made to dismount a database that is still open.

Action: Close the database with the ALTER DATABASE command, then reattempt to dismount the database.

ORA-12335

ORA-12335   **database (link name *name*) is not open**

Cause: An attempt was made to close a database that is not open.

Action: The database is closed; proceed with dismounting it.

ORA-12341

ORA-12341   **maximum number of open mounts exceeded**

Cause: The number specified on the OPEN_MOUNTS parameter in the initialization parameter file exceeds 255, which is the maximum allowed.

Action: Change the value of this initialization parameter so that it reflects the actual number of possible open mounts to secondary databases. This must be less than 255.

ORA-12342

ORA-12342   open mounts exceeds limit set on the OPEN_MOUNTS parameter

Cause: The number of currently open mounts exceeds the value specified in the OPEN_MOUNTS initialization parameter.

Action: Increase the value of the OPEN_MOUNTS parameter so that it accommodates the maximum possible number of open mounts to secondary databases.

ORA-12345

ORA-12345   user *name* lacks CREATE SESSION privilege in database (link name %s)

Cause:　　　There are two causes of this message:
- Your username in the second database specified was not granted the CREATE SESSION system privilege.
- The username specified in the connect string of the database link definition was not granted the CREATE SESSION system privilege.

Action:　　　The action you take depends upon the cause of the message:
- In the first case, check that your username in the secondary database was granted the CREATE SESSION system privilege.
- In the second case, check that the username specified in the connect string of the database link definition was granted the CREATE SESSION system privilege in the secondary database.

ORA-12350

ORA-12350   **database link being dropped is still mounted**

Cause: An attempt was made to drop a ROM: database link while it was still mounted and/or opened.

Action: Close and dismount the database. Then retry the drop statement.

ORA-12351

ORA-12351 **cannot create view using a remote object which has a remoteobject reference**

Cause: An attempt was made to create a view that references a remote object that, in turn, references an object on another database. Because the view that you tried to create references a remote object, that object cannot reference an object on another database.

Action: Choose a different object to reference in your view or change the remote object so that it does not reference another database.

ORA-12352

ORA-12352   **object *name* is invalid**

Cause: An attempt was made to reference, or compile against, an object in a secondary database. However, the object is invalid because it is in a secondary database. Therefore, the system cannot validate or recompile it.

Action: Recompile the invalid object in the secondary database.

ORA-12353

ORA-12353   **secondary stored object cannot reference remote object**

Cause: You tried either to select from a remote view or execute a remote procedure that references an object on another database. Because the remote view or procedure is on a secondary database, an additional reference to another database cannot be done.

Action: Choose a different object to reference or change the remote view or procedure so that it does not reference another database.

ORA-12354

ORA-12354   **secondary object being dropped**

Cause: You tried to access an object, for example, a table or view, on a secondary database that was being dropped.

Action: Repeat the operation. If you receive the message again, try to access the object from the secondary database. If you receive an internal error or a trace file, contact customer support.

ORA-12400

ORA-12400   **DBLOW is not a valid operating system label**

Cause: The operating system label corresponding to DBLOW has become invalid due to a change in operating system label definitions.

Action: Use the operating system facilities to restore the definition of this label, or change the value of DBLOW to a valid label. To change the value of DBLOW, you must mount the database and switch to OSMAC mode.

ORA-12401

ORA-12401 **DBHIGH is not a valid operating system label**

Cause: The operating system label corresponding to DBHIGH has become invalid due to a change in operating system label definitions.

Action: Use the operating system facilities to restore the definition of this label, or change the value of DBHIGH to a valid label. To change the value of DBHIGH, you must mount the database and switch to OSMAC mode.

ORA-12402

ORA-12402 **file:** *str*

Cause: Reporting filename for detail of another message.

Action: Correct the underlying problems mentioned in the other messages.

ORA-12403

ORA-12403   **file label *name* must be equal to DBHIGH *name***

Cause: A datafile was found to have a label not equal to DBHIGH. This may be caused if DBHIGH is changed or if a file is relabeled.

Action: Either relabel the file to match DBHIGH or change DBHIGH to match the file.

ORA-12404

ORA-12404 **device low** *name* **must be dominated by DBLOW** *name*

Cause: A datafile was found on a device with an authorized range insufficient to handle the lowest labeled data in the file.

Action: Lower the authorized minimum device label or move the file to a different device.

ORA-12405

ORA-12405   **device high *name* must dominate DBHIGH *name***

Cause: A datafile was found on a device with an authorized range insufficient to handle the highest labeled data in the file.

Action: Raise the authorized maximum device label or move the file to a different device.

ORA-12406

ORA-12406   **must be at DBHIGH *name* in O/S to add files to a tablespace**

Cause: A file was added to a new or existing tablespace by a user whose operating system label was not equal to DBHIGH.

Action: Log in to the O/S at DBHIGH to add the file. If creating a new tablespace, set the session label to the label at the desired tablespace definition. If adding a file to an existing tablespace, either set the session label to match the tablespace or user WRITEDOWN privilege to alter the tablespace.

ORA-12407

ORA-12407 **cannot insert or update the ROWLABEL column of a view**

Cause: You tried to insert into or update the ROWLABEL column of a view. Trusted Oracle7 does not allow you to modify the ROWLABEL column of a view.

Action: While you cannot modify the ROWLABEL column of a view, you can modify the ROWLABEL column in the base table(s) of the view. Use the INSERT or UPDATE commands to modify the base tables(s). Consequently, this modifies the view.

ORA-12408

ORA-12408 **tablespace previously existed at another label** *name*

Cause: A tablespace by this name was previously created and dropped at another label. Part of the dictionary entry for the old tablespace remains, and the new tablespace is attempting to reuse the old entry.

Action: Either recreate the tablespace at the original label or choose a new name for this tablespace.

ORA-12409

ORA-12409 **DBLOW cannot be null in DBMS MAC mode**

Cause: The database was started in DBMS MAC mode before resetting DBLOW.

Action: Restart the database in OS MAC mode and supply valid values for both DBHIGH and DBLOW before starting up in DBMS MAC mode.

ORA-12410

ORA-12410   **DBHIGH cannot be null in DBMS MAC mode**

Cause: The database was started in DBMS MAC mode before resetting DBHIGH.

Action: Restart the database in OS MAC mode and supply valid values for both DBHIGH and DBLOW before starting up in DBMS MAC mode.

ORA-12411

ORA-12411   **DBHIGH must dominate DBLOW**

Cause: DBHIGH does not dominate DBLOW.

Action: Restart the database in OS MAC mode and supply valid values for DBHIGH and DBLOW before starting up in DBMS MAC mode.

ORA-12412

ORA-12412 **cannot raise DBLOW above** *num*

Cause: An attempt was made to raise DBLOW above its current value. DBLOW can only be lowered, it cannot be raised.

Action: Do not attempt to raise DBLOW.

ORA-12413

ORA-12413 **cannot lower DBHIGH below** *num*

Cause: An attempt was made to lower DBHIGH below its current value. DBHIGH can only be raised.

Action: Do not attempt to lower DBHIGH.

ORA-12414

ORA-12414   must be DBHIGH *name* to startup database

Cause: An attempt was made to open a database while at an O/S label not equal to DBHIGH.

Action: Log in at DBHIGH to start the database.

ORA-12415

ORA-12415   **session label must dominate the creation label of role** *name*

Cause: An attempt was made to grant a privilege to a role while at a label that does not dominate the role's creation label. Your DBMS label must dominate the role's creation label to grant a privilege to the role.

Action: Check the creation label of the role and perform the grant while at a label that dominates the creation label of the role.

ORA-12416

ORA-12416   **label of audit must dominate user**

Cause: An attempt was made to set statement audit options on a user below the label where the user was created.

Action: Check the label of the user and set the audit options at the user label or higher.

ORA-12418

ORA-12418   **label of index must equal table**

Cause: An attempt was made to create an index on a table at a label different from where the table was created. It is necessary to be at the identical label and specifically at DBHIGH or DBLOW if the table was created at DBHIGHor DBLOW.

Action: Check the label of the table and create the index at the correct label.

ORA-12419

ORA-12419 **label of column must equal table**

Cause: An attempt was made to add a column to a table at a different label than where the table was created.

Action: Check the label of the table and add the column at the correct label.

ORA-12420

ORA-12420	**label of stored object must dominate referenced object** *name*

Cause: An attempt was made to create a stored object, such as a view, procedure, function trigger, that references some object not dominated by the current label. While the referenced object may be visible due to enabled MAC privileges, MAC privileges cannot be used for creating the stored object.

Action: Check the label of the indicated object and create the new object at a label that dominates all referenced objects.

ORA-12421

ORA-12421   **label of grant must dominate object**

Cause: An attempt was made to grant privileges on an object at a label lower than where the object was created.

Action: Check the label of the object and perform the grant at the object label or higher.

ORA-12422

ORA-12422   **label of grant must dominate grantee** *name*

Cause: An attempt was made to grant privileges to a user at a label lower than where the user was created.

Action: Check the label of the grantee and perform the grant at that label or higher.

ORA-12423

ORA-12423 **label of constraint must equal table**

Cause: An attempt was made to add a constraint to a table at a label different from the label of the table.

Action: Check the label of the table and add the constraint at the label of the table.

ORA-12424

ORA-12424   **label of object must dominate tablespace**

Cause: An attempt was made to create an object in a tablespace created at a label higher than the object.

Action: Check the label of the tablespace and create the object at that label or higher.

ORA-12425

ORA-12425   **label of table must dominate cluster**

Cause: An attempt was made to create a table in a cluster created higher than the label of the table.

Action: Check the label of the cluster and create the table at that label or higher.

ORA-12426

ORA-12426   **label of user must dominate default or temporary tablespace**

Cause: An attempt was made to give a user a default or temporary tablespace at a higher label.

Action: Check the label of the default and temporary tablespaces and create the user at that label or higher.

ORA-12427

ORA-12427   **label of object must dominate schema**

Cause: An attempt was made to create an object in a schema defined at a label higher than the object.

Action: Check the label of the schema and create the object at that label or higher.

ORA-12428

ORA-12428 **insufficient MAC privileges**

Cause: There are sufficient privileges to read the object, but no MAC write access to the object.

Action: Either retry the operation at the label of the object or see the system administrator for the needed MAC privilege.

ORA-12429

ORA-12429   **label of comment must equal table or view**

Cause: An attempt was made to comment a table or view at a label different from the label at which the table or view was created.

Action: Check the label of the table or view and add the comment at that label.

ORA-12430

ORA-12430	label of trigger must equal table

Cause: An attempt was made to create a trigger at a label different from where the table was created. Your DBMS label must match the table's creation label to create the trigger on the table. While the table may be visible due to enabled MAC privileges, MAC privileges cannot be used to create a trigger.

Action: Check the creation label of the table and create the trigger while at a label that matches the creation label of the table.

ORA-12431

ORA-12431   **label of user must dominate default profile**

Cause: An attempt was made to give a user a default profile created at a label lower than the user.

Action: Check the label of the profile and create the user at that label or higher.

ORA-12432

ORA-12432 **label of default role list must dominate user**

Cause: An attempt was made to assign a user a default role list at a label higher than the user. Your DBMS label must dominate the user's creation label to assign a default role list to a user.

Action: Check the creation label of the user and assign the default role list while at a label that dominates the creation label of the user.

ORA-12433

ORA-12433   **label of default role *name* must dominate user**

Cause: An attempt was made to assign a user a default role created at a label below where the user was created.

Action: Check the labels of the user and role and assign only roles higher than the user.

ORA-12434

ORA-12434   **label of child record must dominate parent**

Cause: An attempt was made to insert a child record at a label lower than the corresponding parent record.

Action: Check the label of the parent key value and insert the child record at that label or higher.

ORA-12435

ORA-12435   **label of package body must equal package**

Cause: An attempt was made to create a package body at a label different from the corresponding package.

Action: Check the label of the package and create the package body at that label.

ORA-12437

ORA-12437 **cannot write down to a database at a lower label**

Cause: An attempt was made to perform a write operation in a secondary database. Write operations are possible only in the primary database.

Action: Connect directly to the database to perform the operation.

ORA-12438

ORA-12438   **insufficient privileges to validate constraint on all rows**

Cause: It is necessary to have access to all rows in a table to attempt to enable a constraint.

Action: To read all rows in the table, either enable MAC privileges or downgrade all rows in the table.

ORA-12439

ORA-12439   cannot modify ROWLABEL column

Cause: An attempt was made to modify the ROWLABEL column.

Action: None. It is not possible to modify the ROWLABEL column.

ORA-12440

ORA-12440   **database has not been mounted in read-compatible mode**

Cause: The requested secondary database was not last started inread-compatible mode.

Action: Set the DB_MOUNT_MODE initialization parameter to READ_COMPATIBLE in the secondary database and startup the secondary database.

ORA-12441

ORA-12441 **grant already exists at a different label**

Cause: One of the granted privileges was previously granted at a different label.

Action: Check USER_SYS_GRANTS or USER_OBJ_GRANTS to locate the label at which the privilege was previously granted. Revoke the old grant and regrant it at the new label.

ORA-12442

ORA-12442   **must be at same label as user to modify default roles**

Cause: An attempt was made to modify a user's default roles at a different label than where the user was created.

Action: Check the ALL_USERS view to see the label at which the user was created and alter the user at that label.

ORA-12443

ORA-12443 **audit option already exists at a different label**

Cause: You tried to audit an event that was previously audited at a different label.

Action: Check the DBA_STMT_AUDIT_OPTS or DBA_PRIV_AUDIT_OPTS dictionary views to identify the label of the previous audit statement. If you want to audit the event at the new label, stop the old audit, using the NOAUDIT statement, and re-audit the event while at the new label.

ORA-12444

ORA-12444   **file label is lower than instance label**

Cause: You tried to reuse a file with a label different from the database instance label.

Action: Examine the message stack for the filename. Remove the file or relabel it to match the database instance label.

ORA-12445

ORA-12445   **index/row label mismatch - see trace file**

Cause: A label in an index has been found not to match the row to which that label points. This may indicate that index labels have been directly changed, which is inappropriate.

Action: To solve the immediate problem, you can drop and recreate the index. If you are unsure of the problem's cause, contact customer support and send them the trace file.

ORA-12446

ORA-12446	**file label is higher than instance label**

Cause: You tried to reuse a file with a label different from the database instance label. You can reuse only files labeled at or below the instance label.

Action: Examine the message stack for the filename. Remove the file or relabel it to match the database instance label.

ORA-12447

ORA-12447   **non-secure protocol used; connection denied**

Cause: You tried to connect to Trusted Oracle7 with a networking protocol that was not secure. You must use a secure networking protocol to connect via a remote location.

Action: Either connect to the database with a secure networking protocol or log in directly to the database.

ORA-12449

ORA-12449   **least upper bound resulted in an invalid operating system label**

Cause: You tried to do an operation that generated a least upper bound (LUB) label, which is not a valid label on your operating system.

Action: Examine the message stack for a more specific message. See the *Trusted Oracle7 Server Administrator's Guide, Release 7.1* for information on least upper-bound (LUB) functions. For more information on invalid label generation, see your operating system-specific Oracle documentation.

ORA-12450

ORA-12450   **label string could not be translated into binary label**

Cause: The operating system could not translate given label string.

Action: Check that the specified string is a valid label and retry the operation.

ORA-12451

ORA-12451   **binary OS label could not be translated into string label**

Cause: The operating system could not translate the specified binary label.

Action: Check that the specified string is a valid label and retry the operation.

ORA-12452

ORA-12452 **label format string could not be parsed; check format codes**

Cause: An invalid format string on a label function was specified.

Action: Check that a valid format string is specified and retry the operation. See the *Trusted Oracle7 Server Administrator's Guide, Release 7.1* for a complete list of valid format strings.

ORA-12453

ORA-12453   **missing quote in literal string**

Cause: A literal string was specified with an incorrect number of quotes.

Action: Check that the string has matching quotes and retry the operation.

ORA-12454

ORA-12454 **punctuation supplied does not match specified input format**

Cause: The punctuation supplied in the label string is inconsistent with the format specified.

Action: Change the punctuation in the label string or the format, whichever is appropriate, and retry the statement.

ORA-12455

ORA-12455   **OS format not allowed in this context**

Cause: The label format string supplied has an OS format that is not allowed with the TO_LABEL function.

Action: Change the label format string to not use the OS format type.

ORA-12456

ORA-12456   **literal string supplied does not match specified input format**

Cause: The literal string supplied is inconsistent with the format string specified.

Action: Change the literal in the label string or format, whichever is appropriate, and retry the statement.

ORA-12457

ORA-12457   **invalid length for MLSLABEL bind variable**

Cause: A bind variable of type MLSLABEL is too long.

Action: Consult your documentation for proper datatype lengths.

ORA-12458

ORA-12458 **the input string could not be translated into a binary label**

Cause: The input string you specified is not a valid binary label on youroperating system.

Action: Check the input string, change any components so that they form a valid label, and retry the operation. Refer to your Oracle7 operating system-specific documentation for more information.

ORA-12459

ORA-12459   **maximum number of columns in a Trusted Oracle table is 253**

Cause: A CREATE or ALTER TABLE statement tried to add more than 253 columns, not counting ROWLABEL, to a table.

Action: Specify no more than 253 columns in your CREATE or ALTERTABLE command.

ORA-12460

ORA-12460   **the value *str* for the MLS_LABEL_FORMAT parameter is not valid**

Cause: The value for the MLS_LABEL_FORMAT initialization parameter you specified does not contain a valid format mask. You must have a valid MLS_LABEL_FORMAT parameter for the database to start up.

Action: Change this parameter setting in your initialization parameter file to a correct value.

ORA-12461

ORA-12461 **invalid binary label**

Cause: You provided a binary label in an OCI program's bind statement that was not a valid operating system label.

Action: Replace the invalid binary label with one that is valid. Refer to your Oracle7 operating system-specific documentation for more information.

ORA-12470

ORA-12470   **process label could not be found**

Cause: An error was encountered in translating the operating system process label into a label number.

Action: Examine the message stack for a more specific message. Refer to your Oracle7 operating system-specific documentation for more information.

ORA-12475

ORA-12475   cannot log in below DBLOW *name*

Cause: An attempt was made to log in at a label dominated by DBLOW.

Action: Either log in at a higher label or lower DBLOW.

ORA-12476

ORA-12476   **cannot log in above DBHIGH** *name*

Cause: The user attempted to log in at a label dominating DBHIGH.

Action: Either log in at a lower label or raise DBHIGH.

ORA-12478

ORA-12478 **cannot insert label above DBHIGH** *name*

Cause: An attempt was made to insert a value explicitly above DBHIGH into the ROWLABEL column or to specify a value higher of datatype MLSLABEL. Any label inserted into a Trusted Oracle7 table must fall between DBLOW and DBHIGH, inclusive.

Action: Either raise the value of DBHIGH or specify a lower label on the INSERT or UPDATE statement.

ORA-12480

ORA-12480 **secondary database changing between OS MAC and DBMS MAC modes**

Cause: An attempt was made to access a secondary database while the secondary database is changing between OS MAC and DBMS MAC modes.

Action: Retry the operation after the change is complete.

ORA-12485

ORA-12485 **cannot find file label for** *name*

Cause: An error was encountered while reading the label of the specified file.

Action: Examine the message stack for an operating system message. Refer to your Oracle7 operating system-specific documentation for more information.

ORA-12487

ORA-12487   **insufficient privileges to set label to DBHIGH or DBLOW**

Cause: It is only possible to set a label to DBHIGH or DBLOW when connected INTERNAL to Trusted Oracle7.

Action: Connect INTERNAL to Trusted Oracle7, then set the label DBHIGH or DBLOW. If this is not possible due to insufficient privileges, contact the database administrator.

ORA-12488

ORA-12488   **session label must be equal to** *name*

Cause: An attempt was made to log in to an OS MAC database at other than the database label.

Action: Log in from an OS process at the specified label.

ORA-12489

ORA-12489 **insufficient MAC privilege to alter session**

Cause: User does not have sufficient MAC privileges to alter session.

Action: Either change the OS label so ALTER SESSION is not needed or obtain MAC privileges.

ORA-12490

ORA-12490   **cannot connect at session label below user definition** *name*

Cause: An attempt was made to alter a session label below the label at which the user was created.

Action: Either do not alter the session label below the specified label or recreate the user at a lower label.

ORA-12491

ORA-12491 **cannot change MAC mode when mounted in READ_COMPATIBLE mode**

Cause: You tried to change between OS and DB MAC mode while the database was mounted in READ_COMPATIBLE mode.

Action: Set the DB_MOUNT_MODE initialization parameter to NORMAL and restart the database.

ORA-12494

ORA-12494 **cannot mount secondary database labeled above DBLOW**

Cause: The secondary database requested was labeled above the DBLOW of the primary database.

Action: Check that that DBHIGH on the secondary database is below DBLOW of the primary database.

ORA-12495

ORA-12495 **multi-threaded server cannot be used with Trusted Oracle**

Cause: The MTS_DISPATCHERS parameter is set in the initialization file, which initiated the configuration of the multi-threaded server.

Action: Remove the MTS_DISPATCHERS parameter from the initialization file and restart the instance.

12500-12699: SQL*Net

See your operating system-specific Oracle documentation. For a listing of other Oracle manuals containing operating system-specific messages, see "Related Publications" in the Preface.

12700-12799: National Language Support

This section lists messages generated when the National Language Support feature of Oracle is accessed.

ORA-12700

ORA-12700   **invalid NLS parameter value** *name*

Cause: An invalid NLS initialization parameter was specified.

Action: Check the syntax and spelling of the parameter. Enter the correct parameter into the initialization parameter file.

ORA-12701

ORA-12701 **CREATE DATABASE character set is not known**

Cause: The character set specified in the CREATE DATABASE command does not exist on the system.

Action: Refer to your Oracle7 operating system-specific documentation for more information about character sets on the system. Use a valid character set in the CREATE DATABASE command.

ORA-12702

ORA-12702   **invalid NLS parameter string used in SQL function**

Cause: An invalid parameter or value was entered in the NLS parameter string in a SQL function.

Action: Check the syntax of the function and the parameters. Enter the command again using the correct syntax and valid values.

ORA-12703

ORA-12703   **this character set conversion is not supported**

Cause: An attempt was made to use the CONVERT function to convert between two character sets, and the requested conversion is not possible due to unavailability of one or both character sets.

Action: Use character sets available on the system in the CONVERT function.

ORA-12705

ORA-12705 **invalid or unknown NLS parameter value specified**

Cause: There are two possible causes: Either an attempt was made to issue an ALTER SESSION statement with an invalid NLS parameter or value; or the NLS_LANG environment variable contains an invalid language, territory, or character set.

Action: Check the syntax of the ALTER SESSION command and the NLS parameter, correct the syntax and retry the statement, or specify correct values in the NLS_LANG environment variable.

ORA-12706

ORA-12706   **this CREATE DATABASE character set is not allowed**

Cause: An attempt was made to create a database on an ASCII-based machine using an EBCDIC-based character set or vice versa. This is not permitted.

Action: Create the database using either ASCII-based or EBCDIC-based character sets, depending on what is supported by the hardware. Refer to your Oracle7 operating system-specific documentation for more information about the type of character set supported.

12800-12849: Parallel Query/Index Creation

This section lists some of the messages generated during a parallel query or parallel index creation.

ORA-12800

ORA-12800   **system appears too busy for parallel query execution**

Cause: The load on the system is too high to perform parallel queries.

Action: Re-enter the statements serially or wait until the system load is reduced.

ORA-12801

ORA-12801   **error signaled in parallel query server** *name*

Cause: A parallel query server encountered an exception as described in the accompanying message(s).

Action: Correct the error described in the accompanying message.

ORA-12802

ORA-12802   **parallel query server lost contact with coordinator**

Cause: A parallel query server was unable to communicate with the server coordinating the parallel query.

Action: Check for operating system or network errors and retry the statement. If this problem persists, contact customer support.

ORA-12803

ORA-12803 **parallel query server lost contact with another server**

Cause: A parallel query server was unable to communicate with another server involved in the parallel query.

Action: Check for operating system or network errors and retry the statement. If this problem persists, contact customer support.

ORA-12804

ORA-12804   **parallel query server appears to have died**

Cause: Process information for a parallel query server thread cannot be found.

Action: Check for operating system errors and retry the statement. If this problem persists, contact customer support.

ORA-12805

ORA-12805   **parallel query server died unexpectedly**

Cause: The PMON process is cleaning up the process because a parallel query server terminated unexpectedly.

Action: Check for operating system errors and retry the statement. If this problem persists, contact customer support.

ORA-12806

ORA-12806   **could not get background process to hold enqueue**

Cause: This is an internal error that is not usually issued.

Action: Contact customer support.

ORA-12807

ORA-12807 **process queue could not receive parallel query message**

Cause: This is an internal error that is not usually issued.

Action: Contact customer support.

ORA-12808

ORA-12808 **cannot set [*CACHE* | *SCAN*]_INSTANCES greater than number of instances *num***

Cause: An attempt was made, using the ALTER SYSTEM command, to set CACHE_INSTANCES or SCAN_INSTANCES to a value greater than the number of available instances.

Action: See the accompanying message for the allowable maximum value or set SCAN_INSTANCES or CACHE_INSTANCES to ALL.

ORA-12809

ORA-12809 **cannot set [CACHE | SCAN]_INSTANCES when mounted inexclusive mode**

Cause: An attempt was made to set CACHE_INSTANCES or SCAN_INSTANCES using the ALTER SYSTEM command while the database was mounted in exclusive mode.

Action: Only set CACHE_INSTANCES or SCAN_INSTANCES when the database is mounted in PARALLEL or SHARED mode.

ORA-12810

ORA-12810 **PARALLEL_MAX_SERVERS must be less than or equal to** *num*

Cause: An attempt was made to set the PARALLEL_MAX_SERVERS initialization parameter to a value higher than the maximum allowed by the system.

Action: See the accompanying message for the allowable maximum value.

ORA-12811

ORA-12811   PARALLEL_MIN_SERVERS must be less than or equal to PARALLEL_MAX_SERVERS, *num*

Cause: An attempt was made to set the PARALLEL_WIN_SERVERS initialization parameter to a value higher than PARALLEL_MAX_SERVERS.

Action: See the accompanying message for the allowable maximum value.

ORA-12812

ORA-12812 **only one PARALLEL or NOPARALLEL clause may be specified**

Cause: PARALLEL was specified more than once, NOPARALLEL was specified more than once, or both PARALLEL and NOPARALLEL were specified in a CREATE TABLE, CREATE CLUSTER, CREATE INDEX, ALTER TABLE, ALTER CLUSTER statement, or a RECOVER command.

Action: Specify PARALLEL or NOPARALLEL only once.

ORA-12813

ORA-12813	**value for PARALLEL or DEGREE must be greater than 0**

Cause: PARALLEL 0 or DEGREE 0 was specified in a CREATE TABLE, CREATE CLUSTER, CREATE INDEX, ALTER TABLE, ALTER CLUSTER statement, or a RECOVER command.

Action: Specify a value greater than 0 or specify default parallelism using PARALLEL with no degree or using DEGREE DEFAULT within a PARALLEL clause.

ORA-12814

ORA-12814 **only one CACHE or NOCACHE clause may be specified**

Cause: CACHE was specified more than once, NOCACHE was specified more than once, or CACHE and NOCACHE, were specified in a CREATE TABLE, CREATE CLUSTER, CREATE INDEX, ALTER TABLE, ALTER CLUSTER, or a RECOVER command.

Action: Specify CACHE or NOCACHE only once.

ORA-12815

ORA-12815 **value for INSTANCES must be greater than 0**

Cause: PARALLEL parameter specifying number of instances must be a positive integer or DEFAULT.

Action: Specify a positive integer or DEFAULT for INSTANCES if parallelism across instances is desired.

ORA-12817

ORA-12817   **parallel query option must be enabled**

Cause: You tried to attempt an operation that requires the parallel query option.

Action: Enable the parallel query option.

ORA-12818

ORA-12818   **invalid option in PARALLEL clause**

Cause: An unrecognized option was specified with a PARALLEL clause.

Action: Specify any combination of DEGREE {<n> | DEFAULT } and INSTANCE { <n> | DEFAULT } within the PARALLEL clause.

ORA-12819

ORA-12819 **missing options in PARALLEL clause**

Cause: The PARALLEL clause was specified without an option.

Action: Specify any combination of DEGREE { <n> | DEFAULT } and INSTANCE { <n> | DEFAULT } within the PARALLEL clause.

ORA-12820

ORA-12820   **invalid value for DEGREE**

Cause: An invalid value for DEGREE was specified within a PARALLEL clause.

Action: Specify a positive integer or DEFAULT.

ORA-12821

ORA-12821   invalid value for INSTANCES

Cause: An invalid value for INSTANCES was specified within a PARALLEL clause.

Action: Specify a positive integer or DEFAULT.

ORA-12822

ORA-12822 **duplicate option in PARALLEL clause**

Cause: DEGREE or INSTANCE was specified more than once within a PARALLEL clause.

Action: Specify the option only once.

ORA-12823

ORA-12823   **default degree of parallelism may not be specified here**

Cause: Either an ALTER DATABASE RECOVER command was specified with the DEGREE DEFAULT option in a PARALLEL clause or another command was specified with no option in the PARALLEL clause.

Action: Specify an explicit degree of parallelism.

ORA-12824

ORA-12824 **INSTANCES DEFAULT may not be specified here**

Cause: An ALTER DATABASE RECOVER command was specified with the INSTANCES DEFAULT option.

Action: Specify an explicit value for INSTANCES or omit the INSTANCES option if single instance recovery is desired.

ORA-12825

ORA-12825   **explicit degree of parallelism must be specified here**

Cause: An ALTER DATABASE RECOVER command was specified without the DEGREE option.

Action: Specify an explicit degree of parallelism.

ORA-12826

ORA-12826   hung parallel query server was killed

Cause: Parallel query server was hung and subsequently killed.

Action: Re-execute query and report suspicious events noted in the trace file to customer support if error persists.

ORA-12827

ORA-12827   **parallel query option must be enabled**

Cause: A parallel query option feature has been invoked, but this option has not been enabled.

Action: Enable the parallel query option.

19999-21000: Stored Procedures

This section lists some of the messages generated when stored procedures are accessed.

ORA-19999

ORA-19999   **SKIP_ROW procedure was called**

Cause: This message is raised when the SKIP_ROW procedure is called.

Action: This procedure should be called only within a trigger or a procedure called by a trigger.

ORA-20000

ORA-20000   **application_specific_message**

Cause: The stored procedure RAISE_APPLICATION_ERROR was called, which causes this message to be generated.

Action: Correct the problem as described in the message or contact the application administrator or database administrator for more information.

ORA-21000

ORA-21000 **error number argument to RAISE_APPLICATION_ERROR of *name* is out of range, must be between -20000 and -20999, inclusive**

Cause: An attempt was made to specify a number not in the range of -20000 to -20999.

Action: The valid range for user-specified error numbers is -20000 to -20999 inclusively. Do not specify numbers outside this range.

21100-21299: Internal Messages

23300-24299: DBMS PL/SQL Packages

This section lists messages generated by the DBMS_SYS_ERROR package.

ORA-23300

ORA-23300   **application_specific_message**

Cause: The stored procedure RAISE_SYSTEM_ERROR was called to generate this message.

Action: Correct the problem described in the message or contact the application programmer or database administrator for more information.

ORA-23301

ORA-23301   **mixed use of deferred RPC destination modes**

Cause: Replication catalog determined deferred remote procedure call destinations were mixed with destination determined by other mechanisms in the same transaction.

Action: Do not mix destination types in the same transaction.

ORA-23302

ORA-23302   **application raised communication failure during deferred RPC**

Cause: An application declared a communication failure during a deferred remote procedure call.

Action: Retry the application when communication is restored.

ORA-23303

ORA-23303   **application raised generic exception during deferred RPC**

Cause: An application declared a generic failure during a deferred remote procedure call.

Action: Dependent on application.

ORA-23304

ORA-23304   **malformed deferred RPC at arg *name* of *name* in call *name*, in tid *name***

Cause: A deferred remote procedure call was issued without the correct number of arguments as determined by the count parameter DBMS_DEFER.CALL.

Action: Ensure the number of arguments matches the count.

ORA-23305

ORA-23305   internal deferred RPC error: *str*

Cause: An internal error occurred in the deferred remote procedure call.

Action: Contact customer support.

ORA-23306

ORA-23306   schema **"name" does not exist**

Cause: The schema name was null or misspelled, or the schema does not exist locally.

Action: Specify the schema correctly, or create it with CREATE USER.

ORA-23307

ORA-23307   **replicated schema *"name"* already exists**

Cause: The given database already replicates the given schema.

Action: Choose a different schema or a different database.

ORA-23308

ORA-23308 object *"name"* does not exist or is invalid

Cause: The given name was null or misspelled, the given type was wrong, the object does not exist as a valid database object, or the object does not exist as a replicated object with the appropriate status.

Action: Ensure the object is valid in the database, is visible to the user, and, if appropriate, is a valid object in ALL_REPOBJECT.

ORA-23309

ORA-23309   object *"name"* of type *name* exists

Cause: An object in the same namespace exists, perhaps with a different type or shape.

Action: Drop the offending object or reinvoke the request using TRUE for a boolean parameter such as RETRY or USE_EXISTING_OBJECT.

ORA-23310

ORA-23310   **replicated schema *"name"* is not quiesced**

Cause: The requested operation requires the replicated schema to be suspended.

Action: Invoke SUSPEND_MASTER_ACTIVITY at the repschema's masterdef, wait until the status has changed to quiesced, and then retry the original request.

ORA-23311

ORA-23311   **replicated schema _"name"_ is quiesced**

Cause: SUSPEND_MASTER_ACTIVITY has been called before the replicated schema has resumed normal operation.

Action: If a RESUME_MASTER_ACTIVITY request is pending, wait until it completes, and then re-invoke SUSPEND_MASTER_ACTIVITY.

ORA-23312

ORA-23312   **not the masterdef according to *name***

Cause: The schema name is null, the schema name is misspelled, the invocation or given database is not the masterdef, or one of the masters does not believe the invocation database is the masterdef.

Action: If the given schema name and masterdef were both correct, connect to the masterdef and retry the request, or relocate the masterdef at the errant databases using RELOCATE_masterdef.

ORA-23313

ORA-23313   schema *"name"* **is not mastered at** *name*

Cause: The schema name is null, the schema name is misspelled, the invocation database is not a master, or the invocation database does not believe the given database is a master.

Action: If the given schema name was correct, connect to a current master and retry the request, make the invocation database a master with ADD_MASTER_DATABASE, or use SWITCH_SNAPSHOT_MASTER if the invocation database is a snapshot site.

ORA-23314

ORA-23314 database is not a snapshot site for *name*

Cause: The invocation database is not a snapshot database for the given replicated schema.

Action: Connect to the desired snapshot database and retry the request, or make the invocation database a snapshot site with CREATE_SNAPSHOT_REPSCHEMA.

ORA-23315

ORA-23315   **repcatlog version or request** *name* **is not supported by version** *name*

Cause: Either incompatible repcat versions are used, or a REPCATLOG record has been corrupted.

Action: Convert the master to a compatible version of repcat or retry the request.

ORA-23316

ORA-23316 **the masterdef is** *name*

Cause: One of the masters to be removed is the masterdef.

Action: Relocate the masterdef to a master that will not be removed, and then reinvoke the REMOVE_MASTER_DATABASES request at the new masterdef.

ORA-23317

ORA-23317   a communication failure has occurred

Cause: The remote database is inaccessible.

Action: Ensure the remote database is running, the communications network is functioning, and the appropriate database links are present.

ORA-23318

ORA-23318 a DDL failure has occurred

Cause: User-supplied or system-generated DDL did not execute successfully.

Action: Examine DDL, database state, REPCATLOG, and ALL_ERRORS to determine why the failure occurred.

ORA-23319

ORA-23319 **parameter value *name* is not appropriate**

Cause: The given value of a parameter is either null, misspelled, or not supported.

Action: Refer to the documentation and use parameter values that are appropriate for the given situation.

ORA-23320

ORA-23320   **the request failed because of values *name* and *name***

Cause: A missing DDL record for a REPCATLOG record, or inconsistency in REPCAT views.

Action: Retry the request, or make the views consistent.

ORA-23323

ORA-23323   **parameter length exceeds deferred remote procedure call limits**

Cause: A deferred remote procedure call parameter was longer than the deferred remote procedure call limit of 2000 bytes for CHAR/VARCHAR2 parameters and 255 bytes for raw parameters.

Action: Use smaller parameters.

ORA-23324

ORA-23324   **error *num*, while creating DEFERROR entry at *name* with error *num***

Cause: The given error was encountered while attempting to create a DEFEROR entry for the given error code and the given database.

Action: Correct the cause of the given error.

ORA-23325

ORA-23325 **parameter type is not** *type*

Cause: A function in DBMS_DEFER_SYS_QUERY was called to retrieve a deferred remote procedure call parameter from the deferred remote procedure call queue, but the type of the parameter does not match the return type ofthe function.

Action: Use the function corresponding to the parameter type.

ORA-23326

ORA-23326   **the system is being quiesced**

Cause: A deferred remote procedure call operation was attempted while the database was quiesced.

Action: Resume database activity with the DBMS_REPCAT.RESUME_MASTER_ACTIVITY call.

ORA-23327

ORA-23327   **imported deferred remote procedure call data does not match *id* of importing db**

Cause: Deferred remote procedure call queues were imported from a database with a different global name or operating system than the importing database.

Action: Deferred remote procedure call data should only be imported into a database with the same global name and hardware and operating system.

ORA-23328

ORA-23328   **snapshot name *name.name* must be same as replicated master table**

Cause: When creating a snapshot through repcat, the snapshot name did not match a replicated table name at the master.

Action: Change the snapshot DDL to use the same name as the replicated table name at the master.

ORA-23330

ORA-23330   **column group *name* already exists**

Cause: The column group was already registered in the replicated schema.

Action: Use a column group name not yet registered in the replicated schema.

ORA-23331

ORA-23331	column group *name* does not exist

Cause: The given column group is either null, misspelled or not registered.

Action: Use a registered column group.

ORA-23332

ORA-23332   **group *name* is in use; cannot drop**

Cause: The given column group or priority group is being used to resolve conflicts.

Action: Call DBMS_REPCAT procedures DROP_UPDATE_RESOLUTION, DROP_DELETE_RESOLUTION, DROP_UNIQUE_RESOLUTION so that the column group or priority group is no longer in use before dropping.

ORA-23333

ORA-23333 **column** *name* **is already part of a column group**

Cause: Attempted to add a column to a column group when the column was already a member of a column group.

Action: Drop the column from its existing column group before trying to add itto another.

ORA-23334

ORA-23334   **column *name* does nor exist in table or column group**

Cause: The given column is either null, misspelled or is not part of the given table or column group.

Action: Use a column that is a member of the table or column group.

ORA-23335

ORA-23335   **priority group *name* already exists**

Cause: The priority group was already registered in the replicated schema.

Action: Use a column group name not yet registered in the replicated schema.

ORA-23336

ORA-23336   **priority group *name* does not exist**

Cause: The priority group was already registered in the replicated schema.

Action: Use a priority group name not yet registered in the replicated schema.

ORA-23337

ORA-23337 **priority or value not in priority group** *name*

Cause: The specified value or priority has not been registered as part of the priority group.

Action: Either specify a different value or priority that is already part of the priority group, or add the value to the priority group.

ORA-23338

ORA-23338   **priority or value already in priority group** *name*

Cause: The specified value or priority has already been registered as part of the priority group.

Action: Either specify a different value or priority that is not already part of the priority group, or drop the value from the priority group.

ORA-23339

ORA-23339   **duplicate conflict resolution information**

Cause: The specified combination of column group, sequence, conflict type and/or parameter table name, parameter column name, and parameter sequence number has already been registered.

Action: Verify that additional conflict resolution information needs to be added and provide a new sequence number. If modifying existing information, the existing information must be dropped first.

ORA-23340

ORA-23340 **incorrect resolution method** *name*

Cause: User function is specified when conflict resolution method was not "USER FUNCTION" or specified resolution method is not one of the predefined methods.

Action: If user function is specified when conflict resolution method was not "USER FUNCTION", either reregister function with method as "USER FUNCTION" or specify a NULL user function. Otherwise, specify one of the documented supported conflict resolution methods.

ORA-23341

ORA-23341   **user function required**

Cause: A NULL user function was specified for the "USER FUNCTION" method.

Action: Provide user function name, e.g., "schema"."package"."function", that conforms to the documented user function specifications or specify one of the documented supported conflict resolution methods.

ORA-23342

ORA-23342   **invalid parameter column** *name*

Cause: The parameter column name is null or misspelled, the invocation databaseis not a master, or is of the wrong type for the specified conflict resolution method.

Action: Specify a parameter column from the specified column group that has a correct type for the conflict resolution method.

ORA-23343

ORA-23343   **no match for specified conflict resolution information**

Cause: The specified combination of column group, sequence, conflict type has not been registered, for example, for adding a comment.

Action: Specify a combination of column group, sequence, conflict type that has been registered.

ORA-23344

ORA-23344   **constraint *name.name* does not exist**

Cause: A null, misspelled or nonexistent constraint was specified when registering a uniqueness conflict.

Action: Register a named constraint for the specified table.

ORA-23345

ORA-23345   **table *name.name* not registered to collect statistics**

Cause: A procedure that deals with conflict resolution statistics-gathering was called for a table that was not registered to collect statistics.

Action: Call DBMS_REPCAT.REGISTER_STATISTICS to register the table.

ORA-23346

ORA-23346   **primary key is undefined for table** *name*

Cause: Trying to generate replication support for a table without a primary key as defined by a constraint or DBMS_REPCAT_COLUMNS.

Action: Add a primary key constraint to the table or define a primary key using DBMS_REPCAT_COLUMNS.

ORA-23347

ORA-23347 **datatype *name* for column *name* table *name* not supported**

Cause: The table has a column whose datatype is not supported by repcat.

Action: Remove the column from the table, or alter the column to have one of the supported datatypes.

ORA-23348

ORA-23348   **cannot replicate procedure *name*; only IN parameters supported**

Cause: Trying to generate replication support for a package that has a procedure with OUT or IN OUT parameters.

Action: Remove the procedure from the package, or remove the OUT or IN OUT parameters from the procedure.

ORA-23349

ORA-23349   **cannot generate replication support for functions**

Cause: Trying to generate replication support for a package that has a public function, or for a stand-alone function.

Action: Remove the public function from the package, or alter the function to be a procedure.

ORA-23350

ORA-23350	maximum number of recursive calls exceeded

Cause: This usually occurs when trying to resolve conflicts in a table while concurrent updates to the same row create more conflicts.

Action: Re-execute the deferred transaction from DEFERROR using DBMS_DEFER_SYS.EXECUTE_ERROR.

ORA-23351

ORA-23351   **parameter datatype *name* for procedure *name* not supported**

Cause: The procedure has a parameter whose datatype is not supported by repcat.

Action: Remove the parameter from the procedure, or alter the parameter to have one of the supported datatypes.

ORA-23352

ORA-23352   **duplicate destination for deferred transaction**

Cause: A duplicate destination was specified for a deferred transaction either in a DBMS_DEFER.CALL call or an earlier DBMS_DEFER.TRANSACTION call or a DBMS_DEFER_SYS.ADD_DEFAULT_DEST call.

Action: Remove the duplicate entry.

ORA-23353

ORA-23353   deferred RPC queue has entries for replicated schema *name*

Cause: The requested action cannot be performed until the queue is empty for the given replicated schema.

Action: Use DBMS_DEFER_SYS.EXECUTE or DBMS_DEFER_SYS.DELETE_TRAN to empty the queue.

ORA-23354

ORA-23354   deferred RPC execution disabled for *name*

Cause: The execution of a deferred remote procedure call at the destination failed because its propagation was disabled.

Action: Enable deferred remote procedure call execution with the DBMS_DEFER_SYS.ENABLE call.

ORA-23364

ORA-23364 **replication option not installed**

Cause: The replication option was not installed at this site. Updatable snapshots, deferred remote procedure calls, and other replication features were, therefore, unavailable.

Action: Install the replication option. The replication option is not part of the Oracle7 Server product and must be purchased separately. Contact an Oracle Sales representative if the replication option needs to be purchased.

ORA-23400

ORA-23400   **invalid snapshot name "*name*"**

Cause: A null, misspelled, or badly formed snapshot name was given to DBMS_SNAPSHOT.REFRESH.

Action: Provide a valid snapshot name to DBMS_SNAPSHOT.REFRESH.

ORA-23401

ORA-23401   snapshot "*name.name*" does not exist

Cause: A snapshot name was given to DBMS_SNAPSHOT.REFRESH that is notin SYS.SNAP$ or its associated views.

Action: Provide a snapshot name that is in SYS.SNAP$, ALL_SNAPSHOTS, or USER_SNAPSHOTS.

ORA-23402

ORA-23402   **refresh was aborted because of conflicts caused by deferred transactions**

Cause: There are outstanding conflicts logged in the DEFERROR tableat the snapshot's master.

Action: Resolve the conflicts in the master DEFERROR table and refresh again after the table is empty. Alternatively, refresh with REFRESH_AFTER_ERRORS set to TRUE, which will proceed with the refresh, even if there are conflicts in the master's DEFERROR table. Proceeding despite conflicts can result with an updatable snapshot's changes appearing to be temporarily lost, until a refresh succeeds after theconflicts are resolved.

ORA-23403

ORA-23403   refresh group "*name.name*" already exists. Refresh group not a stored object?

Cause: Making a new refresh group when there is already a group ofthe same name in SYS.RGROUP$.

Action: Choose a different refresh group name.

ORA-23404

ORA-23404   **refresh group "*name.name*" does not exist**

Cause: A refresh group name was given that is not in SYS.RGROUP$.

Action: Provide a refresh group name that is in SYS.RGROUP$ or DBS_RGROUP.

ORA-23405

ORA-23405   **refresh group number *name* does not exist**

Cause: A refresh group number was given that is not in SYS.RGROUP$.

Action: Provide a refresh group number that is in SYS.RGROUP$ or DBS_RGROUP.

ORA-23406

ORA-23406   **insufficient privileges on user "*name*"**

Cause: The caller is not the owner of the snapshot and does not haveALTER ANY SNAPSHOT privileges.

Action: Perform the operation as the owner of the snapshot or as auser with ALTER ANY SNAPSHOT privileges.

ORA-23407

ORA-23407	object name *name* must be shaped like "SCHEMA.OBJECT" or "OBJECT"
Cause:	The object name, for example, the rollback segment, the snapshot name, or the refresh group, was incorrectly specified.
Action:	Retry the operation with the object name properly specified, like "SCHEMA.OBJECT" or "OBJECT".

ORA-23409

ORA-23409   **could not find an unused refresh group number**

Cause: 1000 consecutive refresh group numbers, as defined by theRGROUPSEQ number, were already used by rows in SYS.RGROUP$.

Action: Alter the sequence number to be within a legal unused rangeand destroy unneeded refresh groups.

ORA-23410

ORA-23410   **snapshot *"name.name"* is already in a refresh group**

Cause: A snapshot of the same name is already in a refresh group.

Action: Subtract the snapshot its the current refresh group and add itto its new refresh group, or combine the two refresh groups into a single refresh group.

ORA-23411

ORA-23411	**snapshot "*name.name*" is not in refresh group "*name.name*"**
Cause:	The specified snapshot is not in the specified refresh group.
Action:	Try again with the proper snapshot and refresh group names.

ORA-23420

ORA-23420   **interval must evaluate to a time in the future**

Cause: The parameter INTERVAL evaluates to a time earlier than SYSDATE.

Action: Choose an expression that evaluates to a time later than SYSDATE.

ORA-23421

ORA-23421   job number *num* is not a job in the job queue

Cause: There is no job visible to the caller with the given job number.

Action: Choose the number of a job visible to the caller.

ORA-23422

ORA-23422   **Oracle Server could not generate an unused job number**

Cause: Oracle Server could not generate a job number that was not used to identify another job.

Action: Retry the operation.

ORA-23423

ORA-23423 **job number *num* is not positive**

Cause: The given job number is less than 1.

Action: Choose a positive integer.

ORA-23430

ORA-23430 **the replicated schema _"name"_ does not exist**

Cause: The caller has provided the name of a replicated schema that is not known at the master site as an argument to a routine in offline instantiation of a schema.

Action: Check that the replicated schema exists at the master site where the user is executing the procedure, and retry the call.

ORA-23431

ORA-23431   site *name* is not the master definition site

Cause: The master site where the routine was executing is not the master definition site.

Action: Find out the master definition site for the replicated schema in question, and re-run the routine.

ORA-23432

ORA-23432 **wrong state:** *name*

Cause: Routine is executing against a replicated schema whose state is in the wrong state.

Action: Make sure that the replicated schema is in the state given in the error message.

ORA-23433

ORA-23433 **master site *name* already exists**

Cause: Trying to instantiate a replicated schema at a master site that is already a part of the schema.

Action: If you were trying to add this site, do nothing because it already exists; otherwise, pick the name of another site, and re-run the routine.

ORA-23434

ORA-23434 **executing against wrong master site** *name*

Cause: Caller provides an argument indicating the site against which the routine should be executing; trying to execute the routine at a site that is different from that argument.

Action: Execute against the correct master site.

ORA-23435

ORA-23435 **malformed master site name** *name*

Cause: Site name as argument to routine is not formed correctly.

Action: Please consult the Oracle manuals on how to construct site names.

ORA-23436

ORA-23436 **site *name* is not master site**

Cause: The new site given as an argument to a routine is not a master site for the given replicated schema.

Action: Execute DBMS_OFFLINE_SCHEMA.BEGIN_INSTANTIATION() routine to add a new site to the replicated schema.

ORA-23437

ORA-23437   **master site *name* not known for replicated schema**

Cause: The site name given as an argument to a routine is not already known to the replicated schema.

Action: Execute DBMS_OFFLINE_SCHEMA.BEGIN_INSTANTIATION() routine to add a new site to the replicated schema.

ORA-24275

ORA-24275 **function** *name* **parameter** *name* **missing or invalid**

Cause: The function <pkgname.funcname> was called with a parameter <parname> that was null, 0 length, or had an invalid value.

Action: Correct the parameter to supply values that comply with its datatype and limits as specified in the documentation.

ORA-24276

ORA-24276 **function** *name* **output** *name* **maximum value exceeded**

Cause: The function <pkgname.funcname> computed a value for the output <parname> that exceeded the maximum allowed. This can occur when multiple input parameters, each valid separately, combine to specify an invalid result. For example, when a length parameter multiplied by a copies parameter yields a total length exceeding the maximum for the output datatype.

Action: Correct the input values to produce a result that will comply with the limits as specified in the documentation.

24300-24999: UPI/OCI Messages

ORA-24300

ORA-24300 **bad value for mode**

Cause: The mode value specified was undefined.

Action: Verify the mode and provide one of the defined values.

PL/SQL and FIPS Messages

This chapter lists messages issued by PL/SQL. For each PL/SQL message, the probable cause and corrective action are given.

To help you find and fix errors, PL/SQL inserts object names, numbers, and character strings into some error messages. These message inserts are represented by *name*, *num*, and *str*, respectively. For example, the error message listed as

```
PLS-00388: undefined column name in subquery
```

might be issued as

```
PLS-00388: undefined column AMPNO in subquery
```

This chapter also lists the warnings that can be generated when the FIPS flagger is set in an Oracle product, and an Oracle extension to the SQL92 standard is used or a SQL92 feature is used in a non-conforming manner. FIPS warnings messages are listed for completeness. The cause and actions are described in the ANSI document ANSI X3.135-1992.

PL/SQL Error Messages

PLS-00101

PLS-00101   reserved for future use

Not in Release 2.2.

PLS-00102

PLS-00102   **parser stack overflow because nesting is too deep**

Cause: The parser, which checks the syntax of PL/SQL statements, uses a data structure called a stack; the number of levels of nesting in the PL/SQL block exceeded the stack capacity.

Action: Reorganize the block structure to avoid nesting at too deep a level. For example, move the lowest-level sub-block to a higher level.

PLS-00103

PLS-00103   **found *str* but expected one of the following: *str***

Cause: This error message is from the parser. It found a token (language element) that is inappropriate in this context.

Action: Check previous tokens as well as the one given in the error message. The line and column numbers given in the error message refer to the end of the faulty language construct.

PLS-00104

PLS-00104   **empty argument list in call of procedure *name* must be omitted**

Cause: In a subprogram call, the name of the subprogram was followed by an empty parameter list. For example, procedure P was called as P(). This is not allowed.

Action: Remove the empty parameter list. In the example, change the procedure callto P.

PLS-00105

PLS-00105   at most one forward declaration of type *name* is permitted

Not in Release 2.2.

PLS-00108

PLS-00108   **declarative units must be a single variable declaration**

Cause: While checking a declarative unit (a top-level declare block without the BEGIN...END), PL/SQL found that there was more than one item declared or that the item was not a variable declaration. A table is a common variable declaration at the unit level. To define a TABLE, compile a DECLARE compilation unit, but only one at a time is allowed.

Action: Declare variables in separate declarative units.

PLS-00109

PLS-00109   **unknown exception name *name* in PRAGMA EXCEPTION_INIT**

Cause: No declaration for the exception name referenced in an EXCEPTION_INIT pragma was found within the scope of the pragma.

Action: Make sure the pragma follows the exception declaration and is within thesame scope.

PLS-00110

PLS-00110   **bind variable *name* not allowed in this context**

Cause: A bind variable, that is, an identifier prefixed with a colon, was found in an inappropriate context.

Action: Remove the colon or replace the bind variable with the appropriate object.

PLS-00111

PLS-00111   **end-of-file in comment**

Cause: A comment had a comment initiator (/*), but before the comment terminator (*/) was found, an end-of-file marker was encountered.

Action: Remove the comment initiator or add a comment terminator. The line and column numbers accompanying the error message refer to the beginning of the last legal token before the comment initiator.

PLS-00112

PLS-00112 **end-of-line in quoted identifier**

Cause: A quoted identifier had a beginning quote ("), but before the ending quote (") was found, an end-of-line marker was encountered.

Action: Remove the beginning quote or add the ending quote. The line and column numbers accompanying the error message refer to the beginning of the quoted identifier.

PLS-00113

PLS-00113   **END identifier *name1* must match *name2* at line *num*, column *num***

Cause: Following the keyword END, which terminates some language constructs (such as loops, blocks, functions, and procedures), you can optionally place the name of that construct. For example, at the end of the definition of loop L you might write END L.

This error occurs when the optional name does not match the name given to the language construct. It is usually caused by a misspelled identifier or by faulty block structure.

Action: Make sure the spelling of the END identifier matches the name given to the language construct and that the block structure is correct.

PLS-00114

PLS-00114   **identifier *name* too long**

Cause: The name of a PL/SQL variable is longer than 30 characters. Legal identifiers (including quoted identifiers) have a maximum length of 30 characters. A string literal might have been mistakenly enclosed in double quotes instead of single quotes, in which case PL/SQL considers it a quoted identifier.

Action: Shorten the name to 30 characters or less. If a string literal is being used, replace the double quotes with single quotes.

PLS-00115

PLS-00115   **this PRAGMA must follow the declaration of** *name*

Cause: The pragma refers to a PL/SQL object that was not declared or is not within the scope of the reference. Identifiers must be declared before they are used in a pragma; forward references are *not* allowed.

Action: Check the spelling and declaration of the identifier. Also confirm that the declaration is placed correctly in the block structure.

PLS-00116

PLS-00116   **duplicate WHERE clause in DELETE, SELECT, or UPDATE statement**

Cause: Two or more WHERE clauses were found in a DELETE, SELECT, orUPDATE statement. The WHERE clause specifies a condition under which rows in a table are processed. The condition can contain several logical expressions connected by AND or OR, but a statement can contain only one WHERE clause.

Action: Remove one of the WHERE clauses and, if necessary, connect logical expressions by AND or OR.

PLS-00117

PLS-00117   **duplicate CONNECT BY clause in SELECT statement**

Cause: Two or more CONNECT BY clauses were found in a SELECT statement. The CONNECT BY clause defines a relationship used to return rows in a hierarchical order. The relationship can contain two expressions separated by a relational operator (such as = or !=), but a statement can contain only one CONNECT BY clause.

Action: Remove one of the CONNECT BY clauses and, if necessary, separate expressions by a relational operator.

PLS-00118

PLS-00118   **duplicate GROUP BY clause in SELECT statement**

Cause: Two or more GROUP BY clauses were found in a SELECT statement. The GROUP BY clause lists column expressions used to form a summary row for each group of selected rows. The list can contain several column expressions separated by commas, but a statement can contain only one GROUP BY clause.

Action: Remove one of the GROUP BY clauses and, if necessary, separate column expressions by commas.

PLS-00119

PLS-00119   **duplicate HAVING clause in SELECT statement**

Cause: Two or more HAVING clauses were found in a SELECT statement. The HAVING clause specifies a condition under which groups of rows (formed by the GROUP BY clause) are included in the result. The condition can include several logical expressions connected by AND or OR, but a statement can contain only one HAVING clause.

Action: Remove one of the HAVING clauses and, if necessary, connect logical expressions by AND or OR.

PLS-00120

PLS-00120   **inappropriate argument in OPEN statement**

Cause: The *cursor_name* parameter in an OPEN statement is misspelled or does not refer to a legally declared cursor.

Action: Check the spelling of the *cursor_name* parameter. Make sure the cursor was declared properly.

PLS-00121

PLS-00121   **only the set function COUNT may take * as an argument**

Cause: The asterisk (*) option was used in the argument list of a SQL group function other than COUNT. For example, the code might look like

```
SELECT SUM(*) INTO emp_count FROM emp;  -- should be COUNT(*)
```

Only COUNT permits the use of the asterisk option, which returns the number of rows in a table.

Action: Remove the asterisk option from the argument list and replace it with an expression that refers to one or more database columns.

PLS-00123

PLS-00123   **program too large**

Cause: PL/SQL was designed primarily for robust transaction processing. One consequence of the special-purpose design is that the PL/SQL compiler imposes a limit on block size. The limit depends on the mix of statements in the PL/SQL block. Blocks that exceed the limit cause this error.

Action: The best solution is to modularize the program by defining subprograms, which can be stored in an Oracle database. Another solution is to break the program into two sub-blocks. Have the first block INSERT any data the second block needs into a temporary database table. Then, have the second block SELECT the data from the table.

PLS-00124

PLS-00124   **name of exception expected for first argument in EXCEPTION_INIT pragma**

Cause: The first argument passed to the EXCEPTION_INIT pragma was something other than an exception name. The first argument must be the name of a legally declared exception.

Action: Replace the first argument with the name of a legally declared exception.

PLS-00125

PLS-00125   **type *name* expected**

Cause: When a constant or variable was declared, its datatype was not specified. For example, the code might look like

```
pi  CONSTANT := 3.14159;  -- should be CONSTANT REAL := 3.14159
```

Every constant and variable must have a datatype, which specifies a storage format, constraints, and valid range of values.

Action: Supply the missing datatype specifier.

PLS-00201

PLS-00201   **identifier *name* must be declared**

Cause: An attempt was made to reference an undefined variable, exception, procedure, function, or other object. Either the identifier was not declared or it was not within the scope of the reference.

Action: Check the spelling and declaration of the identifier. Also confirm that the declaration is placed correctly in the block structure.

PLS-00202

PLS-00202   **type *name* must be declared**

Cause: An attempt was made to reference an undefined type. Either the type specifier was not declared or it is not within the scope of the reference.

Action: Check the spelling and declaration of the type specifier. Also confirm that the declaration is placed correctly in the block structure.

PLS-00203

PLS-00203   **function DECODE must be called with at least 3 non-Boolean arguments**

Cause: Less than three arguments were passed to the built-in function DECODE. Though DECODE takes a variable number of (non-Boolean) arguments, at least three arguments must be passed.

Action: Call DECODE with three or more arguments.

PLS-00204

PLS-00204 **function or pseudocolumn *name* may be used inside a SQL statement only**

Cause: A pseudocolumn or proscribed function was used in a procedural statement. The SQL pseudocolumns (CURRVAL, LEVEL, NEXTVAL, ROWID, ROWNUM) can be used only in SQL statements. Likewise, certain functions such as DECODE, DUMP, and VSIZE and the SQL group functions (AVG, MIN, MAX, COUNT, SUM, STDDEV, VARIANCE) can be used only in SQL statements.

Action: Remove the pseudocolumn reference or function call from the procedural statement. Or, replace the procedural statement with a SELECT INTO statement; for example, replace

```
bonus := DECODE(rating, 1, 5000, 2, 2500, ...);
```

with the following statement:

```
SELECT DECODE(rating, 1, 5000, 2, 2500, ...) INTO bonus FROM dual;
```

PLS-00205

PLS-00205   **aggregate not allowed here**

Cause: An aggregate, that is, a parenthesized list of values such as (7788, 'SCOTT', 20), was found in an inappropriate context.

Action: Remove or relocate the aggregate.

PLS-00206

PLS-00206   **%TYPE must be applied to a variable or column, not** *name*

Cause: The program object declared using the %TYPE datatype attribute is not of the appropriate class. It must be a variable, column, record component, subprogram formal parameter, or other object to which values can be assigned.

Action: Declare an object of the appropriate class or define the datatype in another way (for example, use %ROWTYPE).

PLS-00207

PLS-00207   **identifier *name*, applied to implicit cursor SQL, is not a legal cursor attribute**

Cause: An identifier that is not a cursor attribute was applied to the identifier SQL. For example, this error occurs if the cursor attribute is misspelled.

Action: Check the spelling of the cursor attribute name. Make sure the attribute is one of these: %NOTFOUND, %FOUND, %ROWCOUNT, %ISOPEN.

PLS-00208

PLS-00208   **identifier *name* is not a legal cursor attribute**

Cause: An identifier not declared as a cursor attribute was applied to an identifier declared as a cursor. For example, this error occurs if the cursor attributeis misspelled.

Action: Check the spelling of the cursor attribute name. Make sure the attribute is one of these: %NOTFOUND, %FOUND, %ROWCOUNT, %ISOPEN.

PLS-00209

PLS-00209   **table *name* is not in FROM clause**

Cause: In a query, a table referenced by the select list is not named in the FROM clause.

Action: Check the spelling of the table names, make sure each column in the select list refers to a table in the FROM clause, then re-execute the query.

PLS-00210

PLS-00210   **an OTHERS clause is required in this CASE statement**

Not in Release 2.2.

PLS-00211

PLS-00211   **CASE labels or ranges must not be duplicated in different WHEN clauses**

Not in Release 2.2.

PLS-00212

PLS-00212   **could not obtain enough memory to compile CASE statement**

Not in Release 2.2.

PLS-00213

PLS-00213 **package STANDARD not accessible**

Cause: The PL/SQL compiler could not find package STANDARD in the current Oracle database. To compile a program, PL/SQL needs package STANDARD.

Action: Make sure that package STANDARD is available in the current Oracle database, then retry the operation.

PLS-00214

PLS-00214   **BEGIN...END block nesting is too deep**

Cause: The number of levels of nesting in the PL/SQL block is too large. Blocks can be nested up to 255 levels deep, depending on the availability of system resources such as memory.

Action: Reorganize the block structure to avoid nesting at too deep a level. For example, move the lowest-level sub-block to a higher level.

PLS-00215

PLS-00215   **string length constraints must be in range (1 .. 32767)**

Cause: When a character variable was declared, a length outside the legal range was specified. For example, the following declarations are illegal:

```
flag  CHAR(0);        -- illegal; zero length
name  VARCHAR2(-10);  -- illegal; negative length
```

Action: Change the length constraint, making sure that it lies in the range 1 .. 32767

PLS-00216

PLS-00216 **NUMBER precision constraint must be in range (1 .. 38)**

Cause: A NUMBER variable was declared with a precision that is outside the legal range. Declarations such as N NUMBER(800) or N NUMBER(123,10) are not supported.

Action: Change the illegal NUMBER precision constraint, making sure that it lies in the range 1 .. 38.

PLS-00217

PLS-00217 **NUMBER scale constraint must be in range (-84 .. 127)**

Cause: A NUMBER variable was declared with a scale that is outside the legalrange. Declarations such as N NUMBER(10,345) or N NUMBER(10,-100) are not supported.

Action: Change the illegal NUMBER scale constraint, making sure that it lies in the range -84 .. 127.

PLS-00218

PLS-00218   a variable declared NOT NULL must have an initialization assignment

Cause: In general, variables that have no initialization clause in their declaration are automatically initialized to NULL. This is illogical for NOT NULL variables; therefore, an initialization clause is required.

Action: Add an initialization clause to the variable declaration.

PLS-00219

PLS-00219 **label *name* reference is out of scope**

Cause: A block or loop label was used to qualify a variable (as in *outer_block.date*) that was not declared or is not within the scope of the label. The variable name might be misspelled, its declaration might be faulty, or the declaration might be placed incorrectly in the block structure.

Action: Check the spelling and declaration of the variable name. Also confirm that the declaration is placed correctly in the block structure.

PLS-00220

PLS-00220   **simple name required in this context**

Cause: A qualified name such as A.B or A.B.C is not permitted here.

Action: Use a simple name such as A instead.

PLS-00221

PLS-00221   *name* **is not a procedure or is undefined**

Cause: An identifier being referenced as a procedure was not declared or actually represents another object (for example, it might have been declared as a function).

Action: Check the spelling and declaration of the identifier. Also confirm that the declaration is placed correctly in the block structure.

PLS-00222

PLS-00222   **no function with name *name* exists in this scope**

Cause: An identifier being referenced as a function was not declared or actually represents another object (for example, it might have been declared as a procedure).

Action: Check the spelling and declaration of the identifier. Also confirm that the declaration is placed correctly in the block structure.

PLS-00223

PLS-00223   **parameterless procedure** *name* **used as function**

Cause: An identifier being referenced as a parameterless function actually represents a procedure.

Action: Check the spelling and declaration of the identifier. Also confirm that the declaration is placed correctly in the block structure. If necessary, change the declaration of the identifier or change the reference so that it does not require a return value.

PLS-00224

PLS-00224   **object *name* must be of type function or array to be used this way**

Cause: An identifier being referenced as a function or an array actually represents an object (a number or date, for example) that cannot be referenced in this way.

Action: Check the spelling and declaration of the identifier. Also confirm that the declaration is placed correctly in the block structure.

PLS-00225

PLS-00225   **subprogram or cursor *name* reference is out of scope**

Cause: A subprogram or cursor references a variable that was not declared or is not within the scope of the subprogram or cursor. The variable name might be misspelled, its declaration might be faulty, or the declaration might be placed incorrectly in the block structure.

Action: Check the spelling and declaration of the variable name. Also confirm that the declaration is placed correctly in the block structure.

PLS-00226

PLS-00226 **package *name* used as variable reference**

Cause: A package was referenced in an expression as if it were a variable or function. Either the name of the variable or function is misspelled or the reference is not fully qualified. For example, to call the function *my_function*, which is stored in package *my_package*, dot notation must be used, as follows:

```
... my_package.my_function ...
```

Action: Correct the spelling of the variable or function name or use dot notation to reference the packaged variable or function.

PLS-00227

PLS-00227   **IN formal parameter *name* not allowed in this context**

Cause: When declaring the formal parameters of a subprogram, one parameter was used to initialize another, as in

```
PROCEDURE my_proc (j NUMBER, k NUMBER := j) IS ...
```

The first parameter has no value until run time, so it cannot be used to initialize another parameter.

Action: Remove the illegal formal parameter reference.

PLS-00229

PLS-00229   **attribute expression within SQL expression**

Cause: An attribute expression, such as SQL%NOTFOUND, was used in a SQL statement, but attribute expressions are allowed only in procedural statements.

Action: To work around this limitation, assign the value of the attribute expression toa variable, then use the variable in the SQL statement. For example, replacethe statement

```
INSERT INTO audits VALUES (c1%ROWCOUNT, ...);
```

with the following statements:

```
row_count := c1%ROWCOUNT;
INSERT INTO audits VALUES (row_count, ...);
```

PLS-00230

PLS-00230   **OUT and IN OUT formal parameters may not have default expressions**

Cause: When declaring the formal parameters of a procedure, an OUT or IN OUT parameter was initialized to a default value, as in

```
PROCEDURE calc_bonus (bonus OUT REAL := 0, ...) IS ...
```

However, only IN parameters can be initialized to default values.

Action: Remove the illegal default expression.

PLS-00231

PLS-00231   **function *name* may not be used in SQL**

Cause: A proscribed function was used in a SQL statement. Certain functions such as SQLCODE and SQLERRM can be used only in procedural statements.

Action: Remove the function call from the SQL statement. Or, replace the function call with a local variable. For example, the following statement is illegal:

```
INSERT INTO errors VALUES (SQLCODE, SQLERRM);
```

However, you can assign the values of SQLCODE and SQLERRM to local variables, then use the variables in the SQL statement, as follows:

```
err_num := SQLCODE;
err_msg := SQLERRM;
INSERT INTO errors VALUES (err_num, err_msg);
```

PLS-00232

PLS-00232   **nested packages not permitted**

Cause: A package was declared inside another package, but package declarations are allowed only at the top level. In other words, packages cannot be nested.

Action: Move the package declaration outside the enclosing package.

PLS-00233

PLS-00233   **function name used as an exception name in WHEN clause**

Cause: The WHEN clause in an exception handler contains a function call instead of an exception name. A valid exception handler consists of a WHEN clause, which must specify an exception, followed by a sequence of statements to be executed when that exception is raised.

Action: Check the spelling of the identifier in the WHEN clause, then replace the function call with an exception name.

PLS-00302

PLS-00302   **component *name* must be declared**

Cause: In a reference to a component (for example, in the name A.B, B is a component of A), the component was not declared. The component might be misspelled, its declaration might be faulty, or the declaration might be placed incorrectly in the block structure.

Action: Check the spelling and declaration of the component. Also confirm that the declaration is placed correctly in the block structure.

PLS-00303

PLS-00303	**qualifier *name* must be declared**

Cause: In a name such as A.B, A is a qualifier, and B is a component of the qualifier. This error occurs when no declaration for the qualifier is found. The qualifier might be misspelled, its declaration might be faulty, or the declaration might be placed incorrectly in the block structure.

Action: Check the spelling and declaration of the qualifier. Also confirm that the declaration is placed correctly in the block structure.

PLS-00304

PLS-00304 **cannot compile body of *name* without its specification**

Cause: The compiled package specification required to compile a package body could not be found. Some possible causes follow:

- the package name is misspelled
- the package specification was never compiled
- the compiled package specification is not accessible

 The package specification must be compiled before compiling the package body, and the compiler must have access to the compiled specification.

Action: Check the spelling of the package name. Compile the package specification before compiling the package body. Also, make sure the compiler has access to the compiled specification.

PLS-00305

PLS-00305   **previous use of *name* conflicts with this use**

Cause: While looking for prior declarations of a cursor, procedure, function, or package, the compiler found another object with the same name in the same scope. Or, the headers of subprogram in a package specification and body do not match word for word

Action: Check the spelling of the cursor, procedure, function, or package name. Also check the names of all constants, variables, parameters, and exceptions declared in the same scope. Then, remove or rename the object with the duplicate name. Or, change the headers of the packaged subprogram so that they match word for word.

PLS-00306

PLS-00306 **wrong number or types of arguments in call to** *name*

Cause: This error occurs when the named subprogram call cannot be matched to any declaration for that subprogram name. The subprogram name might be misspelled, a parameter might have the wrong datatype, the declaration might be faulty, or the declaration might be placed incorrectly in the block structure. For example, this error occurs if the built-in square root function SQRT is called with a misspelled name or with a parameter of the wrong datatype.

Action: Check the spelling and declaration of the subprogram name. Also confirm that its call is correct, its parameters are of the right datatype, and, if it is not a built-in function, that its declaration is placed correctly in the block structure.

PLS-00307

PLS-00307   **too many declarations of *name* match this call**

Cause: The declaration of a subprogram name is ambiguous because there was no exact match between the declaration and the call and more than one declaration matched the call when implicit conversions of the parameter datatypes were used. The subprogram name might be misspelled, its declaration might be faulty, or the declaration might be placed incorrectly in the block structure.

Action: Check the spelling and declaration of the subprogram name. Also confirm that its call is correct, its parameters are of the right datatype, and, if it is not a built-in function, that its declaration is placed correctly in the block structure.

PLS-00308

PLS-00308 **this construct is not allowed as the origin of an assignment**

Cause: The construct or expression does not designate a value that can be assigned to a variable. For example, the datatype name NUMBER cannot appear on the right hand side of an assignment statement as in X := NUMBER.

Action: Correct the illegal assignment statement.

PLS-00309

PLS-00309   **with %LAST attribute, *name* must be a variable of an enumerated type**

Not in Release 2.2.

PLS-00310

PLS-00310   **with %ROWTYPE attribute, *name* must name a cursor or table**

Cause: The %ROWTYPE attribute must be applied to an identifier declared as a cursor, cursor variable, or database table. This error occurs when %ROWTYPE follows some identifier that has not been so declared.

Action: Change the declaration or do not apply the %ROWTYPE attribute tothe identifier.

PLS-00311

PLS-00311   **the declaration of the type of *name* is incomplete or malformed**

Cause: This occurrence of the identifier cannot be compiled because its datatype has not been properly defined.

Action: Correct the faulty datatype declaration.

PLS-00312

PLS-00312   **a positional parameter association may not follow a named association**

Cause: When a list of parameters is passed to a subprogram or cursor, if both positional and named associations are used, all positional associations must be placed in their declared order and before all named associations, which can be in any order.

Action: Reorder the parameter list to meet the requirements or use namedassociation only.

PLS-00313

PLS-00313   *name* **not declared in this scope**

Cause: There is no declaration for the given identifier within the scope of reference. The identifier might be misspelled, its declaration might be faulty, or the declaration might be placed incorrectly in the block structure.

Action: Check the spelling and declaration of the identifier. Also confirm that the declaration is placed correctly in the block structure.

PLS-00314

PLS-00314   **TABLE declarations are not allowed as PL/SQL local variables**

Cause: In a precompiled program, the DECLARE TABLE statement was mistakenly used inside an embedded PL/SQL block. If an embedded PL/SQL block refers to a database table that does not yet exist, use the DECLARE TABLE statement to tell the precompiler what the table will look like. However, DECLARE TABLE statements are allowed only in the host program.

Action: Move the DECLARE TABLE statement outside the embedded PL/SQL block. If you want a variable that can store an entire row of data selected from a database table or fetched from a cursor or cursor variable, use the %ROWTYPE attribute.

PLS-00315

PLS-00315   **PL/SQL TABLE declarations must currently use BINARY_INTEGER indexes**

Cause: In the INDEX BY clause of a PL/SQL table declaration, a datatype other than BINARY_INTEGER was specified. PL/SQL tables can have one column and a primary key. The column can have any scalar type, but the primary key must have type BINARY_INTEGER.

Action: Change the datatype specifier to BINARY_INTEGER.

PLS-00316

PLS-00316   **PL/SQL TABLE declarations must currently use a single index**

Cause: In the INDEX BY clause of a PL/SQL table declaration, a composite primary key was specified. PL/SQL tables must have a simple, unnamed primary key of type BINARY_INTEGER.

Action: Change the faulty clause to INDEX BY BINARY_INTEGER.

PLS-00319

PLS-00319   **subquery in an IN or NOT IN clause must contain exactly one column**

Cause: An invalid expression such as

```
a IN (SELECT x, y, z FROM ... )
```

was used. When a [NOT]IN clause is used with a subquery, it does not test for set membership. The number of expressions in the [NOT]IN clause and the subquery select list must match. So, in the example above, the subquery must specify at most one column.

Action: Change the subquery to select only one column.

PLS-00320

PLS-00320 **the declaration of the type of this expression is incomplete or malformed**

Cause: In a declaration, the name of a variable or cursor is misspelled or the declaration makes a forward reference. Forward references are not allowed in PL/SQL. A variable or cursor must be declared *before* it is referenced it in other statements, including other declarative statements. For example, the following declaration of *dept_rec* raises this exception because it refers to a cursor not yet declared:

```
DECLARE
   dept_rec  dept_cur%ROWTYPE;
   CURSOR dept_cur IS SELECT ...
   ...
```

Action: Check the spelling of all identifiers in the declaration. If necessary, move the declaration so that it makes no forward references.

PLS-00321

PLS-00321   **expression *str* is inappropriate as the left hand side of anassignment statement**

Cause: The expression does not designate a variable that can have a value assigned to it. For example, the function SYSDATE cannot appear on the left hand side of an assignment statement such as

```
SYSDATE := '01-JAN-1990';
```

Action: Correct the illegal assignment statement.

PLS-00322

PLS-00322 **declaration of a constant *name* must contain an initialization assignment**

Cause: A constant declaration lacks the assignment of an initial value. For example, in the following declaration" := 3.14159" is the initialization clause:

```
pi CONSTANT NUMBER := 3.14159;
```

Action: Correct the constant declaration by supplying the missinginitialization assignment.

PLS-00323

PLS-00323 **subprogram *name* is declared in a package specification and must be defined in the package body**

Cause: A subprogram specification was placed in a package specification, but the corresponding subprogram body was not placed in the package body. The package body implements the package specification. So, the package body must contain the definition of every subprogram declared in the package specification.

Action: Check the spelling of the subprogram name. If necessary, add the missing subprogram body to the package body.

PLS-00324

PLS-00324   **cursor attribute may not be applied to non-cursor *name***

Cause: This error occurs when a cursor attribute (%FOUND, %NOTFOUND, %ROWCOUNT, or %ISOPEN) appears following an identifier not declared as a cursor or cursor variable. It occurs, for example, if the variable name *my_cur* in *my_cur*%FOUND was not properly declared as a cursor or if the variable declaration was placed incorrectly in the block structure.

Action: Check the spelling and declaration of the identifier. Also confirm that the declaration is placed correctly in the block structure.

PLS-00325

PLS-00325 **non-integral numeric literal *num* is inappropriate in this context**

Cause: A non-integer numeric literal was used in a context that requires an integer (a number with no decimal point).

Action: Replace the inappropriate literal with an integer literal.

PLS-00326

PLS-00326   **IN clause must contain same number of expressions as subquery**

Cause: The number of expressions in an IN clause did not equal the number of expressions in a corresponding subquery select list. For example, the following statement is invalid because the IN clause contains two expressions, but the subquery select list contains just one:

```
... WHERE (ename, sal) IN (SELECT sal FROM emp);
```

Action: Check the number of expressions in each set, then revise the statement to make the numbers equal.

PLS-00328

PLS-00328   **a subprogram body must be defined for the forward declaration of** *name*

Cause: A subprogram specification was declared, but the corresponding subprogram body was not defined. Write the subprogram specification and body as a unit. An alternative solution is to separate the specification from its body, which is necessary when you want to define mutually recursive subprograms or you want to group subprograms in a package.

Action: Check the spelling of the subprogram name. If necessary, supply the missing subprogram body.

PLS-00330

PLS-00330	invalid use of type *name* or subtype *name*

Cause: A datatype or subtype specifier was mistakenly used in place of a constant, variable, or expression. For example, the code might look like

```
IF emp_count > number THEN ... -- illegal; NUMBER is a datatype specifier
```

Action: Replace the datatype or subtype specifier with a valid constant, variable,or expression.

PLS-00335

PLS-00335   **a package name conflicts with an existing object**

Cause: In a CREATE PACKAGE statement, a package (which is a database object that groups logically related PL/SQL types, objects, and subprograms) was given the same name as an existing database object.

Action: Give the package a unique name.

PLS-00341

PLS-00341   **declaration of cursor *name* is incomplete or malformed**

Cause: A cursor declaration is improper or an identifier referenced in the cursor declaration was not properly declared. A return type that does not refer to an existing database table or a previously declared cursor or cursor variable might have been specified. For example, the following cursor declaration is illegal because *c1* is not yet fully defined:

```
CURSOR c1 RETURN c1%ROWTYPE IS SELECT ...  -- illegal
```

In this case, a return type does not have to be specified because it is implicit.

Action: Check the spelling and declaration of the cursor name and any identifiers referenced in the cursor declaration. Also confirm that the declaration is placed correctly in the block structure. If a return type was specified, make sure that it refers to an existing database table or a previously declared cursor or cursor variable.

PLS-00351

PLS-00351   **not logged on to database** *name*

Cause: An attempt was made to access an Oracle database without being logged on. Probably, an invalid username or password was entered.

Action: Log on to Oracle with a correctly spelled username and password before trying to access the database.

PLS-00352

PLS-00352   **unable to access another database** *name*

Cause: An attempt was made to reference an object in a database other than the current local or remote Oracle database.

Action: Correct the reference and make sure the object is in the current Oracle database.

PLS-00353

PLS-00353   *name* **must name a user in the database**

Cause: This error occurs when the username was misspelled or when the user does not exist in the database.

Action: Check the spelling of the username and make sure the user exists.

PLS-00354

PLS-00354 **username must be a simple identifier**

Cause: A qualified username such as *scott.accts* is not permitted in this context.

Action: Specify a simple username such as *scott* instead.

PLS-00355

PLS-00355   **use of PL/SQL table not allowed in this context**

Cause: A PL/SQL table was referenced in the wrong context (for example, with a remote link).

Action: Remove the PL/SQL table reference or change the context.

PLS-00356

PLS-00356 *name* **must name a table to which the user has access**

Cause: The named table is not accessible to the user. This error occurs when the table name or username was misspelled, the table and/or user does not exist in the database, the user was not granted the necessary privileges, or the table name duplicates the name of a local variable or loop counter.

Action: Check the spelling of the table name and username. Also confirm that the table and user exist, the user has the necessary privileges, and the table name does not duplicate the name of a local variable or loop counter.

PLS-00357

PLS-00357   **table, view or sequence reference *name* not allowed in this context**

Cause: A reference to database table, view, or sequence was found in an inappropriate context. Such references can appear only in SQL statements or (excluding sequences) in %TYPE and %ROWTYPE declarations. Some valid examples follow:

```
SELECT ename, emp.deptno, dname INTO my_ename, my_deptno, my_dept
   FROM emp, dept WHERE emp.deptno = dept.deptno;
DECLARE
   last_name  emp.ename%TYPE;
   dept_rec   dept%ROWTYPE;
```

Action: Remove or relocate the illegal reference.

PLS-00358

PLS-00358   **column *name* exists in more than one table; use qualifier**

Cause: The statement is ambiguous because it specifies two or more tables having the same column name. For example, the following statement is ambiguous because *deptno* is a column in both tables:

```
SELECT deptno, loc INTO my_deptno, my_loc FROM emp, dept;
```

Action: Precede the column name with the table name (as in *emp.deptno*) so that the column reference is unambiguous.

PLS-00359

PLS-00359   **assignment target in *str* must have components**

Cause: An assignment target was declared that lacks the components needed to store the assigned values. For example, this error occurs if you try to assign a row of column values to a variable instead of a record, as follows:

```
DECLARE
   dept_rec    dept%ROWTYPE;
   my_deptno   dept.deptno%TYPE;
   ...
BEGIN
   SELECT deptno, dname, loc INTO my_deptno   -- invalid
      FROM dept WHERE ...
   ...
```

Action: Check the spelling of the names of the assignment target and all its components. Make sure the assignment target is declared with the required components and that the declaration is placed correctly in the block structure.

PLS-00360

PLS-00360   **cursor declaration without body needs return type**

Cause: A cursor declaration lacks either a body (SELECT statement) or a return type. If you want to separate a cursor specification from its body, a return type must be supplied, as in

```
CURSOR c1 RETURN emp%ROWTYPE;
```

Action: Add a SELECT statement or return type to the cursor declaration.

PLS-00361

PLS-00361   **a cursor variable must be IN OUT to modify its state**

Cause: A cursor variable was declared as the IN or OUT formal parameter of a subprogram that, when called, will change the state of the cursor variable. In such cases, the cursor variable must be declared as an IN OUT parameter.

Action: Change the parameter mode from IN or OUT to IN OUT.

PLS-00363

PLS-00363   **expression _str_ cannot be used as an assignment target**

Cause: A literal, constant, IN parameter, loop counter, or function call was mistakenly used as the target of an assignment. For example, the following statement is illegal because the assignment target, 30, is a literal:

```
SELECT deptno INTO 30 FROM dept WHERE ...  -- illegal
```

Action: Correct the statement by using a valid assignment target.

PLS-00364

PLS-00364   **loop index variable *name* use is invalid**

Cause: A reference to a loop counter was found in an inappropriate context. For example, the following statement is illegal because the loop counter is used as the terminal value in its own range expression:

```
FOR j IN 1 .. j LOOP ...  -- illegal
```

Action: Change the loop range expression so that it does not reference the loop counter. If you want to refer in the range expression to another variable with the same name as the loop counter, change either name or qualify the variable name with a label.

PLS-00365

PLS-00365 *name* **is an OUT parameter and cannot be read**

Cause: An attempt was made to assign the value of an OUT parameter to another parameter or variable. Inside a procedure, an OUT parameter acts like an uninitialized variable; therefore, its value cannot be read. For example, the following assignments are illegal:

```
PROCEDURE calc_bonus (bonus OUT REAL, ...) IS
   rating  REAL;
   wages   REAL;
BEGIN
   ...
   IF rating > 90 THEN
      bonus := bonus * 2;  -- illegal
      SELECT sal + bonus INTO wages FROM emp ...  -- illegal
      ...
   END IF;
   ...
END calc_bonus;
```

Action: Use an IN OUT parameter instead of the OUT parameter. Inside a procedure, an IN OUT parameter acts like an initialized variable; therefore, its value canbe read.

PLS-00366

PLS-00366   **subtype of a NOT NULL type must also be NOT NULL**

Cause: After defining a subtype as NOT NULL, it was used as the base type for another subtype defined as NULL. That is not allowed. For example, the code might look like

```
DECLARE
   SUBTYPE Weekday IS INTEGER NOT NULL;
   SUBTYPE Weekend IS Weekday NULL;  -- illegal
```

instead of

```
DECLARE
   SUBTYPE Weekday IS INTEGER NOT NULL;
   SUBTYPE Weekend IS Weekday;
```

Action: Revise the subtype definitions to eliminate the conflict.

PLS-00367

PLS-00367   **a RAISE statement with no exception name must be inside anexception handler**

Cause: A RAISE statement not followed by an exception name was found outside an exception handler.

Action: Delete the RAISE statement, relocate it to an exception handler, or supply the missing exception name.

PLS-00368

PLS-00368   **in RAISE statement, *name* must be an exception name**

Cause: The identifier in a RAISE statement is not a valid exception name.

Action: Make sure the identifier in the RAISE statement was declared as an exception and is correctly placed in the block structure. If you are using the name of a PL/SQL predefined exception, check its spelling.

PLS-00369

PLS-00369   **no choices may appear with choice OTHERS in an exception handler**

Cause: A construct of the form

```
WHEN excep1 OR OTHERS =>
```

was encountered in the definition of an exception handler. The OTHERS handler must appear by itself as the last exception handler in a block.

Action: Remove the identifier that appears with OTHERS or write a separate exception handler for that identifier.

PLS-00370

PLS-00370   **OTHERS handler must be last among the exception handlers of a block**

Cause: One or more exception handlers appear after an OTHERS handler. However, the OTHERS handler must be the last handler in a block or subprogram because it acts as the handler for all exceptions not named specifically.

Action: Move the OTHERS handler so that it follows all specific exception handlers.

PLS-00371

PLS-00371   **at most one declaration for *name* is permitted in the declaration section**

Cause: A reference to an identifier is ambiguous because there are conflicting declarations for it in the declarative part of a block, procedure, or function. At most one declaration of the identifier is permitted in a declarative part.

Action: Check the spelling of the identifier. If necessary, remove all but one declaration of the identifier.

PLS-00372

PLS-00372   **in a procedure, RETURN statement cannot contain an expression**

Cause: In a procedure, a RETURN statement contains an expression, which is not allowed. In functions, a RETURN statement *must* contain an expression because its value is assigned to the function identifier. However, in procedures, a RETURN statement simply lets you exit before the normal end of the procedure is reached.

Action: Remove the expression from the RETURN statement, or redefine the procedure as a function.

PLS-00373

PLS-00373   **EXIT label *name* must label a LOOP statement**

Cause: The label in an EXIT statement does not refer to a loop. An EXIT statement need not specify a label. However, if a label is specified (as in EXIT *my_label*), it must refer to a loop statement.

Action: Make sure the label name is spelled correctly and that it refers to aloop statement.

PLS-00374

PLS-00374   **illegal EXIT statement; it must appear inside the loop labeled** *name*

Cause: An EXIT statement need not specify a label. However, if a label is specified (as in EXIT *my_label*), the EXIT statement must be inside the loop designated bythat label.

Action: Make sure the label name is spelled correctly; if necessary, move the EXIT statement inside the loop to which the label refers.

PLS-00375

PLS-00375 **illegal GOTO statement; this GOTO cannot branch to label** *name*

Cause: The line and column numbers accompanying the error message refer to a GOTO that branches from outside a construct (a loop or exception handler, for example) that contains a sequence of statements to a label inside that sequence of statements. Such a branch is not allowed.

Action: Either move the GOTO statement inside the sequence of statements or move the labeled statement outside the sequence of statements.

PLS-00376

PLS-00376   **illegal EXIT statement; it must appear inside a loop**

Cause: An EXIT statement was found outside of a loop construct. The EXIT statement is used to exit prematurely from a loop and so must always appear withina loop.

Action: Either remove the EXIT statement or place it inside a loop.

PLS-00377

PLS-00377   **internal type PLS_INTEGER is not included in this release of PL/SQL**

Cause: In a declaration, the obsolete datatype PLS_INTEGER, which has been replaced by the datatype BINARY_INTEGER, was specified.

Action: Specify the datatype BINARY_INTEGER instead of PLS_INTEGER.

PLS-00378

PLS-00378   **invalid compilation unit for this release of PL/SQL**

Cause: A *compilation unit* is a file containing PL/SQL source code that is passed to the compiler. Only compilation units containing blocks, declarations, statements, and subprograms are allowed. This error occurs when some other language construct is passed to the compiler.

Action: Make sure the compilation unit contains only blocks, declarations, statements, and subprograms.

PLS-00379

PLS-00379   **CASE statements are not included in this release of PL/SQL**

Cause: The unit being compiled contains a CASE statement. However, the current release of PL/SQL does not support CASE statements.

Action: Remove the CASE statement from the compilation unit.

PLS-00381

PLS-00381   **type mismatch found at *name* between column and variable in subqueryor INSERT**

Cause: The datatypes of a column and a variable do not match. The variable was encountered in a subquery or INSERT statement.

Action: Change the variable datatype to match that of the column.

PLS-00382

PLS-00382   **expression is of wrong type**

Cause: An expression has the wrong datatype for the context in which it was found.

Action: Change the datatype of the expression. You might want to use datatype conversion functions.

PLS-00383

PLS-00383 **type mismatch found at *name* inside an IN or NOT IN clause**

Cause: In a test for set membership such as X NOT IN (SELECT Y ...), the expressions X and Y do not match in datatype, and it is unclear which implicit conversion is required to correct the mismatch.

Action: Change the expressions so that their datatypes match. You might want to use datatype conversion functions in the select list.

PLS-00384

PLS-00384   **type mismatch found at** *name* **in UPDATE's SET clause**

Cause: The column to the left of the equal sign in the SET clause of an UPDATE statement does not match in datatype with the column, expression, or subquery to the right of the equal sign, and it is unclear which implicit conversion is required to correct the mismatch.

Action: Change the expressions so that their datatypes match. You might want to use datatype conversion functions in the SET clause.

PLS-00385

PLS-00385   **type mismatch found at *name* in SELECT...INTO statement**

Cause: The expressions to the left and right of the INTO clause in a SELECT...INTO statement do not match in datatype, and it is unclear which implicit conversion is required to correct the mismatch.

Action: Change the expressions so that their datatypes match. You might want to use datatype conversion functions in the select list.

PLS-00386

PLS-00386 **type mismatch found at** *name* **between FETCH cursor and INTO variables**

Cause: An assignment target in the INTO list of a FETCH statement does not match in datatype with the corresponding column in the select list of the cursor declaration, and it is unclear which implicit conversion is required to correctthe mismatch.

Action: Change the cursor declaration or change the datatype of the assignment target. You might want to use datatype conversion functions in the select list of the query associated with the cursor.

PLS-00387

PLS-00387   **INTO variable cannot be a database object**

Cause: An item in the INTO list of a FETCH or SELECT statement was found to be a database object. INTO introduces a list of user-defined variables to which output values are assigned. Therefore, database objects cannot appear in the INTO list.

Action: Check the spelling of the INTO list item. If necessary, remove the item from the INTO list or replace it with a user-defined output variable.

PLS-00388

PLS-00388 **undefined column *name* in subquery**

Cause: A subquery contains a column name that was not defined for thespecified table.

Action: Change the expression to specify a column that was defined.

PLS-00389

PLS-00389   **undefined column *name* in left-hand-side expression**

Cause: A left-hand-side expression in a SQL statement refers to an undefined column.

Action: Check the spelling of the column name, then change the expression so that it refers only to defined columns.

PLS-00390

PLS-00390 **undefined column** *name* **in INSERT statement**

Cause: An INSERT statement refers to a column not defined for the table or view into which data is being inserted.

Action: Check the spelling of the column name, then revise the statement so that it refers only to defined columns.

PLS-00391

PLS-00391   **undefined column *name* in UPDATE statement**

Cause: An UPDATE statement refers to a column not defined for the table or view being updated.

Action: Check the spelling of the column name, then revise the statement so that it refers only to defined columns.

PLS-00392

PLS-00392   **type mismatch in arguments to BETWEEN**

Cause: In a comparison such as X BETWEEN Y AND Z, the expressions X, Y, and Z do not match in datatype, and it is unclear which implicit conversion is required to correct the mismatch.

Action: Change the expressions so that their datatypes match. You might want to use datatype conversion functions.

PLS-00393

PLS-00393 **wrong number of columns in SELECT...INTO statement**

Cause: The number of columns selected by a SELECT...INTO statement does not match the number of variables in the INTO clause.

Action: Change the number of columns in the select list or the number of variables in the INTO clause so that the numbers match.

PLS-00394

PLS-00394   **wrong number of values in the INTO list of a FETCH statement**

Cause: The number of variables in the INTO clause of a FETCH statement does not match the number of columns in the cursor declaration.

Action: Change the number of variables in the INTO clause or the number of columns in the cursor declaration so that the numbers match.

PLS-00395

PLS-00395   **wrong number of values in VALUES clause of INSERT statement**

Cause: The number of columns in an INSERT statement does not match the number of values in the VALUES clause. For example, the following statement is faulty because no column is specified for the value 20:

```
INSERT INTO emp (empno, ename) VALUES (7788, 'SCOTT', 20);
```

Action: Change the number of items in the column list or the number of items in the VALUES list so that the numbers match.

PLS-00396

PLS-00396   **INSERT statement's subquery yields wrong number of columns**

Cause: The number of columns in an INSERT statement does not match the number of columns in a subquery select list. For example, the following statement is faulty because no corresponding column is specified for *col3*:

```
INSERT INTO emp (ename, empno) SELECT col1, col2, col3 FROM ...
```

Action: Change the number of items in the column list of the INSERT statement or the number of items in the select list so that the numbers match.

PLS-00397

PLS-00397   **type mismatch in arguments to IN**

Cause: In a test for set membership such as X IN (Y, Z), the expressions X, Y, and Z do not match in datatype, and it is unclear which implicit conversion is required to correct the mismatch.

Action: Change the expressions so that their datatypes match. You might want to use datatype conversion functions.

PLS-00398

PLS-00398 **wrong number of columns in UNION, INTERSECT, or MINUS expression**

Cause: The SELECT clauses to the left and right of a UNION, INTERSECT, or MINUS expression do not select the same number of columns. For example, the following statement is faulty because the select lists do not contain the same number of items:

```
CURSOR my_cur IS SELECT ename FROM emp
   INTERSECT SELECT ename, empno FROM emp;
```

Action: Change the select lists so that they contain the same number of items.

PLS-00399

PLS-00399   **different types of columns in UNION, INTERSECT, or MINUS expression**

Cause: The select lists to the left and right of a UNION, INTERSECT, or MINUS expression select at least one column that is mismatched in datatype. For example, the following statement is faulty because the constant 3 has datatype NUMBER, whereas SYSDATE has datatype DATE:

```
CURSOR my_cur IS SELECT 3 FROM emp
    INTERSECT SELECT SYSDATE FROM emp;
```

Action: Change the select lists so that they match in datatype. You might want to use datatype conversion functions in the select list of one or more queries.

PLS-00400

PLS-00400   **different number of columns between cursor SELECT statement and return value**

Cause: In a cursor declaration, a return type (such as RETURN *emp*%ROWTYPE) was specified, but the number of returned column values does not match the number of select-list items.

Action: Change the cursor return type or the select list so that the number of returned column values matches the number of select-list items.

PLS-00401

PLS-00401   **different column types between cursor SELECT statement and return value found at** *name*

Cause: In a cursor declaration, a return type (such as RETURN *emp*%ROWTYPE) was specified, but a returned column value and its corresponding select-list item have different datatypes.

Action: Change the cursor return type or the select list so that each returned column value and its corresponding select-list item have the same datatype.

PLS-00402

PLS-00402   **alias required in SELECT list of cursor to avoid duplicate column names**

Cause: A cursor was declared with a SELECT statement that contains duplicate column names. Such references are ambiguous.

Action: Replace the duplicate column name in the select list with an alias.

PLS-00403

PLS-00403   **INTO list of FETCH statement contains illegal assignment target**

Cause: A FETCH statement was unable to assign a value to an assignment target in its INTO list because the target is not a legally formed and declared variable. For example, the following assignment is illegal because 'Jones' is a character string, not a variable:

```
FETCH my_cur INTO 'Jones';
```

Action: Check the spelling and declaration of the assignment target. Make sure that the rules for forming variable names are followed.

PLS-00404

PLS-00404   **cursor *name* must be declared with FOR UPDATE to use withCURRENT OF**

Cause: The use of the CURRENT OF *cursor_name* clause is legal only if *cursor_name* was declared with a FOR UPDATE clause.

Action: Add a FOR UPDATE clause to the definition of the cursor or do not use the CURRENT OF *cursor_name* clause.

PLS-00405

PLS-00405 **subquery not allowed in this context**

Cause: A subquery was used in an inappropriate context, such as:

```
if (SELECT deptno FROM emp WHERE ... ) = 20 then ...
```

Subqueries are allowed only in SQL statements.

Action: The same result can be obtained by using a temporary variable, as in:

```
SELECT deptno INTO temp_var FROM emp WHERE ...;
IF temp_var = 20 THEN ...
```

PLS-00406

PLS-00406   **length of SELECT list in subquery must match number ofassignment targets**

Cause: A query select list is not the same length as the list of targets that will receive the returned values. For example, the following statement is faulty because the subquery returns two values for one target:

```
UPDATE emp SET ename =
   (SELECT ename, empno FROM emp WHERE ename = 'SMITH') ...
```

Action: Change one of the lists so that they contain the same number of items.

PLS-00407

PLS-00407   **"*" not allowed here; a list of columns is required**

Cause: An asterisk (*) was used as an abbreviation for a list of column names. However, in this context the column names must be written out explicitly.

Action: Replace the asterisk with a list of column names.

PLS-00408

PLS-00408   **duplicate column *name* not permitted in INSERT or UPDATE**

Cause: An UPDATE or INSERT statement has a column list that contains duplicate column names.

Action: Check the spelling of the column names, then eliminate the duplication.

PLS-00409

PLS-00409 **duplicate variable *name* in INTO list is not permitted**

Cause: The same variable appears twice in the INTO list of a SELECT orFETCH statement.

Action: Remove one of the variables from the INTO list.

PLS-00410

PLS-00410   **duplicate fields in record or table are not allowed**

Cause: When declaring a user-defined record, the same name was given to two fields. Like column names in a database table, field names in a user-defined record must be unique.

Action: Check the spelling of the field names, then remove the duplicate.

PLS-00411

PLS-00411   **number of values in aggregate and in subquery don't match**

Cause: In a statement of the form *aggregate = subquery*, the numbers of values in the aggregate and subquery are unequal. For example, the code might look like

```
... WHERE (10,20,30) = (SELECT empno,deptno FROM emp WHERE...);
```

Action: Revise the aggregate or subquery so that the numbers of values match.

PLS-00412

PLS-00412 **list of values not allowed as argument to this function or procedure**

Cause: A parenthesized list of values separated by commas (that is, an aggregate) was used in the wrong context. For example, the following usage is *invalid*:

```
WHERE (col1, col2) > (SELECT col3, col4 FROM my_table ...)
```

However, an equal sign can take a list of values and a subquery as left- and right-hand-side arguments, respectively. So, the following usage is valid:

```
WHERE (col1, col2) = (SELECT col3, col4 FROM my_table ...)
```

Action: Rewrite the expression. For example, the clause

```
WHERE (col1, col2) > (SELECT col3, col4 FROM my_table ...)
```

can be rewritten as

```
WHERE col1 > (SELECT col3 FROM my_table ...) AND
   col2 > (SELECT col4 FROM my_table ...)
```

PLS-00413

PLS-00413   **identifier in CURRENT OF clause is not a cursor name**

Cause: The identifier in a CURRENT OF clause names an object other than a cursor.

Action: Check the spelling of the identifier. Make sure that it names the cursor in the DELETE or UPDATE statement and that it names the cursor itself, not a FOR-loop variable.

PLS-00414

PLS-00414   **no column *name* in table**

Cause: A table name or alias was used to qualify a column reference, but the column was not found in that table. Either the column was never defined or the column name is misspelled.

Action: Confirm that the column was defined and check the spelling of thecolumn name.

PLS-00417

PLS-00417	table or view or column does not exist

Cause: A database table, view, or column was specified in a SQL statement that does not exist, or the privileges required to access the table or view were not granted.

Action: Check the spelling of the table (or view) and column names; make sure the table and columns exist. If necessary, ask the DBA to grant the privileges required to access the table.

PLS-00418

PLS-00418   **array bind type must match PL/SQL table row type**

Cause: A host array was passed (by an Oracle Precompiler program, for example) to a PL/SQL subprogram for binding to a PL/SQL table parameter. However, the datatypes of the array elements and PL/SQL table rows are incompatible. So, the binding failed.

Action: Change the datatype of the array elements or PL/SQL table rows to make the datatypes compatible.

Cause: An attempt was made to reference a remote cursor attribute, which is not allowed. For example, the code might look like

```
IF SQL%NOTFOUND@newyork THEN ...
```

Action: Do not try to reference a remote cursor attribute.

PLS-00419

PLS-00419   **reference to remote attribute not permitted**

Cause: An attempt was made to reference a remote cursor attribute, which is not allowed. For example, the code might look like

```
IF SQL%NOTFOUND@newyork THEN ...
```

Action: Do not try to reference a remote cursor attribute.

PLS-00420

PLS-00420   **cannot call built-in routines remotely**

Cause: An attempt was made to call a built-in PL/SQL function remotely, which is not allowed. For example, the code might look like

```
my_sqlerrm := SQLERRM@newyork;
```

or

```
INSERT INTO emp VALUES (my_empno, STANDARD.RTRIM@newyork(my_ename), ...);
```

Action: Always call built-in functions locally; never specify a database link.

PLS-00421

PLS-00421   **synonym definitions nested too deeply; possible loop in synonyms**

Cause: Directly or indirectly, a synonym was defined in terms of itself, creating a circular definition. Or, a chain of synonyms, too long for the PL/SQL compiler to handle, was defined.

Action: Redefine the synonyms to eliminate the circular definition. If necessary, shorten the chain of synonyms.

PLS-00422

PLS-00422   **no PL/SQL translation for the bind type given for this bind variable**

Cause: A host variable was passed (by an Oracle Precompiler program, for example) to PL/SQL for binding. However, its datatype is not compatible with any PL/SQL datatype. So, the binding failed.

Action: Change the datatype of the host variable to make it compatible with aPL/SQL datatype.

PLS-00424

PLS-00424   **all arguments must be provided in remote procedure call; no defaults**

Cause: An attempt was made to call a remote subprogram using defaulted parameters, which is not allowed. When calling remote subprograms, the actual parameters must be passed explicitly, even if the corresponding formal parametershave defaults.

Action: Call the remote subprogram by passing each actual parameter explicitly.

PLS-00425

PLS-00425   **in SQL statement, argument to function must have SQL type**

Cause: When a stored function was called from a SQL statement, parameters of the wrong type were passed. To be callable from SQL statements, a stored function must meet several requirements, one of which is that its arguments have SQL datatypes such as CHAR, DATE, or NUMBER. None of the arguments can have non-SQL types such as BOOLEAN, TABLE, or RECORD.

Action: Make sure all the arguments in the function call have SQL datatypes.

PLS-00428

PLS-00428   **an INTO clause is expected in this SELECT statement**

Cause: The INTO clause of a SELECT INTO statement was omitted. For example, the code might look like

`SELECT deptno, dname, loc FROM dept WHERE ...`

instead of

`SELECT deptno, dname, loc INTO dept_rec FROM dept WHERE ...`

In PL/SQL, only a subquery is written without an INTO clause.

Action: Add the required INTO clause.

PLS-00450

PLS-00450   a variable of this private type cannot be declared here

Not in Release 2.2.

PLS-00451

PLS-00451   **remote types not allowed**

Cause: When a constant or variable was declared, a datatype that is defined in a remote library unit was specified. This is not allowed.

Action: Do not specify the datatype unless it is defined locally.

PLS-00452

PLS-00452   **subprogram *name* violates its associated pragma**

Cause: A packaged function cannot be called from SQL statements unless its purity level is asserted by coding a RESTRICT_REFERENCES pragma in the package specification. The pragma, which is used to control side effects, tells the PL/SQL compiler to deny the packaged function read/write access to database tables, public packaged variables, or both. A SQL statement that violates the pragma will cause a compilation error.

Action: Raise the purity level of the function, or relax the pragma restrictions.

PLS-00483

PLS-00483 **exception *name* may appear in at most one exception handler in this block**

Cause: An exception appears in two different WHEN clauses (that is, two different exception handlers) in the exception-handling part of a PL/SQL blockor subprogram.

Action: Remove one of the references to the exception.

PLS-00484

PLS-00484 **redundant exceptions *name* and *name* must appear in sameexception handler**

Cause: Using the EXCEPTION_INIT pragma, different exceptions were initialized to the same Oracle error number; then, they were referenced in different exception handlers within the same exception-handling part. Such references conflict.

Action: Remove one of the exceptions or initialize it to a different Oracle error number.

PLS-00485

PLS-00485   **in exception handler, *name* must be an exception name**

Cause: An identifier not declared as an exception appears in an exception handler WHEN clause. Only the name of an exception is valid in a WHEN clause.

Action: Check the spelling of the exception name and make sure the exception was declared properly.

PLS-00486

PLS-00486   **SELECT list cannot be enclosed in parentheses**

Cause: In a SELECT statement, the select list was enclosed in parentheses, as in:

```
SELECT (deptno, dname, loc) FROM dept INTO ...
```

This breaks the rules of SQL syntax. Parentheses are not required because the keywords SELECT and FROM delimit the select list.

Action: Remove the parentheses enclosing the select list.

PLS-00487

PLS-00487   **invalid reference to variable *name***

Cause: A variable was referenced in a way that is inconsistent with its datatype. For example, a scalar variable might have been mistakenly referenced as a record, as follows:

```
DECLARE
   CURSOR emp_cur IS SELECT empno, ename, sal FROM emp;
   emp_rec emp_cur%ROWTYPE;
   my_sal  NUMBER(7,2);
BEGIN    ...    total_sal := total_sal + my_sal.sal;  -- invalid
   ...
```

Action: Check the spelling of the variable name. Make sure the variable was declared properly and that the declaration and reference are consistentregarding datatype.

PLS-00488

PLS-00488   **invalid variable declaration: object *name* must be a type or subtype**

Cause: The datatype specifier in a variable declaration does not designate a legal type. For example, the %TYPE attribute might not have been added to a declaration, as in

```
DECLARE
   my_sal    emp.sal%TYPE;
   my_ename emp.ename;      -- missing %TYPE
   ...
```

When declaring a constant or variable, to provide the datatype of a column automatically, use the %TYPE attribute. Likewise, when declaring a record, to provide the datatypes of a row automatically, use the %ROWTYPE attribute.

Action: Make sure the datatype specifier designates a legal type. Remember to use the %TYPE and %ROWTYPE attributes when necessary.

PLS-00489

PLS-00489   **invalid table reference: *name* must be a column in this expression**

Cause: In a query, a select-list item refers to a table in the FROM clause but not to a database column.

Action: Check the spelling of the column names, make sure each column in the select list refers to a table in the FROM clause, then re-execute the query.

PLS-00490

PLS-00490   **illegal statement**

Cause: A constant, variable, function call, or incomplete statement was used where a statement was expected. For example, instead of calling a function from an expression, it might have been called as a statement (as if it were a procedure).

Action: Check the statement, making sure that its commands, identifiers, operators, delimiters, and terminator form a complete and valid PL/SQL statement.

PLS-00491

PLS-00491   **numeric literal required**

Cause: A constant or variable was used where a numeric literal is required. For example, the code might look like

```
my_ename   VARCHAR2(max_len);
```

instead of

```
my_ename   VARCHAR2(15);
```

When specifying the maximum length of a VARCHAR2 variable, an integer literal must be used.

Action: Replace the identifier with a numeric literal.

PLS-00503

PLS-00503   **RETURN <value> statement required for this return from function**

Cause: In a function body, a RETURN statement was used that contains no expression. In procedures, a RETURN statement contains no expression because the statement simply returns control to the caller. However, in functions, a RETURN statement must contain an expression because its value is assigned to the function identifier.

Action: Add an expression to the RETURN statement.

PLS-00504

PLS-00504   **type *name*_BASE may not be used outside of package STANDARD**

Cause: In a declaration, the datatype NUMBER_BASE (for example) was mistakenly specified. The datatypes CHAR_BASE, DATE_BASE, MLSLABEL_BASE, and NUMBER_BASE are for internal use only.

Action: Specify (for example) the datatype NUMBER instead of NUMBER_BASE.

PLS-00506

PLS-00506   **user-defined constrained subtypes are disallowed**

Cause: An attempt was made to define a constrained subtype, but only unconstrained subtypes are allowed in this release of PL/SQL. For example, the following type definition is illegal:

```
SUBTYPE Acronym IS VARCHAR2(5);  -- illegal
```

Action: Remove the illegal type constraint.

PLS-00507

PLS-00507   **PL/SQL tables may not be defined in terms of records or other tables**

Cause: In a TABLE type definition, a composite datatype (RECORD or TABLE) was mistakenly specified for the column. The single, unnamed column must belong to a scalar datatype such as CHAR, DATE, or NUMBER.

Action: Remove the TABLE type definition, or replace the composite datatype specifier with a scalar datatype specifier.

PLS-00508

PLS-00508 **the expression in a RETURN statement cannot be a type**

Cause: A datatype specifier was used instead of an expression in the RETURN statement of a user-defined function, as shown in the example below. Do not confuse the RETURN statement, which sets the function identifier to the result value, with the RETURN clause, which specifies the datatype of theresult value.

```
FUNCTION credit-rating (acct_no NUMBER) RETURN BOOLEAN IS
   BEGIN
      ...
      RETURN NUMBER;  -- should be an expression
   END;
```

Action: Replace the datatype specifier in the RETURN statement with an appropriate expression.

PLS-00510

PLS-00510	**FLOAT cannot have scale**

Cause: When declaring a FLOAT variable, its precision and scale were specified, as shown in the following example:

```
DECLARE
    Salary FLOAT(7,2);
```

However, a scale for FLOAT variables cannot be specified; only a precision can be specified, as in

```
    salary FLOAT(7);
```

Action: Remove the scale specifier from the declaration, or declare a NUMBERvariable instead.

PLS-00700

PLS-00700   **PRAGMA EXCEPTION_INIT of *name* must follow declaration of its exception in same block**

Cause: An EXCEPTION_INIT pragma was not declared in the same block as its exception. They must be declared in the proper order in the same block, with the pragma declaration following the exception declaration.

Action: Place the EXCEPTION_INIT pragma directly after the exception declaration referenced by the pragma.

PLS-00701

PLS-00701 **illegal Oracle error number *num* for PRAGMA EXCEPTION_INIT**

Cause: The error number passed to an EXCEPTION_INIT pragma was out of range. The error number must be in the range -9999 .. -1 (excluding -100) for Oracle errors or in the range -20000 .. -20999 for user-defined errors.

Action: Use a valid error number.

PLS-00702

PLS-00702   **second argument to PRAGMA EXCEPTION_INIT must be a numeric literal**

Cause: The second argument passed to an EXCEPTION_INIT pragma was something other than a numeric literal (a variable, for example). The second argument must be a numeric literal in the range -9999 .. -1 (excluding -100) for Oracle errors or in the range -20000 .. -20999 for user-defined errors.

Action: Replace the second argument with a valid error number.

PLS-00703

PLS-00703   **multiple instances of named argument in list**

Cause: Two or more actual parameters in a subprogram call refer to the sameformal parameter.

Action: Remove the duplicate actual parameter.

PLS-00704

PLS-00704 *name* **must be declared as an exception**

Cause: The *exception_name* parameter passed to an EXCEPTION_INIT pragma is misspelled or does not refer to a legally declared exception. Or, the pragma is misplaced; it must appear in the same declarative section, somewhere after the exception declaration.

Action: Check the spelling of the *exception_name* parameter. Then, check the exception declaration, making sure the exception name and the keyword EXCEPTION are spelled correctly. Also make sure the pragma appears in the same declarative section somewhere after the exception declaration.

PLS-00705

PLS-00705   **exceptions not allowed in an expression**

Cause: An incorrect reference to an exception within an expression was made. Exceptions have names but not values and therefore cannot contribute values to an expression. For example, the following RETURN statement is illegal:

```
FUNCTION credit_limit (cust_no INTEGER) RETURN NUMBER IS
   limit        NUMBER;
   over_limit  EXCEPTION;
   ...
BEGIN
   ...
   RETURN over_limit;  -- illegal
END;
```

Action: Check the spelling of the identifiers in the expression, then rewrite the expression so that it does not refer to an exception.

PLS-00706

PLS-00706 **exception *name* cannot be used as prefix of a selected component**

Cause: An exception name was mistakenly used to qualify a reference to a component. For example, when dot notation was used to specify fields within a record, an exception name might have been coded instead of the record name.

Action: Rewrite the component reference using a valid prefix (for example, the name of a package, record, or schema).

PLS-00900

PLS-00900 **cannot find body of unit** *name*

Cause: At run time, the body of a program unit could not be found. This might have happened, for example, if a procedure that has a specification but no body was referenced. (No compile-time errors were generated because the specification exists.)

Action: Define a body for the program unit.

PLS-00901

PLS-00901   **the datatype of column *name* of table *name* is not supported**

Cause: A column in a database table belongs to a datatype that is not supported by the current release of PL/SQL.

Action: Remove the offending column from the table or copy the desired columns to another table.

PLS-00902

PLS-00902   **a read-only bind variable used in OUT or IN OUT context**

Cause: A host variable that is protected from update was used in a context that allows an update.

Action: Check the context and change the use of the host variable, or assign the value of the host variable to a PL/SQL local variable, then use the localvariable instead.

PLS-00904

PLS-00904   **insufficient privilege to access object** *name*

Cause: An attempt was made to operate on a database object without the required privilege. This error occurs, for example, if an attempt was made to UPDATE a table for which only SELECT privileges were granted.

Action: Ask the DBA to perform the operation or to grant you the required privilege.

PLS-00905

PLS-00905   **object *name* is invalid**

Cause: An invalid package specification or stored subprogram was referenced. A package specification or stored subprogram is invalid if its source code or any database object it references has been DROPped, REPLACEd, or ALTERed since it was last compiled.

Action: Find out what invalidated the package specification or stored subprogram, then make sure that Oracle can recompile it without errors.

PLS-00906

PLS-00906 **compilation is not possible**

Cause: The PL/SQL compiler cannot run properly because its operating environment is corrupted. For example, its error message file might be inaccessible.

Action: Check the PL/SQL operating environment, making sure that all files required by the compiler are accessible.

PLS-00990

PLS-00990   **PL/SQL tables of cursor variables are not allowed**

Cause: An attempt was made to use a REF CURSOR type to define a TABLE type, which is not allowed. The rows in a PL/SQL table cannot store the values of cursor variables.

Action: Redefine the TABLE type using one of the PL/SQL scalar datatypes (such as CHAR, DATE, or NUMBER) to specify its column type.

PLS-00992

PLS-00992 **cursor variables cannot be FETCHed from**

Cause: An attempt was made to FETCH from a cursor variable inside a PL/SQL block or subprogram, which is not allowed. Although a cursor variable is OPENed on the server side by passing it to an anonymous PL/SQL block, it must be fetched from on the client side (that is, in the host environment).

Action: Move the FETCH statement from the PL/SQL block or subprogram to thehost environment.

PLS-00993

PLS-00993   **cursor variables cannot be passed as RPC arguments or results**

Cause: An attempt was made to pass a cursor variable to or from a remote subprogram, which is not allowed. For example, a remote procedure cannot be used to open a cursor variable because remote subprograms cannot return the values of cursor variables.

Action: Change the subprogram call to reference a local database.

PLS-00994

PLS-00994   **cursor variables cannot be declared as part of a package**

Cause: An attempt was made to declare a cursor variable in a package specification, which is not allowed. Although REF CURSOR types can be defined in a PL/SQL block, subprogram, or package, cursor variables can be declared only in a block or subprogram.

Action: Move the cursor variable declaration into a PL/SQL block or subprogram.

PLS-00995

PLS-00995 **unhandled exception #** *num*

Cause: An exception was raised for which no handler was found. If it cannot find a handler for a raised exception, PL/SQL returns an unhandled exception to the host environment. The number embedded in the message is an Oracle error code listed in this manual.

Action: Fix the condition that raised the exception, write an appropriate exception handler, or use the OTHERS handler. If there is an appropriate handler in the current block, the exception was raised in a declaration or exception handler, and therefore propagated immediately to the enclosing block.

PLS-00996

PLS-00996   **out of memory**

Cause: A request from PL/SQL for more memory failed.

Action: Make sure that you are not referencing the wrong row in a PL/SQL table and that the program is not recursing too deeply.

01400-01799: FIPS Warnings

This section lists the warnings that can be generated when the FIPS flagger is set in an Oracle product and an Oracle extension to the SQL92 standard is used or a SQL92 feature is used in a non-conforming manner. FIPS warnings messages are listed for completeness. The cause and actions are described in the ANSI document ANSI X3.135-1992. The Oracle products that can generate FIPS warning messages are SQL*DBA, Server Manager, SQL*Plus, Oracle precompilers, and so on.

PLS-01400

PLS-01400   use of ||' token

PLS-01401

PLS-01401   **identifier over 18 characters long**

PLS-01402

PLS-01402   use of quoted identifier

PLS-01403

PLS-01403   Comment delimiters /* */

PLS-01404

PLS-01404   ANSI Identifiers can only consist of letters, digits, and the underscore character

PLS-01405

PLS-01405   under ANSI's grammar, numeric data represented in exponent notation must use an uppercase E

PLS-01406

PLS-01406   an identifier has been found that is considered a keyword in ANSI's grammar but not in PL/SQL's

PLS-01407

PLS-01407   **one of the identifiers CHAR, CHARACTER, or INTEGER has been redefined by the programmer**

PLS-01408

PLS-01408   use of & token

PLS-01409

PLS-01409   use of PRIOR_ token

PLS-01410

PLS-01410   use of MOD token

PLS-01411

PLS-01411   use of REM token

PLS-01412

PLS-01412   use of EXP token

PLS-01413

PLS-01413   use of NULL an expression

PLS-01414

PLS-01414   use of qualified expression here

PLS-01415

PLS-01415   use of aggregate value here

PLS-01416

PLS-01416   use of <id> (<value>...) here

PLS-01417

PLS-01417   use of %attribute

PLS-01418

PLS-01418   **subquery cannot include set operators in ANSI's grammar**

PLS-01419

PLS-01419   subquery must have either *' or exactly one column in its select list

PLS-01420

PLS-01420   INTERSECT and MINUS set operators are not ANSI

PLS-01421

PLS-01421   FOR UPDATE clause

PLS-01422

PLS-01422   aliases

PLS-01423

PLS-01423   **subquery on right-hand-side in set clause**

PLS-01424

PLS-01424   non-ANSI order of clauses

PLS-01425

PLS-01425   connect-by clause

PLS-01450

PLS-01450   **this <value expression> contains a nonconforming datatype**

PLS-01451

PLS-01451   the datatypes of these <value expressions> must be comparable

PLS-01452

PLS-01452   this function is not part of the ANSI standard

PLS-01453

PLS-01453   this procedure is not part of the ANSI standard

PLS-01454

PLS-01454   no operator may be used with values of datatype CHAR

PLS-01455

PLS-01455   the predicates IS NULL and IS NOT NULL are defined only forcolumn specifications

PLS-01456

PLS-01456   name length cannot exceed 3.

PLS-01458

PLS-01458   the set clause in an UPDATE statement requires a column name of length 1

PLS-01460

PLS-01460   cannot access object on a remote host

PLS-01463

PLS-01463   STDDEV is not a standard set function

PLS-01464

PLS-01464   VARIANCE is not a standard set function

PLS-01466

PLS-01466   null strings are not allowed

PLS-01467

PLS-01467   the value list of the IN predicate, if not a subquery, must contain only value_specifications

PLS-01468

PLS-01468   **the first argument of the like-predicate must be a column of typecharacter string**

PLS-01469

PLS-01469   ANSI standard does not permit records

PLS-01470

PLS-01470   the escape character in the like-predicate must be a literal or a variable of type CHAR

PLS-01471

PLS-01471 the keyword DISTINCT must be present in a COUNT(DISTINCT sim_expr)

PLS-01472

PLS-01472   in a set_function_specification, if DISTINCT is present, the expression must be a column_specification

PLS-01473

PLS-01473   use <> instead of != or ~=

PLS-01474

PLS-01474   use of ANY is non_ANSI

PLS-01500

PLS-01500   unions are not allowed in the definition of a view

PLS-01501

PLS-01501   **options in CREATE TABLE statement is non-ANSI.**

PLS-01502

PLS-01502   constraint name and constraint status are non-ANSI.

PLS-01503

PLS-01503   the expression in a SET clause must not include a set function

PLS-01504

PLS-01504   use of AS phase in CREATE TABLE statement is non_ANSI

PLS-01505

PLS-01505   ALTER is non-ANSI

PLS-01506

PLS-01506   CREATE INDEX is non-ANSI

PLS-01507

PLS-01507   DROP is non-ANSI

PLS-01701

PLS-01701   illegal syntax in ROLLBACK WORK statement

PLS-01702

PLS-01702   illegal syntax in COMMIT WORK statement

PLS-01703

PLS-01703   cursor name in CLOSE statement must be of length 1

PLS-01704

PLS-01704   the expressions in a sort clause must be column specs or unsigned integers, with optional ASC or DESC

PLS-01705

PLS-01705   table specified by a cursor not updatable if cursor specification has a UNION or ORDER_BY

PLS-01707

PLS-01707   in positioned DELETE or UPDATE statement, table must be identified in specification of cursor

PLS-01708

PLS-01708   **in searched DELETE, UPDATE or INSERT, table must not appear in FROM clause of subqueries in search condition**

PLS-01709

PLS-01709   value list of INSERT statement, if specified with value list rather than subquery, must be value specification

PLS-01710

PLS-01710   in a positioned DELETE or UPDATE statement, the cursor name must be of length 1

PLS-01711

PLS-01711   a SELECT statement may not contain ORDER_BY, HAVING, or GROUP_BY clause

PLS-01712

PLS-01712   ANSI does not allow bind variables as INDICATORS

PLS-01713

PLS-01713   the constraints on these types do not match

PLS-01714

PLS-01714   ANSI expects a column name not a literal value here

PLS-01715

PLS-01715   ANSI does not accept SAVEPOINTs

Precompiler Messages

These messages might occur when running the Oracle Precompilers. Also listed are messages that the Oracle runtime library might return to the SQLCA. The Oracle Precompilers issue various warning and error messages. For example, the following message code and message statement might appear:

PCC-U-00036: No input file name specified

The message code consists of a prefix, severity code, and sequence number. The prefix PCC shows that the message was generated by an Oracle Precompiler. There are four severity codes. The following table gives their meanings:

Code	Meaning
Warning	despite an error, a compilable output file was created.
Severe error	despite an error, an output file was created. However, it might not be compilable.
Fatal error	no output file was created because of an internal problem or because a resource (such as memory) was unavailable or ran out.
Unrecoverable error	no output file was created because an input requirement was not met.
Information only	no error or warning was encountered. Information messages are used only for FIPS flagging.

00001-01600: Oracle Precompiler, Version 1 Messages

PCC-00001

PCC-00001   **Unable to open file name**

Cause: The precompiler was unable to open a temporary file for internal use. There might be insufficient disk space, too many open files, or read-only protection on the output directory.

Action: Check that there is enough disk space, that the limit for open files is set high enough (check with the system manager) and that protection on the directory allows opening a file for writing.

PCC-00002

PCC-00002	**Invalid syntax at column name in line num of file name**

Cause: There is a syntax error in an EXEC statement or the statement is notproperly terminated.

Action: Correct the syntax of the EXEC statement. If the error occurred at the end of the input file, check that the last EXEC statement is properly terminated.

PCC-00003

PCC-00003   **Invalid SQL Identifier at column name in line num of file name**

Cause: The symbol in a conditional precompilation statement (such as EXEC ORACLE IFDEF) is invalid, or the name of a SQL descriptor, statement, or cursor is invalid or was not properly declared.

Action: Check the statement syntax and spelling of the identifier and check that a reserved word was not accidentally used. If necessary, define the identifier in a variable declaration or DECLARE statement ahead of the line in error.

PCC-00004

PCC-00004   **Mismatched IF/ELSE/ENDIF block at line num in file name**

Cause: There is an EXEC ORACLE ELSE or EXEC ORACLE ENDIF statement without a matching EXEC ORACLE IFDEF statement.

Action: Add the missing EXEC ORACLE IFDEF statement or delete or move the EXEC ORACLE ELSE or EXEC ORACLE ENDIF statement.

PCC-00005

PCC-00005   **Unsupported datatype in line num of file name**

Cause: A host variable defined in the Declare Section has an unsupported datatype or has a scale or precision outside the supported range.

Action: Redefine the host variable using a supported datatype. Check that the scale and precision of a numeric variable are in the accepted range.

PCC-00007

PCC-00007   **Invalid WHENEVER condition at column name in line num of file name**

Cause: A condition other than SQLERROR, SQLWARNING, or NOT FOUND was specified in an EXEC SQL WHENEVER statement, or one of these was used but spelled incorrectly.

Action: Correct the spelling of the WHENEVER condition or use a host-language IF statement to test the special condition.

PCC-00008

PCC-00008   Invalid WHENEVER action at column name in line num of file name

Cause: At least one of the following:
- An action other than CONTINUE, GOTO, or STOP was specified in an EXEC SQL WHENEVER statement.
- One of the specified actions was spelled incorrectly.
- The host language does not allow the action (STOP is illegal in Pro*Pascal programs).
- A GOTO label is invalid.

Action: Check that the host language allows the specified WHENEVER action. If necessary, correct the spelling of the WHENEVER action or correct theGOTO label.

PCC-00009

PCC-00009   **Invalid host variable at column name in line num of file name**

Cause: A host variable used in an EXEC SQL statement was not declared in the Declare Section or has an unsupported datatype.

Action: Declare the host variable in the Declare Section, making sure it has one of the supported datatypes.

PCC-00010

PCC-00010   **Statement out of place at line num in file name**

Cause: An EXEC statement was not placed properly in the host program. For example, there might be a data manipulation statement in the Declare Section. In a Pro*COBOL program, the Declare Section might be outside the WORKING-STORAGE SECTION.

Action: Remove or relocate the statement.

PCC-00011

PCC-00011 **Already in a Declare Section at line num in file name**

Cause: A BEGIN DECLARE SECTION statement was found inside a Declare Section.

Action: Remove the extra BEGIN DECLARE SECTION statement.

PCC-00012

PCC-00012   **Not in a Declare Section at line num in file name**

Cause: An END DECLARE SECTION statement without a matching BEGIN DECLARE SECTION statement was found. Either the BEGIN DECLARE SECTION statement is missing or misspelled or the END DECLARE SECTION statement is an extra.

Action: Add or correct the BEGIN DECLARE SECTION statement or remove the extra END DECLARE SECTION statement.

PCC-00013

PCC-00013   **Unable to open INCLUDE file name at line num in file name**

Cause: The precompiler was unable to open the input file specified in the INCLUDE statement. Some possible causes follow:

- The filename is misspelled.
- The file does not exist.
- The search path to the file is incorrect.
- File access privileges are insufficient.
- Another user has locked the file.
- There is not enough disk space.
- There are too many open files.

Action: Check that the file exists, that the search path to the file is correct, that sufficient privileges to access the file have been granted, and that it is not locked by another user. Also, check that there is enough disk space and that the limit for open files is set high enough (check with the system manager).

PCC-00014

PCC-00014   **Undeclared SQL identifier ident at line num in file name**

Cause: The name of a descriptor, statement, or cursor was not declared oris misspelled.

Action: Add or correct the descriptor, statement, or cursor declaration.

PCC-00015

PCC-00015   **Unrecognized host language syntax ignored at line num in file name**

Cause: The host language syntax used to define a host variable in the Declare Sectionis incorrect.

Action: Check the syntax and the spelling, then correct the declaration.

PCC-00016

PCC-00016   **Unable to open a cursor at line num in file name**

Cause: The syntax in a SQL statement is faulty. The precompiler was expecting a host variable but found something else.

Action: Check the syntax and the spelling, then correct the SQL statement.

PCC-00017

PCC-00017   **Unable to parse statement at line num in file name**

Cause: There is a syntax error in an array declaration. The precompiler was expecting a right bracket (]) but found something else.

Action: Check the syntax, then correct the array declaration.

PCC-00018

PCC-00018   **Expected token, but found token at line num in file name**

Cause: The syntax in a SQL statement is faulty. The precompiler found an unexpected or illegal token.

Action: Check the syntax and the spelling, then correct the SQL statement.

PCC-00019

PCC-00019   **Unable to obtain bind variables at line num in file name**

Cause: The precompiler was unable to find information about an input host variable (bind variable) used in a SQL statement.

Action: Check that the input host variable is declared in the Declare Section and used properly in the SQL statement.

PCC-00020

PCC-00020   **Unable to obtain define variables at line num in file name**

Cause: The precompiler was unable to find information about an output host variable (define variable) used in a SQL statement.

Action: Check that the output host variable is declared in the Declare Section and used properly in the SQL statement.

PCC-00021

PCC-00021   Oracle Error: ORA-nnnnn

Cause: An Oracle error occurred.

Action: Refer to the indicated message in Chapter 2msgchpt2 of this manual.

PCC-00022

PCC-00022   **Out of space - unable to allocate num bytes**

Cause: The precompiler process ran out of memory.

Action: Allocate more memory to the process, then retry.

PCC-00023

PCC-00023   **Unable to log off Oracle**

Cause: An Oracle connection error occurred while the precompiler was trying to log off, probably because Oracle has been shut down.

Action: Check that Oracle is available, then retry.

PCC-00024

PCC-00024   **Indicator variable var has wrong type or length at line num in file name**

Cause: An indicator variable was not declared in the Declare Section as a 2-byte integer. Indicator variables must be defined as 2-byte integers.

Action: Redefine the indicator variable as a 2-byte integer.

PCC-00025

PCC-00025   **Undeclared indicator variable var at line num in file name**

Cause: An indicator variable used in a SQL statement was not declared in the Declare Section or its name is misspelled.

Action: Add or correct the indicator variable declaration.

PCC-00026

PCC-00026   Invalid host variable vat at line num in file name

Cause: A host variable used in a SQL statement was not declared properly. Some possible causes follow:
- It was not declared in the Declare Section.
- Its datatype is not supported.
- Its name is misspelled.
- It is a COBOL group item; in Pro*COBOL, only elementary items are allowed in a SQL statement.

Action: Add or correct the host variable declaration.

PCC-00027

PCC-00027	**Redeclared SQL identifier ident at line num in file name**

Cause: The name of a SQL descriptor, statement, or cursor was re-declared (that is, declared twice).

Action: Check the spelling of the identifier, then, if necessary, remove theextra declaration.

PCC-00028

PCC-00028   **Option optnam not legal as EXEC ORACLE OPTION**

Cause: A precompiler option was specified inline in an EXEC ORACLE statement, instead of on the command line. Some options can be specified only on the command line. For example, INAME cannot be specified inline.

Action: Respecify the precompiler option on the command line, instead of in an EXEC ORACLE statement. To see an online display of the precompiler options, enter the precompiler command (with no options) at the operating-system prompt.

PCC-00029

PCC-00029   **Ambiguous option optnam**

Cause: The name of a precompiler option was abbreviated ambiguously. For example, MAX= might refer to MAXLITERAL or MAXOPENCURSORS.

Action: Respecify the full option name or an unambiguous abbreviation. To see an online display of the precompiler options, enter the precompiler command (with no options) at the operating-system prompt.

PCC-00031

PCC-00031	Invalid value given for option optnam

Cause: The value specified for a precompiler option is invalid, probably because the value is misspelled (as in LTYPE=HORT) or out of range (as in PAGELEN=-55).

Action: Check the value, making sure it is spelled correctly and within the legal range.

PCC-00032

PCC-00032   Invalid option optnam

Cause: The precompiler found an invalid precompiler option name. Some possible causes follow:

- The option name is misspelled.
- The specified option does not exist.
- The equal sign (=) between the option name and value is missing or has space around it.

Action: Check that the option exists and that its name is spelled correctly. To see an online display of the precompiler options, enter the precompiler command (with no options) at the operating-system prompt. Also check that there is an equal sign between the option name and value.

PCC-00033

PCC-00033   **Missing operand for option optnam**

Cause: No value was specified for a precompiler option. Either the value is missing or there is space around the equal sign (as in LTYPE=SHORT).

Action: Check that a value for each option has been specified and that there is no space around the equal sign.

PCC-00036

PCC-00036   **No input file name specified**

Cause: The input file was not specified on the command line.

Action: Use the INAME command-line option to specify the input file.

PCC-00037

PCC-00037   **Unable to log on to Oracle with username. Oracle error number: ora-nnnnn**

Cause: The precompiler was unable to log on to Oracle with the specified username and password. An Oracle error with given number occurred when the logon was attempted.

Action: Refer to the indicated message in Chapter 2msgchpt2. of this manual.

PCC-00038

PCC-00038   **Unable to open a cursor**

Cause: This is an internal error message not usually issued.

Action: Call customer support for assistance. If the application does not require syntactic or semantic checking of SQL statements and does not use PL/SQL, specify SQLCHECK=NONE on the command line.

PCC-00039

PCC-00039   Unable to open input file name

Cause: The precompiler was unable to open the input file specified by the INAME precompiler option. Some possible causes follow:

- The filename is misspelled.
- The file does not exist.
- The search path to the file is incorrect.
- File access privileges are insufficient.
- Another user has locked the file.
- There is not enough disk space.
- There are too many open files.

Action: Check that the file exists, that the search path to the file is correct, that sufficient privileges have been granted to access the file, and that it is not locked by another user. Also check that there is enough disk space and that the limit for open files is set high enough (check with the system manager).

PCC-00040

PCC-00040   **Unable to open listing file name**

Cause: The precompiler was unable to open the listing file specified by the LNAME precompiler option. Some possible causes follow:

- The filename is misspelled.
- The file does not exist.
- The search path to the file is incorrect.
- File access privileges are insufficient.
- Another user has locked the file.
- There is not enough disk space.
- There are too many open files.

Action: Check that the file exists, that the search path to the file is correct, that sufficient privileges have been granted to access the file, and that it is not locked by another user. Also check that there is enough disk space and that the limit for open files is set high enough (check with the system manager). If a listing file is not needed, specify LTYPE=NONE on the command line.

PCC-00041

PCC-00041   Unable to open output file name

Cause: The precompiler was unable to open the output file specified by the ONAME precompiler option. Some possible causes follow:

- The filename is misspelled.
- The file does not exist.
- The search path to the file is incorrect.
- File access privileges are insufficient.
- Another user has locked the file.
- There is not enough disk space.
- There are too many open files.

Action: Check that the file exists, that the search path to the file is correct, that sufficient privileges have been granted to access the file, and that it is not locked by another user. Also check that there is enough disk space and that the limit for open files is set high enough (check with the system manager).

PCC-00042

PCC-00042 **Must include SQLCA file when MODE=ANSI and WHENEVER SQLWARNING used**

Cause: When MODE={ANSI|ANSI14}, an attempt was made to use the WHENEVER SQLWARNING statement without declaring the SQLCA. When MODE={ANSI |ANSI14}, declaring the SQLCA is optional, but to use the WHENEVER SQLWARNING statement, the SQLCA must be declared.

Action: Remove all WHENEVER SQLWARNING statements from the program or declare the SQLCA by hardcoding it or copying it into the program with the INCLUDE statement.

PCC-00044

PCC-00044   Array size mismatch in INTO/USING. Minimum is: min (num:num)

Cause: The size of an array variable in an INTO/USING clause is too small for the number of rows processed.

Action: Declare all array variables in the INTO/USING clause to have at least the minimum dimension given.

PCC-00045

PCC-00045   **clause_name clause inappropriate at line num in file name. Ignored**

Cause: There is a misplaced clause at the end of an EXEC SQL statement (an AT clause at the end of a SELECT statement, for example), or the action specified in a FOR clause is invalid (for example, FOR:loop INTO...).

Action: Check the statement syntax, then relocate or correct the misplaced orinvalid clause.

PCC-00047

PCC-00047   **Unterminated comment/string constant beginning near line num in file name**

Cause: A string constant is missing an ending quotation mark, or a comment is missing an ending delimiter.

Action: Check that all comments are delimited and all string constants are enclosed by quotation marks.

PCC-00050

PCC-00050 **Unable to generate descriptor in program unit ending line num in file name**

Cause: The precompiler was unable to generate a descriptor for the compilation unit. This can occur from either of the following:

- a non-existent or improperly implemented Declare Section
- undetected syntax errors near the beginning of the source file or just before the END DECLARE SECTION statement

Action: Verify that the Declare Section is properly implemented. Then check for syntax errors at the beginning of the compilation unit and before any END DECLARE SECTION statement, and correct as appropriate.

PCC-00051

PCC-00051   **Size of VARCHAR hostvar at num is larger than 65533 at line num infile name**

Cause: The declared size of a VARCHAR host variable exceeds the precompiler limit of 65533 bytes.

Action: Check the Declare Section, making sure the size of each VARCHAR variable does not exceed 65533 bytes.

PCC-00053

PCC-00053   **FOR variable var is invalid type at line num in file name**

Cause: The count variable in a FOR clause has the wrong datatype. The datatype must be NUMBER or LONG (or be compatible with NUMBER or LONG).

Action: Check the declaration and check that the count variable has a datatype of NUMBER or LONG (or a compatible Oracle or host-language datatype).

PCC-00054

PCC-00054 **Expected end-of-statement at column name in line num of file name**

Cause: The precompiler expected to find a statement terminator at the end of an EXEC statement but found something else. This can happen if tabs were embedded in the source code (because the precompiler has no way of knowing how many spaces a tab represents).

Action: If tabs are embedded in the source code, replace them with spaces. Check the statement syntax and check that each EXEC statement has a terminator. For embedded CREATE {FUNCTION|PROCEDURE|PACKAGE} statements and for embedded PL/SQL blocks, check that the statement terminator is END-EXEC.

Action: Correct or remove the erroneous CREATE statement.

PCC-00055

PCC-00055   **Array name not allowed as bind variable at line num in file name**

Cause: A host array was used as a bind (input) variable in the WHERE clause of a SELECT statement. This is not allowed.

Action: Remove the host array or replace it with a simple host variable.

PCC-00056

PCC-00056   **FOR clause not allowed in SELECT statement at line num in file name**

Cause: FOR clause was used with a SELECT statement. This is not allowed, as data returned from the last loop in the execution of the SELECT overwrites data returned in a previous loop.

Action: Remove the FOR clause from the SELECT statement and use a host-language construct to iteratively execute the SELECT statement.

PCC-00060

PCC-00060 **Both CURSOR and STATEMENT have AT clauses at line num of file name**

Cause: Two AT clauses, one in a DECLARE STATEMENT statement, the other in a DECLARE CURSOR statement, pertain to the same SQL statement. The AT clause may be specified with either DECLARE STATEMENT or DECLARE CURSOR but not with both.

Action: Remove the AT clause from one of the statements.

PCC-00061

PCC-00061   **Error at line num, column name. PLS-nnnnn**

Cause: The precompiler found an error in an embedded SQL statement orPL/SQL block.

Action: Refer to the indicated PL/SQL error message and correct the SQL statement or embedded PL/SQL block. See Chapter 3msgchpt3 "PL/SQL Message"..

PCC-00062

PCC-00062   **Must use option SQLCHECK=SEMANTICS when there isembedded PL/SQL**

Cause: The precompiler tried to parse an embedded PL/SQL block when SQLCHECK={SYNTAX|NONE}. PL/SQL blocks can be parsed only when SQLCHECK=SEMANTICS is specified.

Action: Remove the PL/SQL block or specify SQLCHECK=SEMANTICS.

PCC-00063

PCC-00063   Reached end of file name before end-of-statement at line num

Action: The precompiler encountered an end-of-file while parsing a PL/SQL block.

Action: Add the appropriate statement terminator (;) or end-of-block statement (END;) to the PL/SQL block.

PCC-00064

PCC-00064   **All uses of a given host variable must use identical indicator variables**

Cause: Two or more occurrences of a host variable in an EXEC SQL statement were associated with different indicator variables. This is not allowed.

Action: Rename the indicator variables so that each occurrence of the host variable is associated with the same indicator variable.

PCC-00065

PCC-00065   **USERID required, but not specified**

Cause: The SQLCHECK=SEMANTICS option was specified, but the USERID option on the command line was not specified.

Action: Specify USERID=username/password or enter a username and password when prompted or specify SQLCHECK={SYNTAX|NONE}.

PCC-00066

PCC-00066   USERID only used when SQLCHECK=SEMANTICS, USERID ignored

Cause: The USERID option was specified when SQLCHECK={SYNTAX|NONE}. This is unnecessary.

Action: Specify the USERID option only when SQLCHECK=SEMANTICS.

PCC-00067

PCC-00067   **IRECLEN exceeded. Line num in file name truncated**

Cause: While reading the input file, the precompiler found a line longerthan IRECLEN.

Action: Either shorten the input line or specify a larger IRECLEN value on the command line.

PCC-00068

PCC-00068   **Host and indicator variables may not have the same name**

Cause: In an EXEC SQL statement, an indicator variable had the same name as a host variable. The names of a host variable and its associated indicator variable must be different. Also, an indicator variable cannot be used as a host variable.

Action: Rename the host or indicator variable.

PCC-00069

PCC-00069   **Host variable var has unsupported datatype at line num in file name**

Cause: A host variable had an unsupported datatype. For a list of supported datatypes, see the language-specific supplement to the Programmer's Guide to the Oracle Precompilersorg@LGPG7.hlp.

Action: Redefine the host variable in the Declare Section, giving it asupported datatype.

PCC-00070

PCC-00070   **Illegal syntax. Exponential value in SQL statement: text**

Cause: The precompiler found a syntax error while parsing a number coded in scientific notation. The precompiler expected to find a signed integer following the exponentiation indicator (E), but found something else.

Action: Reformat the number correctly.

PCC-00072

PCC-00072   **Input file name length exceeds 14 characters**

Cause: The filename specified exceeded the maximum length. On some operating systems, the maximum length of a filename is 14 characters.

Action: Use a filename of 14 or fewer characters.

PCC-00073

PCC-00073   **Cursor is declared but never OPENed at line num in file name**

Cause: A cursor was DECLAREd but was not referenced in an OPEN statement. This is only an informational message.

Action: Remove the cursor declaration or code an OPEN statement for the cursor.

PCC-00074

PCC-00074   **FIPS warning: Multiply defined host variable in line num of file name**

Cause: An Oracle extension to the ANSI/ISO SQL standard was used. Specifically, the name of a global host variable was used to declare a local host variable. This informational message is issued by the FIPS Flagger when FIPS=YES.

Action: No action required. However, for ANSI/ISO compliance, do not reuse the names of global host variables to declare local host variables.

PCC-00075

PCC-00075   **":" expected before indicator variable**

Cause: An indicator variable was not prefixed with a colon, as required.

Action: Prefix a colon to the indicator variable in question.

PCC-00076

PCC-00076   **DISPLAY type must be SIGN LEADING SEPARATE**

Cause: This message is issued only by Pro*COBOL. DISPLAY SIGN LEADING SEPARATE is the only DISPLAY type supported by Pro*COBOL.

Action: Check the spelling of the variable declaration. If necessary, remove the reference to the unsupported DISPLAY type.

PCC-00077

PCC-00077   Colon usage with numeric label in WHENEVER statement is not ANSI

Cause: An Oracle extension to the ANSI/ISO SQL standard was used. Specifically, a numeric WHENEVER... GOTO label was prefixed with a colon. For example, the code might have looked like:

EXEC SQL WHENEVER SQLERROR GOTO :99;

This informational message is issued by the FIPS Flagger when FIPS=YES.

Action: No action required. However, for ANSI/ISO compliance, prefix alphanumeric (but not numeric) WHENEVER... GOTO labels with a colon.

PCC-00078

PCC-00078   FIPS warning: Invalid ANSI SQL identifier

Cause: An Oracle extension to the ANSI/ISO SQL standard was used. Specifically, the name given to a host variable:

- is longer than 18 characters,
- does not begin with a letter, or
- contains consecutive or trailing underscores

In the following Pro*C example, the host variable name is 19 characters long and therefore non-compliant:

```
EXEC SQL BEGIN DECLARE SECTION;

    int   department_location;   -- not ANSI/ISO-compliant

     ...

EXEC SQL END DECLARE SECTION;
```

This informational message is issued by the FIPS Flagger when FIPS=YES.

Action: No action required. However, for ANSI/ISO compliance, change the host variable name so that it is <=18 characters long, begins with a letter, and does not contain consecutive or trailing underscores.

PCC-00079

PCC-00079   **ANSI requires colon on label in WHENEVER statement**

Cause: An Oracle extension to the ANSI/ISO SQL standard was used. Specifically, an alphanumeric WHENEVER... GOTO label was not prefixed with a colon. For example, the offending code might look like:

```
EXEC SQL WHENEVER NOT FOUND GOTO no_more;
```

This message is only a warning issued by the FIPS Flagger when FIPS=YES.

Action: No action required. However, for ANSI/ISO compliance, prefix alphanumeric (but not numeric) WHENEVER... GOTO labels with a colon.

PCC-00080

PCC-00080   **TYPE identifier already TYPEd**

Cause: The identifier being TYPEd in an EXEC SQL TYPE statement appeared in a previous EXEC SQL TYPE statement. A given identifier can appear in only one EXEC SQL TYPE statement.

Action: Check the spelling of the identifiers. Use different identifiers in the EXEC SQL TYPE statements, or remove one of the EXEC SQL TYPE statements.

PCC-00081

PCC-00081   **Scale specification not allowed for given datatype**

Cause: The Oracle external datatype referenced in an EXEC SQL VAR or EXEC SQL TYPE statement does not allow a scale specification.

Action: Check the precision specification and remove the scale specification.

PCC-00082

PCC-00082   **Length and scale specifications must be an integer**

Cause: A floating point number or a non-number to specify a length or scale was used. Only integers can be used.

Action: Correct or remove the length and/or scale specification.

PCC-00083

PCC-00083   **Bind and define variables not allowed in CREATE statement**

Cause: Host variables cannot appear in a CREATE statement. If the makeup of a CREATE statement cannot be known until run time, a dynamic SQL statement must be used to execute it. That is, the program must accept or build the CREATE statement at run time, store it in a host string, then EXECUTE it.

Action: Correct or remove the erroneous CREATE statement.

PCC-00085

PCC-00085   Error writing to file name

Cause: The precompiler was unable to write to the named output file. Some possible causes follow:

- Sufficient file access privileges were not defined.
- Another user has locked the file.
- There is not enough disk space.
- There are too many open files.

Action: Check that sufficient privileges exist to access the file and that it is not locked by another user. Also check that there is enough disk space and that the limit for open files is set high enough (check with the system manager).

PCC-00086

PCC-00086   **Source file name has zero length**

Cause: The source file specified on the command line contains no code. Consequently, there is nothing for the precompiler to process.

Action: Specify a valid source file containing embedded SQL statements.

PCC-00087

PCC-00087   EXEC SQL TYPE statement not allowed for this host language

Cause: An EXEC SQL TYPE statement was used with a host language that does not support user-defined datatype
equivalencing. This feature is available only in Pro*C and Pro*Pascal.

Action: Remove the offending EXEC SQL TYPE statement.

PCC-00088

PCC-00088   **User-defined type identifier expected**

Cause: The user-defined datatype name in an EXEC SQL TYPE statement is missing or misspelled, is a reserved word, is not a legal identifier in the host language or conflicts with a base datatype in that language.

Action: Check the spelling of the user-defined datatype name. If necessary, declare a valid user-defined datatype. User-defined datatype equivalencing is available only in Pro*C and Pro*Pascal.

PCC-00089

PCC-00089   **Invalid Oracle TYPE specification**

Cause: The Oracle external datatype name in an EXEC SQL TYPE or EXEC SQL VAR statement is missing or misspelled.

Action: Check the spelling of the external datatype name. If necessary, supply the missing datatype name.

PCC-00090

PCC-00090   **Precision/scale specification must be given for DECIMAL datatype**

Cause: A precision and/or scale specification for the Oracle external datatype DECIMAL in an EXEC SQL TYPE or EXEC SQL VAR statement was omitted.

Action: Add the precision and/or scale specification to the EXEC SQL TYPE or EXEC SQL VAR statement.

PCC-00091

PCC-00091 **TYPE statement requires format specification for this Oracle datatype**

Cause: A length, precision, and/or scale specification for an Oracle external datatype in an EXEC SQL TYPE or EXEC SQL VAR statement was omitted.

Action: Add the length, precision, and/or scale specification for the external datatype to the EXEC SQL TYPE or EXEC SQL VAR statement.

PCC-00092

PCC-00092	**Length and/or scale incompatible with specified Oracle datatype**

Cause: An invalid length or scale for an Oracle external datatype in an EXEC SQL TYPE or EXEC SQL VAR statement was specified.

Action: Check that a length large enough to accommodate the external datatype is specified. If a scale is specified, check that it lies in the range -84 .. 99.

PCC-00093

PCC-00093   **Invalid or obsolete option, ignored**

Cause: The precompiler found an option available in a prior version or different host language but not in the current version or host language.

Action: Remove the option specification.

PCC-00094

PCC-00094   **Array length for char[n] datatype must be => 2**

Cause: When MODE={ANSI|ANSI14}, a length of less than 2 characters for a char[n] host variable was specified or a simple CHAR variable was specified. When MODE={ANSI|ANSI14}, the length must be at least 2 characters. This message is issued only by the Pro*C Precompiler.

Action: Correct the declaration so that it specifies a length of at least 2 characters.

PCC-00095

PCC-00095 **Missing PROGRAM, SUBROUTINE, FUNCTION, or BLOCKDATA statement**

Cause: FORTRAN source files are expected to have at least one PROGRAM, SUBROUTINE, FUNCTION, or BLOCK DATA statement, which the precompiler uses to detect the beginning of a routine or compilation unit.

Action: Add one of these statements to the source file.

PCC-00096

PCC-00096   Array FETCH not allowed for MODE=ANSI14

Cause: When MODE=ANSI14, an array SELECT or FETCH was attempted. However, array operations are not allowed when MODE=ANSI14.

Action: If MODE=ANSI14 must be specified, place the SELECT or FETCH statement in a host-language loop, instead of using the array interface.

PCC-00097

PCC-00097   **Use of DECIMAL and DISPLAY types allowed only for COBOL and PLI**

Cause: The DECIMAL or DISPLAY external datatype was used in an EXEC SQL VAR statement with an Oracle Precompiler other than Pro*COBOL or Pro*PL/I. These external datatypes are available only in Pro*COBOL and Pro*PL/I.

Action: Remove the reference to the DECIMAL or DISPLAY external datatype from the EXEC SQL VAR statement.

PCC-00098

PCC-00098   **Scale specification cannot be used in this context**

Cause: In a Pro*C, Pro*FORTRAN, or Pro*Pascal program, scale in an EXEC SQL TYPE or EXEC SQL VAR statement in the current context cannot be specified.

Action: Remove the scale specification from the EXEC SQL TYPE or EXEC SQLVAR statement.

PCC-00099

PCC-00099   Length cannot be given for types ROWID, DATE, or MLSLABEL

Cause: A length for the ROWID, DATE, or MLSLABEL external datatype was specified in an EXEC SQL TYPE or EXEC SQL VAR statement. This is unnecessary because those are fixed-length types.

Action: Remove the length specification from the EXEC SQL TYPE or EXECSQL VAR statement.

PCC-00100

PCC-00100	**Non integer label is not ANSI**

Cause: An Oracle extension to the ANSI/ISO SQL standard was used. Specifically, a non-integer WHENEVER... GOTO label was in a Pro*Pascal program. For example, the offending code might look like:

```
EXEC SQL WHENEVER NOT FOUND GOTO 5.0;
```

This informational message is issued by the FIPS Flagger when FIPS=YES.

Action: No action required. However, for ANSI/ISO compliance, use only integer WHENEVER... GOTO labels in a Pro*Pascal program.

PCC-00101

PCC-00101   **Lower case 'e' in floating point number is not ANSI**

Cause: An Oracle extension to the ANSI/ISO SQL standard was used. Specifically, a lowercase "e" was used in scientific notation. For example, the offending code might look like:

```
maxnum = 10e38;
```

This informational message is issued by the FIPS Flagger when FIPS=YES.

Action: No action required. However, for ANSI/ISO compliance, use an uppercase 'E' in scientific notation.

PCC-00102

PCC-00102   **FOR UPDATE is an Oracle extension**

Cause: An Oracle extension to the ANSI/ISO SQL standard was used. Specifically, the FOR UPDATE OF clause was used in a cursor declaration. For example, the offending code might look like:

```
EXEC SQL DECLARE emp_cursor CURSOR FOR

    SELECT ENAME, SAL FROM EMP WHERE DEPTNO = :dept_number

        FOR UPDATE OF SAL;
```

This informational message is issued by the FIPS Flagger when FIPS=YES.

Action: No action required. However, for ANSI/ISO compliance, do not use the FOR UPDATE OF clause.

PCC-00103

PCC-00103   AT clause is an Oracle extension

Cause: An Oracle extension to the ANSI/ISO SQL standard was used. Specifically, the AT db_name clause was used in a SQL statement. For example, the offending code might look like:

```
EXEC SQL AT oracle3 COMMIT RELEASE;
```

This informational message is issued by the FIPS Flagger when FIPS=YES.

Action: No action required. However, for ANSI/ISO compliance, do not use the AT db_name clause.

PCC-00104

PCC-00104   **FOR clause is an Oracle extension**

Cause: An Oracle extension to the ANSI/ISO SQL standard was used. Specifically, the FOR clause was used in an array-processing SQL statement. For example, the offending code might look like:

```
EXEC SQL FOR :limit INSERT INTO EMP (EMPNO, JOB, DEPTNO)

    VALUES (:emp_number, :job_title, :dept_number);
```

This informational message is issued by the FIPS Flagger when FIPS=YES.

Action: No action required. However, for ANSI/ISO compliance, do not use theFOR clause.

PCC-00105

PCC-00105   **Keyword WORK required here by ANSI**

Cause: An Oracle extension to the ANSI/ISO SQL standard was used. Specifically, the keyword WORK was used in a COMMIT or ROLLBACK statement. For example, the offending code might look like:

```
EXEC SQL COMMIT WORK RELEASE;
```

This informational message is issued by the FIPS Flagger when FIPS=YES.

Action: No action required. However, for ANSI/ISO compliance, do not use the keyword WORK.

PCC-00106

PCC-00106   **RELEASE is an Oracle extension to the COMMIT and ROLLBACK statements**

Cause: An Oracle extension to the ANSI/ISO SQL standard was used. Specifically, the parameter RELEASE was used in a COMMIT or ROLLBACK statement. For example, the offending code might look like:

```
EXEC SQL ROLLBACK RELEASE;
```

This informational message is issued by the FIPS Flagger when FIPS=YES.

Action: No action required. However, for ANSI/ISO compliance, do not use the parameter RELEASE.

PCC-00107

PCC-00107   **The CONNECT statement is Oracle implementation dependent**

Cause: An Oracle extension to the ANSI/ISO SQL standard was used. Specifically, the CONNECT statement was used to log on to Oracle. For example, the offending code might look like:

```
EXEC SQL CONNECT :username IDENTIFIED BY :password;
```

This informational message is issued by the FIPS Flagger when FIPS=YES.

Action: No action required. However, for ANSI/ISO compliance, specify the AUTO_CONNECT command-line option instead of using aCONNECT statement.

PCC-00108

PCC-00108 **This statement is not supported by ANSI**

Cause: An Oracle extension to the ANSI/ISO SQL standard was used. Specifically, a nonconforming SQL statement such as PREPARE was used. For example, the offending code might look like:

```
EXEC SQL PREPARE sql_statement FROM :sql_string;
```

This informational message is issued by the FIPS Flagger when FIPS=YES.

Action: No action required. However, for ANSI/ISO compliance, do not use the nonconforming SQL statement.

PCC-00109

PCC-00109   **Dynamic SQL and PL/SQL are Oracle extensions to ANSI SQL**

Cause: An Oracle extension to the ANSI/ISO SQL standard was used. Specifically, dynamic SQL or embedded PL/SQL was used. For example, the offending code might look like:

```
EXEC SQL EXECUTE
    BEGIN
        SELECT ...
        ...
    END;
END-EXEC;
```

This informational message is issued by the FIPS Flagger when FIPS=YES.

Action: No action required. However, for ANSI/ISO compliance, do not use dynamic SQL or embedded PL/SQL.

PCC-00110

PCC-00110   **Oracle extension to the WHENEVER statement**

Cause: An Oracle extension to the ANSI/ISO SQL standard was used. Specifically, a nonconforming keyword such as NOTFOUND, STOP, RAISE, or DO was used in the WHENEVER statement. (Note that NOT FOUND is ANSI-compliant.) For example, the offending code might look like:

```
EXEC SQL WHENEVER SQLERROR STOP;
```

This informational message is issued by the FIPS Flagger when FIPS=YES.

Action: No action required. However, for ANSI/ISO compliance, do not use the nonconforming keyword.

PCC-00111

PCC-00111   **SQLCHECK value in EXEC ORACLE statement exceeds command line value**

Cause: The SQLCHECK option was entered inline and specified a level of checking higher than the level specified (or accepted by default) on the command line. This is not allowed. For example, if SQLCHECK={SYNTAX|LIMITED} is specified on the command line, SQLCHECK={SEMANTICS|FULL} cannot be specified inline.

This message is only a warning; the precompiler ignores the inline value and continues processing.

Action: Revise the EXEC ORACLE statement or specify a lower level of checking on the command line.

PCC-00112

PCC-00112   **Datatype not supported by ANSI**

Cause: An Oracle extension to the ANSI/ISO SQL standard was used. Specifically, a pointer or nonconforming datatype such as VARCHAR was used. For example, the offending code might look like:

```
EXEC SQL BEGIN DECLARE SECTION;

    VARCHAR  username[20];

    ...

EXEC SQL END DECLARE SECTION;
```

This informational message is issued by the FIPS Flagger when FIPS=YES.

Action: No action required. However, for ANSI/ISO compliance, do not use pointers or nonconforming datatypes.

PCC-00113

PCC-00113   **Value of DBMS option invalid with given value of MODE option**

Cause: When MODE={ANSI14|ANSI13}, DBMS=V7 was specified, or when MODE=ANSI, DBMS=V6 was specified. These option settings are incompatible. Note that the DBMS option was not available before release 1.5 of the Oracle Precompilers.

Action: With DBMS=V7, instead of MODE={ANSI14|ANSI13}, specify MODE={ANSI|ORACLE}. With DBMS=V6, instead of MODE=ANSI, specify MODE={ANSI14|ANSI13|ORACLE} but MODE=ORACLE is recommended.

PCC-00114

PCC-00114   **Length spec required in EXEC SQL VAR statements for VARxxx types**

Cause: In a EXEC SQL VAR statement, a VARCHAR or VARRAW external datatype was specified without a length. Unlike other types, the maximum length of the data field must be specified for VARCHAR and VARRAW.

Action: Add a length specification to the EXEC SQL VAR statement.

PCC-00115

PCC-00115   Array required here

Cause: In an ARRAYLEN statement, the name of a previously declared host array was not specified. The first host variable in an ARRAYLEN statement must be an array. The second host variable, which specifies an array dimension, must be a 4-byte integer. The correct syntax follows:

```
EXEC SQL ARRAYLEN host_array (dimension);
```

The ARRAYLEN statement must appear in the Declare Section along with, but somewhere after, the declarations of host_array and dimension.

Action: Check the spelling of both identifiers in the ARRAYLEN statement. If necessary, supply the missing host array name.

PCC-00116

PCC-00116   **This array already given in an ARRAYLEN statement**

Cause: The same host array was specified in two different ARRAYLEN statements. A given host array cannot be specified in more than one ARRAYLEN statement.

Action: Check the spelling of the host array names in both ARRAYLEN statements. Change one of the names so that they refer to different host arrays or remove one of the ARRAYLEN statements.

PCC-00117

PCC-00117   **Invalid ARRAYLEN length variable type**

Cause: A valid array dimension was not specified in an ARRAYLEN statement. The array dimension must be specified using a previously declared 4-byte integer host variable, not a literal or expression. For example, the offending code might look like:

```
EXEC SQL ARRAYLEN ename_array (25);  -- illegal dimension
```

Action: Supply a valid array dimension. If necessary, declare a 4-byte integer host variable for use in the ARRAYLEN statement.

PCC-00118

PCC-00118   **Use of host variable initialization not supported by ANSI SQL**

Cause: An Oracle extension to the ANSI/ISO SQL standard was used. Specifically, a host variable was initialized in its declaration, as shown in the following Pro*C example:

```
EXEC SQL BEGIN DECLARE SECTION;

    int  dept_number = 20;  -- not ANSI/ISO-compliant

    ...

EXEC SQL END DECLARE SECTION;
```

This informational message is issued by the FIPS Flagger when FIPS=YES.

Action: No action required. However, for ANSI/ISO compliance, do not initialize host variables in their declarations.

PCC-00119

PCC-00119   **Value of const variable in INTO clause will be modified**

Cause: A variable declared with the type specifier const was used in an INTO clause. Such variables should not be modified and should not be used in an INTO clause.

Action: Check the spelling of all identifiers in the INTO clause. If necessary, remove const from the host variable declaration or use a different host variable.

PCC-00120

PCC-00120   **File I/O error during code generation**

Cause: A file I/O error occurred during code generation. This may be caused by insufficient disk space.

Action: Check that there is enough disk space.

PCC-00121

PCC-00121   **Arrays of VARCHAR pointers are not supported**

Cause: An array of pointers was declared, which is not allowed. However, pointers to scalar types are allowed. With Pro*C, declare pointers to char[n] and VARCHAR[n] variables as pointers to CHAR or VARCHAR (with no length specification).

Action: Correct or remove the declaration.

PCC-00122

PCC-00122   **Input file name and output file name are identical**

Cause: On the command line, the same pathname for INAME and ONAME was specified, which designates the precompiler input and output files, respectively.

Action: Change one of the path/filenames.

PCC-00123

PCC-00123 **Entire VARCHAR declaration must be on same line**

Cause: In a Pro*C program, a VARCHAR declaration spans more than one line, which is not allowed.

Action: Revise the declaration so that it uses only one line.

PCC-00124

PCC-00124   COMMON_NAME option must be before PROGRAM or subroutine beginning

Cause: In a FORTRAN program, subroutine, or function, the precompiler option COMMON_NAME was mistakenly specified after the PROGRAM, SUBROUTINE, or FUNCTION statement. If COMMON_NAME is specified inline, its EXEC ORACLE OPTION statement must precede the PROGRAM, SUBROUTINE, or FUNCTION statement.

Action: Relocate the EXEC ORACLE OPTION statement or specify COMMON_NAME on the command line.

PCC-00126

PCC-00126   Could not find or open system configuration file

Cause: The precompiler was unable to find or open the system configuration file (a text file containing preset command-line options, which the precompiler uses by default). Some possible causes follow:

- The file does not exist.
- The search path to the file is incorrect.
- File access privileges are insufficient.
- There are too many open files.

However, this message is just a warning. Processing continues even if the system configuration file does not exist.

Action: Check that the file exists, that the search path to the file is correct, and that sufficient privileges exist to access the file. Also check that the limit for open files is set high enough (check with the system manager).

PCC-00127

PCC-00127 **text**

Cause: This is a generic error message from the command-line processor.

Action: Correct the indicated error.

PCC-00128

PCC-00128   **Command line processor severe error**

Cause: The command line processor encountered a severe error.

Action: After making sure that all the command-line options are specified correctly,call customer support with a full account of all the options and configuration files used.

PCC-00129

PCC-00129   **option optnam must be given**

Cause: A required command-line option is missing. For example, the INAME option, which specifies the name of the input file, might be missing.

Action: Supply the missing command-line option.

PCC-00130

PCC-00130   **Invalid value val for option optnam**

Cause: The precompiler found an invalid option value. For example, the value might have been misspelled (as in DBMS=6V) or illegal (as in DBMS=V5).

Action: Check the value, making sure that it is legal and correctly spelled.

PCC-00132

PCC-00132 **Indicator array size must not be less than its host variable**

Cause: An host variable array was used with an indicator array declared with a smaller dimension. For example:

```
EXEC SQL BEGIN DECLARE SECTION;
int   dept_no[20];
short dept_no_ind[10];
EXEC SQL END DECLARE SECTION;
...
SELECT ... INTO dept_no:dept_no_ind ...
```

Action: Increase the size of the indicator array.

PCC-00133

PCC-00133 **Command line option MODE=ANSI required with option NLS_LOCAL=YES**

Cause: The precompiler option NLS_LOCAL=YES was used without also specifying MODE=ANSI. The precompiler option MODE=ANSI must be specified if NLS_LOCAL=YES.

Action: Set the precompiler option MODE=ANSI in addition to NLS_LOCAL=YES or do not set NLS_LOCAL=YES.

PCC-00135

PCC-00135   **Result Set Cursor usage is not standard SQL**

Cause: An Oracle extension to the ANSI/ISO SQL standard was used. Specifically, an Oracle Result Set Cursor was used. This informational message is issued by the FIPS Flagger when FIPS=YES.

Action: No action required. However, for ANSI/ISO compliance, do not use Result Set Cursors.

PCC-00136

PCC-00136   **Invalid SQL_CURSOR declaration**

Cause: An array of SQL_CURSOR host variables was declared or, if Pro*COBOL, a PICTURE clause was specified in the SQL_CURSOR declaration. In Pro*FORTRAN, a length specification (i.e., "*n") on the SQL_CURSOR declaration may have been specified. These usages are not supported.

Action: Check that the SQL_CURSOR variable is not declared as an array or, if Pro*COBOL, that no PICTURE clause is specified for the SQL_CURSOR. In Pro*FORTRAN, check that there is no length specification (i.e., "*n") for the SQL_CURSOR.

PCC-00137

PCC-00137   **Invalid use of SQL_CURSOR host variable**

Cause: A SQL_CURSOR variable was used in an INTO or WHERE clause. SQL_CURSOR variables can be used only where ordinary cursor identifiers or, in a PL/SQL block, a cursor would normally be used.

Action: Remove the SQL_CURSOR variable reference from the INTO or WHERE clause, or use a host variable of another type in the INTO or WHERE clause.

PCC-00138

PCC-00138   **Result Set Cursors are not implemented in this precompiler**

Cause: There was an attempt to reference a host variable as a cursor reference. Cursor variables are not implemented in Pro*Pascal or Pro*PL/I.

Action: Rewrite your host-language code to use standard SQL cursors.

PCC-00140

PCC-00140   CREATE FUNCTION/PROCEDURE/PACKAGE/TRIGGER are Oracle extensions

Cause: An Oracle extension to the ANSI/ISO SQL standard was used. Specifically, one of the following statements was used:

- CREATE FUNCTION
- CREATE PROCEDURE
- CREATE PACKAGE
- CREATE TRIGGER

These statements are Oracle extensions to the ANSI/ISO SQL standards. This informational message is issued by the FIPS Flagger when FIPS=YES.

Action: No action is required. However, for ANSI/ISO compliance, do not use any of these statements.

PCC-01000

PCC-01000   **You are not authorized to run Pro*COBOL**

Cause: The authorization or license to run the Pro*COBOL Precompiler has expired.

Action: Call customer support for assistance.

PCC-01001

PCC-01001   **Your Pro*COBOL authorization is about to expire**

Cause: The authorization or license to run the Pro*COBOL Precompiler is aboutto expire.

Action: Call customer support for assistance.

PCC-01002

PCC-01002   **Invalid character char in indicator area at line num in file name**

Cause: In a Pro*COBOL Precompiler program, only a blank, hyphen (-), asterisk (*), slash (/), or letter "D" is allowed in the indicator area, but the precompiler found another character.

Action: Remove or replace the invalid character. If the FORMAT=ANSI option is specified, check for an end-of-line in column 7.

PCC-01003

PCC-01003   **Invalid continuation at line num in file name**

Cause: In a Pro*COBOL program, a continuation line was completely blank, except for the continuation character.

Action: Remove or replace the empty continuation line.

PCC-01004

PCC-01004   In an EXEC statement at end-of-file

Cause: In a Pro*COBOL input file, the last EXEC statement was notterminated properly.

Action: Terminate the last EXEC statement with an END-EXEC.

PCC-01005

PCC-01005   PROCEDURE DIVISION not found

Cause: The precompiler could not find the PROCEDURE DIVISION header in a Pro*COBOL program. Some possible causes follow:

- A keyword in the header is missing or misspelled.
- There is an apostrophe in the REMARKS section (the precompiler mistook the apostrophe for the beginning of a string literal).
- There is an unterminated literal in the WORKING-STORAGE SECTION.
- The wrong value for the FORMAT option was specified.

Action: Check that the PROCEDURE DIVISION header is in place and spelled correctly, that there is no apostrophe in the REMARKS section, that all literals in the WORKING-STORAGE SECTION are terminated, and that the right value for the FORMAT option is specified.

PCC-01006

PCC-01006   **EXEC statement cannot begin in Area A at line num in file name**

Cause: In a Pro*COBOL program, EXEC statements must begin in Area B, but the precompiler found a statement beginning in Area A.

Action: Move the statement rightward so that it begins in Area B.

PCC-01007

PCC-01007 **WORKING-STORAGE SECTION not found**

Cause: The precompiler could not find the WORKING-STORAGE SECTION header in a Pro*COBOL program, probably because a keyword is missing or misspelled, or the wrong value for the FORMAT option might have been specified.

Action: Check that the WORKING-STORAGE SECTION header is in place and spelled correctly and that the right value for the FORMAT option is specified.

PCC-01008

PCC-01008   **Multiple element records not allowed in Declare Section**

Cause: A few COBOL compilers do not allow group items to be passed as parameters in a CALL statement. (Check the COBOL compiler user's guide.) If the compiler is one of these, group items within the Declare Section can contain only one elementary item.

Action: Assign each host variable its own group item.

PCC-01009

PCC-01009   **For HOST=COB74, a SQL statement must be followed by ELSE or "."**

Cause: In a Pro*COBOL program, an EXEC SQL statement is followed by another statement in the same sentence. An EXEC SQL statement must be the last statement in a COBOL-74 sentence and so must be terminated by the keyword ELSE or a period.

Action: Change the program logic, making the EXEC SQL statement the last statement in the sentence.

PCC-01010

PCC-01010   **Invalid use of NULL character in character literal**

Cause: A null character (binary zero) was found in a string literal. This is not allowed by Pro*COBOL.

Action: Remove the null character from the string literal.

PCC-01011

PCC-01011   **USAGE IS BINARY clause must be terminated by "." on same line**

Cause: The period terminating the USAGE IS BINARY clause must be on the same line as the USAGE clause. This message only occurs on operating systems that support the COMP5=YES command line option of Pro*COBOL, where COMP-5 is substituted for BINARY.

Action: Check that the period terminating the USAGE IS BINARY clause is on the same line as the USAGE clause.

PCC-01012

PCC-01012   **DATA DIVISION not found**

Cause: The Pro*COBOL precompiler did not find a DATA DIVISION in the input source file. Pro*COBOL applications are required to have a DATA DIVISION.

Action: Add a DATA DIVISION to the input source file.

PCC-01013

PCC-01013   **Invalid use of PIC N array variable "variable_name" at line line_number in file name**

Cause: A PIC N variable was declared using an OCCURS clause. Tables of PIC N variables are not supported by the Pro*COBOL precompiler.

Action: Declare the PIC N variable without an OCCURS clause or, if the PIC N variable is not used in any SQL statements, declare it outside the Declare Section.

PCC-01014

PCC-01014 **"VARYING" not supported with PIC N PICTURE clause**

Cause: The keyword VARYING was used in a PIC N variable declaration.

Action: Remove the keyword VARYING from the variable declaration. If you want to declare a PIC N VARCHAR variable, specify the precompiler option VARCHAR=YES and declare the variable as an implicit VARCHAR group item as illustrated by the following example:

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.

    ...

    01  ENAME.

        05 ENAME-LEN  PIC S9(4) COMP.

        05 ENAME-ARR  PIC N(20).

    ...

EXEC SQL END DECLARE SECTION END-EXEC.
```

PCC-01015

PCC-01015   **Cannot equivalence PIC N variable name**

Cause: A PIC N variable or an implicit VARCHAR group item (with a PIC N variable as an elementary item) was used in an EXEC SQL VAR statement. This is not allowed.

Action: Do not use a PIC N variable in an EXEC SQL VAR statement. If you want an equivalence to an Oracle type using an EXEC SQL VAR statement, use a PIC X variable instead of a PIC N variable.

PCC-01016

PCC-01016   **"N" character literals not supported in embedded PL/SQL**

Cause: An "N" character literal was used within a PL/SQL block.

Action: Remove the "N" character literal from the PL/SQL block.

PCC-01100

PCC-01100   **You are not authorized to run Pro*FORTRAN**

Cause: The authorization or license to run the Pro*FORTRAN Precompiler has expired.

Action: Call customer support for assistance.

PCC-01101

PCC-01101   **Your Pro*FORTRAN authorization is about to expire**

Cause: The authorization or license to run the Pro*FORTRAN Precompiler is aboutto expire.

Action: Call customer support for assistance.

PCC-01102

PCC-01102   **Invalid label at line num in file name**

Cause: The Pro*FORTRAN Precompiler found an invalid FORTRAN statement label in columns 1 through 6.

Action: Correct or remove the statement label.

PCC-01200

PCC-01200   You are not authorized to run Pro*C

Cause: The authorization or license to run the Pro*C Precompiler has expired.

Action: Call customer support for assistance.

PCC-01201

PCC-01201   Your Pro*C authorization is about to expire

Cause: The authorization or license to run the Pro*C Precompiler is about to expire.

Action: Call customer support for assistance.

PCC-01202

PCC-01202   **Identifier ident truncated to 31 characters**

Cause: A host identifier (the name of a host variable, for example) was truncated to the maximum length (31 characters) allowed by the precompiler.

Action: No action required. This message is just informational.

PCC-01300

PCC-01300   **You are not authorized to run Pro*PL/I**

Cause: The authorization or license to run the Pro*PL/I Precompiler has expired.

Action: Call customer support for assistance.

PCC-01301

PCC-01301   **Your Pro*PL/I authorization is about to expire**

Cause: The authorization or license to run the Pro*PL/I Precompiler is about to expire.

Action: Call customer support for assistance.

PCC-01400

PCC-01400   **You are not authorized to run Pro*Pascal**

Cause: The authorization or license to run the Pro*Pascal Precompiler has expired.

Action: Call customer support for assistance.

PCC-01401

PCC-01401   **Your Pro*Pascal authorization is about to expire**

Cause: The authorization or license to run the Pro*Pascal Precompiler is aboutto expire.

Action: Call customer support for assistance.

PCC-01500

PCC-01500   **You are not authorized to run Pro*Ada**

Cause: The authorization or license to run the Pro*Ada Precompiler has expired.

Action: Call Oracle Customer Support for assistance.

PCC-01501

PCC-01501   **Your Pro*Ada authorization is about to expire**

Cause: The authorization or license to run the Pro*Ada Precompiler is about to expire.

Action: Call Oracle Customer Support for assistance.

PCC-01510

PCC-01510   **No package, procedure, or function statement seen in name**

Cause: Every Ada source file must have a package, procedure, or function statement. The precompiler did not find one.

Action: Add the appropriate statement(s) to the source file and rerun Pro*Ada.

PCC-01511

PCC-01511 **Identifier on line num in file name was too long for code generated infile name**

Cause: An identifier used in the host program is of a length that causes precompiler-generated code to exceed the length limitation on source lines.

Action: Use a shorter identifier or use the ORECLEN option to increase the allowed line length.

PCC-01512

PCC-01512   **Invalid EXEC SQL INCLUDE of file name found at line num in file name**

Cause: The EXEC SQL INCLUDE statement is not allowed in Pro*Ada.

Action: Embed required source text directly in the Pro*Ada file. Use the "with" context clause to include required packages.

PCC-01513

PCC-01513   **FIPS warning: Unsupported datatype in line num of file name**

Cause: An Oracle extension to the ANSI/ISO SQL standard was used. Specifically, a host variable was declared with a non-ANSI/ISO standard type.

Action: No action required. However, for ANSI/ISO compliance, use only ANSI/ISO standard types.

PCC-01515

PCC-01515   **FIPS warning: Undeclared host variable hostvar at line num in file name**

Cause: An Oracle extension to the ANSI/ISO SQL standard was used. Specifically, a host variable was declared outside of a Declare Section.

Action: No action required. However, for ANSI/ISO compliance, declare the host variable within a Declare Section.

02010-02400: Oracle Precompiler, Release 2.0 Messages

PCC-02010

PCC-02010   **Found end-of-file while scanning string literal**

Cause: A string in a SQL statement, which should be delimited by single quotation marks, was not terminated properly.

Action: Check that all strings are delimited.

PCC-02011

PCC-02011   Found identifier greater than 128 characters (truncated)

Cause: The precompiler found an identifier that was too long.

Action: Shorten the identifier. SQL identifiers should be limited to 18 characters.

PCC-02012

PCC-02012   **Did not find a matching quote for CHAR constant**

Cause: A character constant with a single quotation mark was not terminated.

Action: Terminate the character constant.

PCC-02013

PCC-02013   **Unknown escape sequence**

Cause: The precompiler found an escape sequence that it could not process inside a string literal. This error can arise when multibyte character strings that can contain shift-in or shift-out escape sequences are used.

Action: Correct the string representation.

PCC-02014

PCC-02014   **Syntax error at line *num*, column *colnam*, file *name***

Cause: The precompiler detected an error in C or embedded SQL syntax. This message is followed by a more specific error message.

Action: Correct the syntax error.

PCC-02015

PCC-02015   **Unable to open include file**

Cause: The precompiler could not open a header file that was specified using the **#include** preprocessor directive or the EXEC SQL INCLUDE statement. This can happen if the file specification is inaccurate or if read permission or read access rights on the file or on one or more of the directories in the path werenot granted.

Action: Recheck that the file specification is correct, that the file exists, and that read permission have been granted so that the file can be read.

PCC-02016

PCC-02016   **Include file pathname is too long**

Cause: The pathname for include files exceeded the maximum length of 80 characters.

Action: Move the include file, or create an environment variable or logical to shorten the pathname.

PCC-02017

PCC-02017   **Unable to open output file**

Cause: The precompiler could not open an output file. This could be a generated code output file (*.c* file) or a listing file. This message can result from many causes. For example:

- A pathname for a specified output file contains a non-existent directory.
- An operating-system error occurred because the file system or diskis full.
- Write permission on the specified directory or directories in the path do not exist.

Action: Track down the cause of the error, as suggested above, and correct it.

PCC-02018

PCC-02018   **Found end-of-file while scanning comment**

Cause: A C comment, either in C code or in an embedded SQL statement, was not terminated.

Action: Find the unterminated comment and terminate it with */.

PCC-02019

PCC-02019   **Preprocessor warning at line *num*, column *colnam*, file *name***

Cause: This message indicates that a warning condition occurred as the precompiler was doing the preprocessor pass. A more specific warning message will follow this message.

Action: Correct the condition according to the action specified for the messagethat follows.

PCC-02020

PCC-02020 **Preprocessor error at line *num*, column *colnam*, file *name***

Cause: This message indicates that an error condition occurred as the precompilerwas in the preprocessing phase. A more specific error message will followthis message.

Action: Correct the error according to the action specified for the message that follows.

PCC-02021

PCC-02021   **Found newline while scanning string literal**

Cause: A string constant contains a newline character. For example,

```
char x[] = "Hello
   world";
```

Action: Remove the newline character.

PCC-02022

PCC-02022   **Found end of file while scanning a SQL statement**

Cause: The precompiler encountered an end of file while parsing an exec sql statement.

Action: Add the statement terminator(;) or complete the exec sql statement.

PCC-02023

PCC-02023   **Found end of file while scanning a PL/SQL statement**

Cause: The precompiler encountered an end of file while parsing a PL/SQL statement (EXEC SQL EXECUTE ...).

Action: Complete the PL/SQL statement.

PCC-02024

PCC-02024   **NCHAR literals are not permitted inside PL/SQL blocks**

Cause: The precompiler found a multi-byte character string in an embedded PL/SQL statement. For example, the offending code might look like

```
EXEC SQL execute
    declare
        name char(10) := N'Matsuda';
    BEGIN
        ....
    END;
END-EXEC;
```

Action: Remove multi-byte character string from the PL/SQL block or rewrite the declaration without using NLS_CHAR.

PCC-02025

PCC-02025   **NLS_LOCAL should be TRUE when using NCHAR literals**

Cause: The precompiler detected a multi-byte character host variable(NCHAR literal) when the option NLS_LOCAL was not set to TRUE on the command line. When NLS_LOCAL=TRUE, the runtime library (SQLLIB) performs blank-padding and blank-stripping for host variables that are of multi-byte types.

Action: Specify NLS_LOCAL=TRUE on the command line.

PCC-02035

PCC-02035   **CMD-LINE: CONFIG= option is invalid in a config file**

Cause: A user configuration file inside a user configuration file cannot be specified. That is, nested configuration files cannot be nested.

Action: If there are nested configuration files, move the options from the nested files to the top-level file.

PCC-02040

PCC-02040   **CMD-LINE:**   **Option selected, but no value provided**

Cause: An option on the command line was specified, but a value for it was not included. For example, the offending code might look like:

```
proc ina=sample1 oname=
```

could trigger this error message.

Action: Provide a value for the option.

PCC-02041

PCC-02041   CMD-LINE: Option does not exist:

Cause: A non-existent option on the command line was specified.

Action: See the *Programmer's Guide to the Oracle Precompilers* for a list of the valid command-line options and their possible values.

PCC-02043

PCC-02043 **CMD-LINE:** **Option syntax is incorrect**

Cause: A value for a command-line option was incorrectly specified.

Action: See the *Programmer's Guide to the Oracle Precompilers* for the correct syntax for specifying command-line option values.

PCC-02044

PCC-02044   **CMD-LINE:**   **Illegal or out of range value for option:** *option string*

Cause: A value specified for a command-line option was not within the accepted range. For example, the range for the MAXOPENCURSORS option is 5 to 256. If a value outside this range is specified, this message is triggered.

Action: See the *Programmer's Guide to the Oracle Precompilers* for the ranges that option values can take.

PCC-02045

PCC-02045   **CMD-LINE:**   **Option is not unique:** *option name*

Cause: An option name was partially specified on the command line that made it non unique. For example, in the command line

```
proc in=t.pc
```

the "in" option can imply either the INAME or INCLUDE option.

Action: Provide sufficient characters on the command line to make the optionname unique.

PCC-02046

PCC-02046   **CMD-LINE:**   **Unable to open config file:** *filename*

Cause: A non-existent user configuration file was specified on the command line with the config option.

Action: Provide a valid filename for the configuration file.

PCC-02047

PCC-02047   **CMD-LINE: Option is not allowed inline:** *option string*

Cause: An option was entered inline that can only be entered on the command line or in a configuration file. For example, the offending code might look like

```
EXEC ORACLE OPTION (NLS_CHAR=<name>);
```

The NLS_CHAR option can only be entered on the command line or in a configuration file.

Action: Remove the option from the source file, and specify it on the command line.

PCC-02066

PCC-02066   **CMD-LINE: Could not find or could not open system config file**

Cause: The system configuration file has a standard name (*pmscfg.h*) and a location that is operating-system dependent. On UNIX systems, it is located in the *$ORACLE_HOME/proc* directory. If a file named *pmscfg.h* is not found in the standard location, this warning message is issued.

Action: Create a system configuration file in the standard location. The file can be empty. See also the operating system-specific Oracle documentation.

PCC-02100

PCC-02100   **Unable to initialize PL/SQL**

Cause: The precompiler connected to Oracle but could not invoke the PL/SQL engine. This error can result if an earlier release of Oracle7 is used without the Procedural Option

Action: To use PL/SQL, upgrade to a more recent release of Oracle7.

PCC-02101

PCC-02101   **Unable to open input file**

Cause: The precompiler could not open the input file. This is the *.pc* file specified in the INAME= option. This means that the file does not exist, that a directory was incorrectly specified in the pathname, or that the person running the precompiler does not have read permission for the file.

    This message could also result from operating-system errors. For example, an unmounted file system or disk I/O errors, could trigger this error.

Action: Check that permission to read the file exists and that the full pathname has been specified correctly. Check for operating system-specific problems. See also the operating system-specific Oracle documentation.

PCC-02102

PCC-02102   **Fatal error while doing C preprocessing**

Cause: The precompiler issues this message after a more specific message.

Action: Correct the problem specified in the previous message or messages.

PCC-02103

PCC-02103   **Please enter your password:**

Cause: The username was specified on the command line without a password.For example:

```
proc sqlcheck=full iname=sample1.pc userid=scott/
```

Action: Re-enter the command line, and include the password.

PCC-02104

PCC-02104   **Unable to connect to Oracle**

Cause: The precompiler could not connect to Oracle with the username, password, and, if applicable, database link that was supplied. Note that this is a precompiler message, not a run time message. This means that the USERID option value, either on the command line or in a configuration file,was incorrect.

Action: Check that the username and password are current and correct. Run SQL*DBA or SQL*Plus and verify that connection can be made using that usernameand password.

PCC-02105

PCC-02105   Unable to open list file

Cause: The precompiler could not open the listing file. This message can result from many causes. For example:

- A pathname for a specified listing file contains a non-existent directory.
- An operating-system error occurred because the file system or diskis full.
- Write permission on the specified directory has not been granted.

Action: Track down the cause of the error, as suggested above, and correct it.

PCC-02106

PCC-02106   **Userid only used when SQLCHECK=FULL, userid ignored**

Cause: The USERID option was specified on the command line, but SQLCHECK was not equal to FULL or SEMANTICS. The USERID has no effect, unless SQLCHECK=FULL or SQLCHECK=SEMANTICS.

Action: This is a warning message only.

PCC-02107

PCC-02107   You may not specify PARSE = FULL when CODE = CPP

Cause: The PARSE=FULL and CODE=CPP options were both specified on the command line. The PARSE=FULL option invokes the C parser, which doesnot understand any C++ constructs generated by the precompiler with CODE=CPP option.

Action: Set the PARSE option to either NONE or PARTIAL if the CODE=CPP optionis specified.

PCC-02129

PCC-02129   **CMD-LINE: Client supplied static options table is invalid**

Cause: This is an internal error message not usually issued.

Action: Call Oracle customer support.

PCC-02132

PCC-02132   **CMD-LINE: Could not allocate memory**

Cause: This is an internal error message not usually issued.

Action: Call Oracle customer support.

PCC-02133

PCC-02133   **CMD-LINE:**   **Error processing string function**

Cause: This is an internal error message not usually issued. It indicates that a C string function, such as *strcpy* or *strlen*, returned an error.

Action: Call Oracle customer support.

PCC-02134

PCC-02134   CMD-LINE: Null option supplied

Cause: A zero length option was specified on the command line.

Action: Re-enter the command line with a valid option.

PCC-02135

PCC-02135   **CMD-LINE:**   **User asked for help**

Cause: This is a final message that the precompiler issues when information about the command-line options has been requested. For example, if the command

```
proc ?
```

is issued to get a list of the current default values for the command-line options, this message appears at the end of the list.

Action: No action required.

PCC-02200

PCC-02200   **Found unrecognized punctuation sequence**

Cause: This error indicates that the precompiler parser encountered a badly-formed identifier or keyword

Action: Correct the syntax.

PCC-02201

PCC-02201   **Found syntax error**

Cause: This general message precedes one or more specific messages that detail the nature of the error.

Action: No action required. Correct the errors diagnosed in the following message(s).

PCC-02202

PCC-02202   No TYPEDEF name was given

Cause: The precompiler parser encountered a TYPEDEF statement that had no name after the type specification. For example

```
typedef int;
```

Action: Correct the syntax.

PCC-02203

PCC-02203   **Found end of file when not expected**

Cause: The parser can emit this message when a general syntax error occurs, for example, an unmatched `{' or `('.

Action: Correct the syntax.

PCC-02204

PCC-02204   **EXEC SQL INCLUDE not permitted from within an included file**

Cause: EXEC SQL INCLUDE statements cannot be nested. Also, an EXEC SQL INCLUDE statement cannot be put inside a file that is included using the **#include** preprocessor command.

Action: Recode the program so that the nested include statement is not required.

PCC-02205

PCC-02205   **Parser error at line *num*, column *colnam*, file *name***

Cause: The precompiler parser encountered a syntax error, either in C code or in SQL code. A more specific message should follow.

Action: No action required for this message. Take the appropriate action for any following messages.

PCC-02206

PCC-02206   **Host variables are not permitted within a DDL statement**

Cause: A Data Definition Language statement cannot use host variables. For example, the statement

```
CREATE TABLE :table_name (c1 char(10));
```

is illegal, because the name of the table in a CREATE TABLE statement cannot be represented using a host variable.

Action: Use dynamic SQL to create the names of database objects (tables, views, columns, etc.) at runtime. See the *Programmer's Guide to the Oracle Precompilers* for information about dynamic SQL.

PCC-02207

PCC-02207 **Invalid macro name**

Cause: The precompiler parser encountered a #DEFINE directive that had no macro name associated with it. For example,

`#define`

Action: Correct the syntax.

PCC-02208

PCC-02208   **No filename specified in #include statement**

Cause: The precompiler parser encountered a #INCLUDE directive that had no filename associated with it. For example,

`#include`

Action: Specify appropriate filename with the #INCLUDE directive, or correctthe syntax.

PCC-02209

PCC-02209   **Macro invocation has incorrect number of arguments**

Cause: A macro invocation in the source does not have the same number of arguments as the macro definition in the #DEFINE line.

Action: Correct the macro reference or the macro definition.

PCC-02210

PCC-02210   A C++ punctuation sequence was found

Cause: C++ punctuation sequences are not supported by the Pro*C/C++ precompiler.

Action: Rewrite your C++ code without using punctuation sequences and precompileit again.

PCC-02301

PCC-02301   **Cannot open input file for reading**

Cause: The semantic analysis phase of the precompiler could not re-open the input file to generate output code.

Action: Check that the input file and its directory's protections and privileges do not change during precompilation.

PCC-02302

PCC-02302   **Cannot open code generation output file**

Cause: The precompiler was not able to open one or both of the temporary files that are required for code generation. The user executing the precompiler must have write permission (and/or the appropriate privileges) on the current directory.

Action: Check that permission exists to create files in the current directory.

PCC-02303

PCC-02303   **Cannot open include file**

Cause: The precompiler was not able to open a header file specified using the#INCLUDE preprocessor directive or the EXEC SQL INCLUDE statement. This can happen if the file specification is inaccurate or if read permission or read-access rights on the file or on one or more of the directories in the path have not been granted.

Action: Recheck that the file specification is correct, that the file exists, and that read permission has been granted so that the file can be read.

PCC-02304

PCC-02304   **Invalid declaration of C function**

Cause: A C function using invalid syntax was declared.

Action: Consult a C programming manual and correct the declaration.

PCC-02305

PCC-02305	**Illegal mixing of new and old style C function declarations**

Cause: All C functions using either the traditional style (K&R style) function declarations or the new (ANSI) style must be declared; the two styles cannotbe mixed.

Action: Adopt a consist declaration style for functions.

PCC-02306

PCC-02306   **Illegal name of C function**

Cause: A C function was declared with a name that is not a legal C identifier.

Action: Use legal C identifiers for all function names.

PCC-02307

PCC-02307   **Void can only be used when single parameter**

Cause: A function can be declared or defined using the following syntax:

```
int func1(void)
```

to indicate that the function has no parameters. **void** can be used only once in this case.

Action: Remove the extra **void**s in the function definition or declaration.

PCC-02308

PCC-02308   **Identifier required in this function definition**

Cause: A function definition written in ANSI C must have both the name and the type for all parameters.

Action: Rewrite the function definition so that it contains the name of each parameter.

PCC-02309

PCC-02309   **Illegal formal parameter declaration**

Cause: A formal parameter in a function declaration was specified without givingits type.

Action: Include the types of all parameters in function declarations.

PCC-02310

PCC-02310   **Formal parameter VARCHARs must be declared as pointers**

Cause: Many C compilers allow structures to be passed to and returned from functions. Although a VARCHAR is implemented as a C **struct**, VARCHARs must be passed to a function as pointers.

Action: Take the address of a VARCHAR when it is passed to a function. See the example in the section "Referencing VARCHAR Variables" in the *Programmer's Guide to the Oracle Precompilers.*

PCC-02311

PCC-02311   **Cannot have VARCHAR bit fields**

Cause: Host variables cannot contain bit fields.

Action: Recode the application to remove the bit fields.

PCC-02312

PCC-02312   **Arrays of VARCHAR greater than two dimensions not allowed**

Cause: A VARCHAR array with more than two dimensions was declared. Multidimensional arrays are not supported as host variables.

Action: Recode the application to remove multidimensional array usage.

PCC-02313

PCC-02313   **Malformed VARCHAR declaration - missing length**

Cause: When a VARCHAR is declared, a length specification is mandatory. For example, the following VARCHAR declaration is meaningless, hence illegal:

```
VARCHAR v1[];
```

Action: Specify the length for each declared VARCHAR.

PCC-02314

PCC-02314   **Cannot evaluate constant SIZEOF expression**

Cause: A SIZEOF operator was used where a precompiler expression was expected.

Action: Eliminate the SIZEOF operator.

PCC-02315

PCC-02315   **Cannot evaluate expression as a constant.**

Cause: The specified expression does not evaluate to a constant. Such expressions are required, for example, as the length of a VARCHAR.

Action: Replace the expression with one that does evaluate to a constant integer

PCC-02316

PCC-02316   **Illegal operator in constant expression**

Cause: A non-arithmetic operator was present in the expression.

Action: Rewrite the expression to eliminate the non-arithmetic operator.

PCC-02317

PCC-02317   **Illegal cast type expression**

Cause: A illegal cast is present in the expression.

Action: Remove the illegal cast.

PCC-02318

PCC-02318   Missing type expression

Cause: The specified expression is missing the declaration of a type.

Action: Specify a type for the expression.

PCC-02319

PCC-02319   **Expression type does not match usage**

Cause: The type of a variable does not match its usage. For example, in dynamic SQL, a host variable containing the text of a SQL statement must be declared as a C character type or be equivalenced to the SQL type STRING.

Action: Remove the declaration.

PCC-02320

PCC-02320   **Arithmetic expression does not have correct operand types**

Cause: The arithmetic expression must be specified with integral types.

Action: Rewrite the expression using integral types.

PCC-02321

PCC-02321   **Only subtraction between two pointers is permitted**

Cause: Pointer values cannot be added, multiplied, or divided.

Action: Recode to avoid this error message.

PCC-02322

PCC-02322   Found undefined identifier

Cause: An identifier used in a SQL statement was not defined. For example, a cursor name was referenced that had not been declared, or in a DECLARE CURSOR statement, a statement name was used that had not been PREPAREd.

Action: Check that all SQL identifiers, such as cursor names and statement names, have been defined before use.

PCC-02326

PCC-02326	**Illegal structure reference operation**

Cause: A structure component using invalid syntax was referenced. For example, a -> operator was used instead of a required `.' operator.

Action: Correct the invalid reference.

PCC-02327

PCC-02327   **Struct or struct pointer required**

Cause: A scalar host variable was used in a context where a structure (or its pointer)is required.

Action: Make the host variable a structure or add more scalar host variables to satisfy the SQL syntax requirements.

PCC-02328

PCC-02328   **Undefined struct member**

Cause: A structure component was referenced that was not declared as part ofthe structure.

Action: Redefine the structure.

PCC-02330

PCC-02330	Expecting an expression of integer type
Cause:	The expression does not evaluate to an integer. For example, a SQL FOR expression must evaluate to an integral type.
Action:	Rewrite the expression.

PCC-02331

PCC-02331   **Undefined SQL identifier**

Cause: All SQL identifiers must be defined before they are used. This message can result when a CURSOR or STATEMENT is not declared (defined) beforebeing referenced.

Action: Define the identifier.

PCC-02332

PCC-02332   Attempted to redefine SQL identifier

Cause: A SQL identifier (such as a cursor name) can be defined only once.

Action: Do not redefine the identifier. Use a different name.

PCC-02333

PCC-02333   **SQL identifier was not declared as a statement**

Cause: A SQL statement identifier was referenced in a DECLARE... CURSOR statement that was not PREPAREd.

Action: Check that all SQL statement names have been declared. Remember that SQL statement names are identifiers, not variables, and that they are local to the precompilation unit.

PCC-02334

PCC-02334   **SQL identifier was not declared as a cursor**

Cause: A cursor name was used in an OPEN, FETCH, or CLOSE statement that had not been DECLAREd.

Action: Check that all SQL cursor names have been declared. Remember that cursors are identifiers, not variables, and that they are local to the precompilation unit.

PCC-02335

PCC-02335   **Body of cursor must name statement identifier**

Cause: In a Dynamic SQL Method 4 application, the DECLARE... CURSOR statement must name a statement identifier that has been PREPAREd in a preceding statement. The PREPARE statement must physically (not logically) precede the DECLARE command.

Action: This error message is followed by another message that gives the line and column of the reference to the statement. Recode the application.

PCC-02336

PCC-02336   **Host variable expression has invalid type**

Cause: The host variable was declared using a C type that is not permitted as a host variable. See the *Programmer's Guide to the Oracle Precompilers* for a list of the permissible C types for host variables.

Action: Use a permissible host variable type

PCC-02337

PCC-02337   **Cannot declare multi-dimensioned array for this type**

Cause: Host variable arrays of scalars with more than 1 dimension cannot be declared.

Action: Recode the application because this kind of host variable usage is not possible with database operations.

PCC-02338

PCC-02338   Structure contains a nested struct or union

Cause: A host structure was used that contained a **struct** or **union**. This usage is not legal for Pro*C Release 2.0.

Action: This error message is followed by another message that gives the line and column of the reference to the illegal **struct**. Redefine the structure.

PCC-02339

PCC-02339   **Host variables cannot be of union types**

Cause: A C **union** as a host variable cannot be used.

Action: This error message is followed by another message that gives the line and column of the reference to the illegal **union**. Recode, using a **struct** or individual scalar host variables.

PCC-02340

PCC-02340   **Structure contains a bit field**

Cause: Bit fields are not allowed in host variables because they are meaningless for database DML operations.

Action: This error message is followed by another message that gives the line and column of the reference to the offending host variable. Recode the application.

PCC-02341

PCC-02341	Host variable has illegal type

Cause: A host variable cannot be declared using the **void** type.

Action: Rewrite the declaration, using a permissible host variable type.

PCC-02342

PCC-02342   **Using WHERE CURRENT OF on cursor defined without FORUPDATE clause**

Cause: When MODE=Oracle, a cursor defined with a WHERE CURRENT OF clause must also have a FOR UPDATE clause.

Action: Rewrite the cursor definition.

PCC-02343

PCC-02343   **Body of cursor must be a query expression**

Cause: A cursor definition must be a SELECT statement.

Action: Rewrite the cursor definition.

PCC-02344

PCC-02344   **Host variable arrays must all be of equal size**

Cause: This is a warning message. If host arrays are not of equal size, the size of the smallest array is used.

Action: No action required.

PCC-02345

PCC-02345 **SQLCHECK=SEMANTICS must be given when embedded PL/SQL blocks are used**

Cause: When a Pro*C program contains embedded PL/SQL blocks, the program must be precompiled using the SQLCHECK=SEMANTICS option, specified either on the command line or in a configuration file.

Action: Use the SQLCHECK=SEMANTICS option. This also requires that the USERID connect option be given.

PCC-02346

PCC-02346   **PL/SQL found semantic errors**

Cause: A database entity, such as a table or column name, was referenced, that does not exist. This is a compile time error, not a runtime error.

Action: Check that all referenced objects in PL/SQL statements actually exist and that the necessary access permission on them have been granted.

PCC-02347

PCC-02347   **PL/SQL found syntax errors**

Cause: A PL/SQL statement was used illegally.

Action: Check the *PL/SQL User's Guide and Reference* for the correct syntax.

PCC-02348

PCC-02348   **Indicators are not allowed in EXEC IAF statements**

Cause: Indicator variables associated with host variables cannot be used in EXEC IAF statements such as GET and PUT in a user exit.

Action: Eliminate the indicator variables. If feasible (for example with Forms V4), use EXEC TOOLS statements, which *do* allow indicator variables. See the *Programmer's Guide to the Oracle Precompilers* for more information of EXEC IAF and EXEC TOOLS statements.

PCC-02349

PCC-02349   **Precision must be specified for this type**

Cause: In a VAR or TYPE statement, certain Oracle types require that the precision be specified. For example, VARCHAR2 or CHAR.

Action: Indicate the precision.

PCC-02350

PCC-02350 **Cannot equivalence this SQL type**

Cause: Datatype or variable equivalencing to the datatypes NUMBER or DECIMAL cannot be used. See the "Datatype Equivalencing" section in the *Programmer's Guide to the Oracle Precompilers* for more information.

Action: Use the appropriate datatype for the application.

PCC-02351

PCC-02351	**Illegal datatype equivalencing operation**

Cause: The datatype specified could not be equivalenced, or the syntax was incorrect in the VAR or TYPE statement.

Action: See the *Programmer's Guide to the Oracle Precompilers* for a list of the datatypes that can be equivalenced and the correct syntax of the VAR or TYPE statement.

PCC-02352

PCC-02352   **Out of bounds bind position from PL/SQL**

Cause: This is an internal error.

Action: Call Oracle customer support for assistance.

PCC-02353

PCC-02353   **Semantic error at line *num*, column *colnam*, file *name***

Cause: This error message precedes a more specific error message.

Action: No action required. Correct the error(s) reported in the following message(s).

PCC-02354

PCC-02354   **A file included with #INCLUDE cannot contain SQL statements**

Cause: The Pro*C Precompiler reads header files referenced in #INCLUDE directives and uses the values defined in them. But the precompiler never generates code using statements in header files, so use of SQL statements in these filesis illegal.

Action: Move the SQL statement(s) into the main body of the application or use EXEC SQL INCLUDE to check that the included file is precompiled.

PCC-02355

PCC-02355   **Invalid or obsolete option, ignored**

Cause: A command-line option that is not used in Pro*C Release 2.0 was specified. For example, the option AREASIZE is no longer valid with precompilers used with the Oracle7 Server.

Action: No action required. This is an informational message.

PCC-02356

PCC-02356   **Warning at line *num*, column *colnam*, file *name***

Cause: This is a generic warning message. It precedes a more specific warning.

Action: No action required. Correct the error(s) indicated in the following messages.

PCC-02357

PCC-02357   **Function calls may not be used as host variable expressions**

Cause: Only objects that are *lvalues* (that resolve to an address) can be host variables. Because a function *call* is not an lvalue, one cannot be used in place of ahost variable.

Action: Recode the application.

PCC-02358

PCC-02358 **Identifier following ARRAYLEN must be the name of an array**

Cause: A declared array must be specified as the argument for an ARRAYLEN statement. Declare the array textually *before* the ARRAYLEN statement. Seethe *Programmer's Guide to the Oracle Precompilers* for the syntax of the ARRAYLEN statement.

Action: Correct the statement, specifying a host array as the argument.

PCC-02359

PCC-02359   **Identifier specifying ARRAYLEN dimension must be an integer**

Cause: The expression that specifies the ARRAYLEN dimension must evaluate to an integer. For example, the statement

```
EXEC SQL ARRAYLEN my_array(1.3)
```

cannot be parsed.

Action: Correct the statement, using an integral dimension.

PCC-02360

PCC-02360 **This array type is invalid for use with ARRAYLEN statement**

Cause: Arrays of some host variables types are not allowed and hence also cannot be used in ARRAYLEN statements. VARCHAR is an example.

Action: Specify an array with a datatype that can be used in host arrays.

PCC-02361

PCC-02361   **Use of ARRAYLEN with SQL host arrays is ignored**

Cause: ARRAYLEN is only valid with arrays that can be bound in PL/SQL blocks.

Action: Use an appropriate array type.

PCC-02362

PCC-02362   **Host variable not declared within SQL Declare Section**

Cause: When MODE=ANSI is specified at precompile time, all host variables must be declared inside Declare Sections. Remember that MODE=ANSI refers to ANSI SQL, not ANSI C.

Action: Add the EXEC SQL BEGIN DECLARE SECTION... EXEC SQL END DECLARE SECTION statements around all host variable declarations.

PCC-02363

PCC-02363   **Indicator variable must be a structure**

Cause: If the host variable is a structure, the associated indicator variable must also be a structure.

Action: Recode the application.

PCC-02364

PCC-02364   **Host structure and its indicator must have the same number of fields**

Cause: When a structure containing indicator variables is declared and associated with a host structure, the indicator structure must contain the same number of fields as the host structure. This is so even when some of the indicators will not be used or even when it would not make sense to do so (for fields constrained as NON NULL, for example).

Action: Redeclare the indicator structure.

PCC-02365

PCC-02365   **Host and its indicator variable must have the same array dimensions**

Cause: When associating an array of indicators with a host array, it is good practiceto make the indicator array the same length as the host array. It should neverbe shorter.

Action: This is a warning message. The smallest array size will determine the number of items fetched, inserted, or updated.

PCC-02366

PCC-02366   **Command line argument MODE=ANSI may not be used with DBMS=V6**

Cause: The semantics of certain operations (such as character comparison) in Oracle Version 6 are not 100% compliant with the ANSI/ISO SQL standards. When V6 semantics are requested, using the DBMS=V6 or DBMS=V6_CHAR option, precompilation with MODE=ANSI is not permitted.

Action: Do not use ANSI mode with the V6 options.

PCC-02367

PCC-02367   **Indicator variables must be declared as type short**

Cause: The only C type allowed for indicator variables is **short.** An array of indicator variables must be an array of **shorts.**

Action: Redeclare the indicator variables and/or indicator arrays.

PCC-02368

PCC-02368   An EXEC TOOLS host variable context name is not type char

Cause: If a host variable is used to define a context name in an EXEC TOOLS GET CONTEXT or EXEC TOOLS SET CONTEXT statement, the host variable must be a character type.

Action: Redeclare the context name host variable.

PCC-02369

PCC-02369   **An EXEC TOOLS host pointer variable is not a pointer**

Cause: The host variable specifying an EXEC TOOLS context must be a pointer type

Action: Redeclare the variable, making sure that it is a pointer.

PCC-02370

PCC-02370 **An EXEC TOOLS message host variable is not type char**

Cause: If a host variable is used to define a context name in an EXEC TOOLS GET CONTEXT or EXEC TOOLS SET CONTEXT statement, the host variable must be a character type.

Action: Redeclare the context name host variable.

PCC-02371

PCC-02371   **Illegal FOR clause**

Cause: The argument of a FOR clause must be specified as an integer or as an identifier containing an integer.

Action: Rewrite the FOR clause.

PCC-02372

PCC-02372   **FOR clause not allowed in a SELECT statement**

Cause: A SQL statement containing a SELECT command cannot contain a FOR clause. The meaning of such a statement would be unclear.

Action: Rephrase the SELECT statement, removing the FOR clause.

PCC-02373

PCC-02373   **Invalid declaration in EXEC SQL DECLARE section**

Cause: An improper declaration was placed in a Declare Section. This message is usually caused by including an EXEC SQL TYPE or EXEC SQL VAR declaration in a Declare Section.

Action: Move the declaration outside the Declare Section.

PCC-02374

PCC-02374   **SQLCHECK value exceeds command line value**

Cause: The value assigned to the SQLCHECK option in an EXEC Oracle statement in a program was greater than the value given either on the command line or greater than the default value if no SQLCHECK option was given on the command line.

The order of the option values is SEMANTICS > SYNTAX > NONE.

When this warning message is issued, the original value of SQLCHECK (the default or the command-line value) stays in effect.

Action: Remove or recode the EXEC Oracle option(s) in the *.pc* source file to avoid this warning message.

PCC-02375

PCC-02375   **SQL statement found outside the body of a function**

Cause: A SQL statement other than a declarative, datatype equivalence or WHENEVER statement was found outside the body of a function when PARSE=FULL.

Action: Move the SQL statement inside the body of a function.

PCC-02376

PCC-02376   **You may not declare SQLCODE when DEF_SQLCODE = TRUE**

Cause: The DEF_SQLCODE option may not be specified if a SQLCODE declaration already explicitly appears in the program.

Action: Remove the SQLCODE declaration from the program or specify DEF_SQLCODE=NO (the default).

PCC-02377

PCC-02377   **Arrays of implicit VARCHARs are not permitted**

Cause: An array of implicit VARCHARs was declared when HOST_VARCHAR=TRUE.

Action: Rewrite the implicit VARCHAR declaration without using arrays.

PCC-02378

PCC-02378   **Invalid type for National Language character variable**

Cause: A variable declared as a National Language character using the NLS_CHAR option was not declared as a char or implicit VARCHAR.

Action: Declare the variable as a char or implicit VARCHAR or do not specify using the NLS_CHAR option.

PCC-02379

PCC-02379   **Cannot equivalence National Language character variables**

Cause: A National Language character variable (specified using the NLS_CHAR option) has undergone datatype equivalencing using either the EXEC SQL VAR or TYPE statements.

Action: Do not equivalence National Language character variables.

PCC-02380

PCC-02380   NLS_LOCAL should be TRUE when using NLS_CHAR

Cause: NLS_CHAR was used to specify National Language character variables without specifying NLS_LOCAL=TRUE.

Action: Specify NLS_LOCAL=TRUE. Note that in future releases, this will nolonger apply.

PCC-02381

PCC-02381   **Cannot use National Language character variables inside PL/SQL**

Cause: A variable or fixed length National Language character variable was used in an embedded PL/SQL block.

Action: Remove the variable from the block or use another one having a legalbind type.

PCC-02382

PCC-02382   **You must specify MODE = ANSI when using NLS_CHAR**

Cause: NLS_CHAR was used to specify NLS multi-byte character variables without specifying MODE=ANSI.

Action: Specify MODE=ANSI on the command line.

PCC-02383

PCC-02383   **Arrays of National Language variables are not permitted**

Cause: A variable used as a bind or define variable declared using NLS_CHAR was declared as an array of National Language character strings.

Action: Rewrite the declaration without using arrays or choose not to specify using NLS_CHAR.

PCC-02384

PCC-02384	Missing array length specifier

Cause: An array declaration was given without an array length specifier.

Action: Provide the length for the array declaration.

PCC-02385

PCC-02385   You may not specify DBMS = V6_CHAR when using NLS_CHAR

Cause: The DBMS=V6_CHAR option was specified on the command line while indicating which host variables are to be treated by the precompiler as NLS multi-byte character variables. A multi-byte character string must be null-terminated, but if the DBMS=V6_CHAR option is specified, the precompiler pads the character arrays with blanks up to the length of the array without the null-terminator.

Action: Remove the DBMS=V6_CHAR option from the command line.

PCC-02386

PCC-02386   **Use of the AT clause is not permitted here**

Cause: An explicit AT clause was used with an ALLOCATE statement or a FETCH/CLOSE statement also using a SQL_CURSOR declaration.

Action: Remove the AT clause.

PCC-02387

PCC-02387 **Expecting an expression of type SQL_CURSOR**

Cause: A host bind variable was used in an ALLOCATE, FETCH or CLOSE statement that was not declared as a result set SQL_CURSOR.

Action: Declare the variable to be of type SQL_CURSOR.

02100-02139: Oracle Runtime Library SQL Messages

The following error messages might be issued at run time by SQLLIB, the Oracle runtime library.

SQL-02100

SQL-02100   **Out of memory (i.e., could not allocate)**

Cause: SQLLIB was unable to allocate enough memory to execute the program.

Action: Allocate more memory to the user session, then rerun the program. If the error persists, call customer support for assistance.

SQL-02101

SQL-02101   **Inconsistent cursor cache. Unit cursor/global cursor mismatch**

Cause: This internal error typically indicates a memory-related error.

Action: Check the program for memory-related errors, such as invalid pointers or array-bounds violations.

SQL-02102

SQL-02102	Inconsistent cursor cache. No global cursor entry

Cause: This internal error typically indicates a memory-related error.

Action: Check the program for memory-related errors, such as invalid pointers or array-bounds violations.

SQL-02103

SQL-02103	**Inconsistent cursor cache. Out of range cursor cache reference**

Cause: This internal error typically indicates a memory-related error.

Action: Check the program for memory-related errors, such as invalid pointers or array-bounds violations.

SQL-02104

SQL-02104	Inconsistent host cache. No cursor cache available

Cause: This internal error typically indicates a memory-related error.

Action: Check the program for memory-related errors, such as invalid pointers or array-bounds violations.

SQL-02105

SQL-02105   **Inconsistent cursor cache. Global cursor not found**

Cause: This internal error typically indicates a memory-related error.

Action: Check the program for memory-related errors, such as invalid pointers or array-bounds violations.

SQL-02106

SQL-02106   **Inconsistent cursor cache. Invalid Oracle cursor number**

Cause: This internal error typically indicates a memory-related error.

Action: Check the program for memory-related errors, such as invalid pointers or array-bounds violations.

SQL-02107

SQL-02107   **Program too old for runtime library; please re-precompile it**

Cause: The program was precompiled by an older version of the Oracle Precompilers, which is incompatible with this
release of SQLLIB.

Action: Precompile the program with a newer version of the Oracle Precompilers.

SQL-02108

SQL-02108   **Invalid descriptor passed to run-time library**

Cause: This internal error typically indicates a memory-related error.

Action: Check the program for memory-related errors, such as invalid pointers or array-bounds violations.

SQL-02109

SQL-02109   **Inconsistent host cache. Host reference is out of range**

Cause: This internal error typically indicates a memory-related error.

Action: Check the program for memory-related errors, such as invalid pointers or array-bounds violations.

SQL-02110

SQL-02110   **Inconsistent host cache. Invalid host cache entry type**

Cause: This internal error typically indicates a memory-related error.

Action: Check the program for memory-related errors, such as invalid pointers or array-bounds violations.

SQL-02111

SQL-02111   **Heap consistency error**

Cause: This internal error typically indicates a memory-related error.

Action: Check the program for memory-related errors, such as invalid pointers or array-bounds violations.

SQL-02112

SQL-02112   **SELECT ... INTO returns too many rows**

Cause: A SELECT ... INTO statement returned more rows than can be stored in the host variable provided.

Action: There are three possible solutions:
- Use the precompiler option SELECT_ERROR=NO.
- Declare a host variable with a larger array dimension.
- Declare a cursor or cursor variable for the select statement.

SQL-02113

SQL-02113   **Unable to open message file**

Cause: SQLLIB was unable to find or open the file, *ora_sqllib*:*sqlli*.msb*.

Action: Check that *ora_sqllib* is properly defined, and that a *sqlli*.msb* file exists andis readable.

SQL-02114

SQL-02114   **Invalid SQL Cursor usage: trying to CLOSE a CLOSEd cursor**

Cause: An attempt was made to CLOSE a cursor that was already CLOSEd with MODE={ANSI|ANSI14}. A CLOSEd cursor can be re-CLOSEd only when MODE={ORACLE|ANSI13}.

Action: When MODE={ANSI|ANSI14}, verify that a cursor is not already CLOSEd before trying to CLOSE it. Specify MODE={ORACLE|ANSI13} if a CLOSEd cursor is to be re-CLOSEd.

SQL-02115

SQL-02115   **Code interpretation problem -- check COMMON_NAME usage**

Cause: With Pro*FORTRAN, this error occurs if the precompiler option COMMON_NAME is specified incorrectly. With other Oracle Precompilers, this error occurs when the precompiler cannot generate a segment of code.

Action: With Pro*FORTRAN, when using COMMON_NAME to precompile two or more source modules, make sure to specify a different common name for each module. With other Oracle Precompilers, if the error persists, call customer support for assistance.

SQL-02116

SQL-02116   **FATAL ERROR: Reentrant code generator gave invalid context**

Cause: This internal error typically indicates a memory-related error.

Action: Check the program for memory-related errors, such as invalid pointers or array-bounds violations.

SQL-02117

SQL-02117   **Invalid SQL Cursor usage: trying to OPEN an OPENed cursor**

Cause: An attempt was made to OPEN an cursor that was already OPENed with MODE={ANSI|ANSI14|ANSI13}. An open cursor can be re-OPENed only when MODE=ORACLE.

Action: When MODE={ANSI|ANSI14|ANSI13}, verify that a cursor is not already OPEN before trying to OPEN it. Specify MODE=ORACLE if an OPENed cursor is to be re-OPENed to avoid reparsing.

SQL-02118

SQL-02118 **Invalid row for a WHERE CURRENT OF operation**

Cause: An attempt was made to reference a nonexistent row using the CURRENT OF clause in an UPDATE or DELETE statement. This happens when no FETCH has been executed or when FETCH returns a "no data found" error that the program fails to trap.

Action: Check that the last cursor operation succeeded and that the current row of the cursor is valid. The outcome of a cursor operation can be checked in two ways: implicit checking with the WHENEVER statement or explicit checking of SQLCODE in the SQLCA.

SQL-02119

SQL-02119   **Invalid HSTDEF argument**

Cause: A non-null pointer was given for an HSTDEF as the second argument to a SQLRCN call when the first argument was also used. Either the first or the second argument to SQLRCN must be null.

Action: Pass a null pointer in either the first or second argument to SQLRCN.

SQL-02120

SQL-02120   **First and second arguments to SQLRCN both null**

Cause: Both the first and second arguments to SQLRCN were null. An HSTDEF must be passed into SQLRCN in the form of either an OCI LDA (first argument) or as an HSTDEF pointer itself (second argument).

Action: Pass either an OCI LDA or a HSTDEF, but not both.

SQL-02121

SQL-02121   **Invalid host name**

Cause: The host name passed into a SQLFCN call was not used in a previous callto SQLRCN.

Action: Use the same identifier used in the corresponding SQLRCN call.

SQL-02122

SQL-02122   **Invalid OPEN or PREPARE for this database connection**

Cause: An attempt was made to execute an OPEN or PREPARE statement using a cursor that is currently open for another database connection and, therefore, cannot be used for this connection.

Action: Close the cursor to make it available for this connection or use a different cursor for this connection.

SQL-02123

SQL-02123   **Context referenced in EXEC TOOLS GET CONTEXT statement not found**

Cause: The context name given in the EXEC TOOLS GET CONTEXT statement was never stored with an EXEC TOOLS SET CONTEXT statement.

Action: Use the EXEC TOOLS SET CONTEXT statement to save any contexts to be retrieved later.

SQL-02124

SQL-02124   **NULL value returned by EXEC TOOLS statement**

Cause: An EXEC TOOLS statement returned a null to a host variable that lacks an indicator variable. This error occurs only when MODE=ANSI. When MODE=ORACLE, although the value of the host variable is indeterminate, no error is generated.

Action: Associate an indicator variable with each host variable to which nulls mightbe returned.

SQL-02125

SQL-02125   **Connect error, can't get error text**

Cause: No connection (not even to the default host) was available, so SQLLIB could not get the message text for the Oracle error that occurred. However, SQLLIB returns the Oracle error number, which can be used to look up the message.

Action: Lookup up the appropriate message in *Oracle7 Server Messages* and follow the Cause and Action information provided.

SQL-02126

SQL-02126   **Number of array elements cannot be negative (i.e. < 0)**

Cause: The precompiler found a negative number of array elements in the N or F variable of the SQLDA (SQL Descriptor Area, which is used with dynamic SQL Method 4). Before executing the DESCRIBE statement, N must be set to the dimension of the descriptor arrays. After executing the DESCRIBE statement, N must be reset to the actual number of variables DESCRIBEd, which is stored in the F variable.

Action: Check that the N and F variables are set to non-negative values.

SQL-02127

SQL-02127   **Precompiler/SQLLIB version mismatch**

Cause: The program was linked to an older version of SQLLIB, which is incompatible with this release of the Oracle
Precompilers.

Action: Relink the program with the appropriate version of SQLLIB.

SQL-02128

SQL-02128   **Sessions still exist, not logged off**

Cause: A host was not removed ("logged off") from SQLLIB because XA still has sessions associated with that host.

Action: Before calling SQLXDH to drop a host, XA must either drop all sessions for that host or set the *doit_anyway* flag.

SQL-02129

SQL-02129   **FETCHed number of bytes is odd**

Cause: The program tried to FETCH an invalid column value into a multi-byte NLS host variable. Specifically, the column did not contain valid double-byte data.

Action: Make sure the column contains only double-byte data. To verify this, use SQL*Plus or Server Manager.

SQL-02130

SQL-02130   EXEC TOOLS interface is not available

Cause: An attempt was made to link with a version of an Oracle tool that does not support the EXEC TOOLS interface.

Action: Either upgrade the Oracle tool, or use the EXEC IAF interface.

00250-00300: PCF FIPS Messages

This section contains messages that are generated when you specify the precompiler command-line option FIPS=YES and use an Oracle extension to the ANSI/ISO embedded SQL standard. If you wish your code to comply with the standard, do not use the Oracle extension described in the message.

PCF-00251

PCF-00251   SQLCODE is a deprecated feature

PCF-00252

PCF-00252   use of the C preprocessor within SQL

PCF-00253

PCF-00253   use of the AT clause

PCF-00254

PCF-00254   use of the FOR clause

PCF-00255

PCF-00255   use of the CONNECT statement

PCF-00256

PCF-00256   SQLCODE returns non-standard values

PCF-00257

PCF-00257   keyword WORK required after COMMIT

PCF-00258

PCF-00258   keyword WORK required after ROLLBACK

PCF-00259

PCF-00259   use of RELEASE clause

PCF-00260

PCF-00260   use of dynamic SQL

PCF-00261

PCF-00261   use of PL/SQL

PCF-00262

PCF-00262   use of SQLWARN within WHENEVER clause

PCF-00263

PCF-00263   use of NOTFOUND within WHENEVER clause

PCF-00264

PCF-00264   use of NOT FOUND within WHENEVER clause

PCF-00265

PCF-00265   use of STOP within WHENEVER clause

PCF-00266

PCF-00266   use of RAISE within WHENEVER clause

PCF-00267

PCF-00267   use of DO within WHENEVER clause

PCF-00268

PCF-00268   use of DECLARE TABLE statement

PCF-00269

PCF-00269   use of DECLARE DATABASE statement

PCF-00270

PCF-00270   use of EXEC SQL INCLUDE

PCF-00271

PCF-00271   use of EXEC SQL TYPE

PCF-00272

PCF-00272   use of SQL FORMS statement

PCF-00273

PCF-00273   use of EXEC ORACLE statement

PCF-00274

PCF-00274   invalid datatype

PCF-00275

PCF-00275   use of DROP TABLE statement

PCF-00276

PCF-00276   use of non-ANSI function

PCF-00277

PCF-00277   use of non-ANSI function

PCF-00278

PCF-00278   use of the ALLOCATE statement

SQL*DBA Messages

This chapter lists messages generated by SQL*DBA. The likely cause(s) of the event(s) generating the message and recommended corrective action are also presented.

The messages listed in this chapter are displayed with the prefix DBA, signifying SQL*DBA, or LCC, signifying the command line. Refer to *Oracle7 Server Utilities* for more information on, and syntax for, SQL*DBA.

00001-00199: SQL*DBA Messages

DBA-00072

DBA-00072   **warning: *name* created with compilation errors**

Cause: The package, package body, procedure, or function could not be created because there were code errors.

Action: Enter the SHOW ERRORS command and correct the listed compilation errors.

DBA-00073

DBA-00073   **warning:** *name* **created with compilation errors**

Cause: The package, package body, procedure, or function could not be created because there were code errors.

Action: Enter the SHOW ERRORS command and correct the listed compilation errors.

DBA-00100

DBA-00100   **invalid SPOOL filename**

Cause: When using the SPOOL command, the name of a file that already existswas specified.

Action: Specify a valid name for a file that does not currently exist.

DBA-00101

DBA-00101   extraneous text at end of command

Cause: There were unrecognized commands or other text on the command line.

Action: Check the syntax of the command and issue the command again.

DBA-00102

DBA-00102   **missing LOG keyword**

Cause: The LOG keyword was missing from the ARCHIVE LOG command.

Action: Check the syntax of the command and issue the command again.

DBA-00103

DBA-00103   **illegal ARCHIVE LOG option**

Cause: An option specified was not LIST, STOP, START, NEXT, ALL, a number, or a filename in the ARCHIVE LOG command.

Action: Check the syntax of the command and issue the command again.

DBA-00104

DBA-00104   illegal RECOVER option

Cause: An option specified was not DATABASE, MANUAL, UNTIL, TABLESPACE, or DATAFILE in the RECOVER command.

Action: Check the syntax of the command and issue the command again.

DBA-00105

DBA-00105   **invalid INSTANCE name**

Cause: An invalid SQL*Net CONNECT string was specified for the instance name.

Action: Refer to the SQL*Net documentation for a complete description of SQL*Net CONNECT strings.

DBA-00106

DBA-00106   **invalid ECHO switch**

Cause: An invalid option for the SET ECHO command was specified.

Action: Use either ON or OFF as an option for the SET ECHO command.

DBA-00107

DBA-00107   invalid TERMOUT switch

Cause: An invalid option for the SET TERMOUT command was specified.

Action: Use either ON or OFF as an option for the SET TERMOUT command.

DBA-00108

DBA-00108   **invalid TIMING switch**

Cause: An invalid option for the SET TIMING command was specified.

Action: Use either ON or OFF as an option for the SET TIMING command.

DBA-00109

DBA-00109   **invalid CYCLE value**

Cause: An invalid value for the SET CYCLE command was specified.

Action: The SET CYCLE command must use an integer between 1 and 3600.

DBA-00110

DBA-00110   illegal SET option

Cause: An invalid option for the SET command was specified.

Action: Check the syntax of the SET command in *Oracle7 Server Utilities* and issue the command again.

DBA-00111

DBA-00111   **illegal SHOW option**

Cause: An invalid option for the SHOW command was specified.

Action: Check the syntax of the SHOW command in *Oracle7 Server Utilities* and issue the command again.

DBA-00113

DBA-00113   **invalid PFILE name**

Cause: The specified PFILE name is not recognized.

Action: Specify a legal PFILE name.

DBA-00114

DBA-00114 **invalid database name**

Cause: The specified database name is not recognized.

Action: Specify a legal database name.

DBA-00115

DBA-00115   **unexpected end of command**

Cause: An option was specified without any arguments. For example:

```
SQLDBA> STARTUP OPEN PFILE
```

Action: Check the syntax of the command and enter the command again with the appropriate arguments for options that require values.

DBA-00116

DBA-00116   **illegal SHUTDOWN option**

Cause: An option was specified that was not NORMAL, IMMEDIATE, or ABORT for the SHUTDOWN command.

Action: Check the syntax of the command and issue the command again.

DBA-00117

DBA-00117   **invalid tablespace name**

Cause: The specified tablespace name was not recognized.

Action: Specify a legal tablespace name.

DBA-00118

DBA-00118   **invalid process identifier or invalid statistics class**

Cause: Something other than a number for a process identifier, or an invalid class was specified after the MONITOR STATISTICS command.

Action: Specify only numbers as process identifiers or specify a valid class for the MONITOR STATISTICS command.

DBA-00119

DBA-00119   **illegal STARTUP option**

Cause: An option was specified that was not DBA, PFILE, EXCLUSIVE, SHARED, MOUNT, OPEN, RECOVER, or NOMOUNT for the STARTUP command.

Action: Check the syntax of the command and issue the command again.

DBA-00120

DBA-00120   **invalid STOPONERROR switch**

Cause: An invalid option was specified for the SET STOPONERROR command.

Action: Use either ON or OFF as an option for the SET STOPONERROR command.

DBA-00121

DBA-00121   illegal MONITOR option

Cause: An illegal option was specified for the MONITOR command.

Action: Check the syntax of the MONITOR command in *Oracle7 Server Utilities* and issue the command again.

DBA-00122

DBA-00122   **invalid SET numeric parameter**

Cause: A character or an invalid value was specified when a number was expected as a value for a SET command option.

Action: Check the syntax of the command, use an appropriate number for the option, and enter the command again.

DBA-00123

DBA-00123   invalid tablespace name list

Cause: An invalid string was specified as one of the values in the list of tablespaces in the ARCHIVE LOG command. For example, "" is an invalid string.

Action: Check the syntax of the command, use valid tablespace names for all names in the list, and enter the command again.

DBA-00124

DBA-00124   **invalid ARCHIVE destination**

Cause: The specified ARCHIVE destination was not recognized.

Action: Specify a legal ARCHIVE destination.

DBA-00125

DBA-00125   integer value overflow

Cause: A numeric value was specified that was too large.

Action: Use a smaller number.

DBA-00126

DBA-00126   **invalid VERIFY switch**

Cause: The VERIFY switch was not recognized.

Action: Specify a legal VERIFY switch.

DBA-00127

DBA-00127   **invalid combination of STARTUP options**

Cause: The specified options of the STARTUP command cannot be used simultaneously. For example, the following command is not valid:

```
SQLDBA> STARTUP MOUNT OPEN testdb
```

Action: Check the syntax of the STARTUP command in *Oracle7 Server Utilities* for options that can be used together.

DBA-00128

DBA-00128   invalid DEBUG switch

Cause: The DEBUG switch was not recognized.

Action: Specify a legal DEBUG switch.

DBA-00129

DBA-00129   **value out of range (1 - *num*)**

Cause: The specified value was out of range. The valid range is given by theerror message.

Action: Use a number within the range specified by this error.

DBA-00131

DBA-00131   **invalid ARCHIVE TO device**

Cause: The specified ARCHIVE TO device was not recognized.

Action: Specify a legal ARCHIVE TO device.

DBA-00132

DBA-00132   null *hostname/password* specified

Cause: *hostname/password* was not specified.

Action: Specify the correct *hostname/password*.

DBA-00133

DBA-00133   invalid datafile list

Cause: An invalid string was entered as one of the values in the list of datafiles in the ARCHIVE LOG command. For example, "" is an invalid string.

Action: Check the syntax of the command, use valid datafile names for all names in the list, and enter the command again.

DBA-00134

DBA-00134   invalid AUTORECOVERY switch

Cause: An invalid option was specified for the SET AUTORECOVERY command.

Action: Use either ON or OFF as options for the SET AUTORECOVERY command.

DBA-00136

DBA-00136   **bad variable specification**

Cause: A variable was incorrectly specified using the VARIABLE command.

Action: Check the syntax of the command and issue the command again.

DBA-00137

DBA-00137   **syntax error in PL/SQL Block**

Cause: The PL/SQL block contains a syntax error.

Action: Correct the syntax error.

DBA-00139

DBA-00139   **invalid ALTER DATABASE option**

Cause: The specified ALTER DATABASE option was not recognized.

Action: Specify a legal option.

DBA-00140

DBA-00140   **invalid COMPATIBILITY switch**

Cause: The specified COMPATIBILITY switch was not recognized.

Action: Specify a legal COMPATIBILITY switch.

DBA-00141

DBA-00141   invalid RETRIES value

Cause: The specified RETRIES value was not recognized.

Action: Specify a legal RETRIES value.

DBA-00142

DBA-00142   **cannot recognize object type, owner or name**

Cause: The specified object type, owner, or name was not recognized.

Action: Specify a legal object type, owner or name.

DBA-00143

DBA-00143   **variable has not been defined**

Cause: The specified variable was not recognized.

Action: Specify an existing variable.

DBA-00144

DBA-00144   invalid object type for DESCRIBE

Cause: The specified object type was not TABLE, VIEW, or PROCEDURE.

Action: Check that the object is a table, view, or procedure. If so, check that you specified the correct name and try again. If not, you cannot DESCRIBEthe object.

DBA-00145

DBA-00145   **invalid object name for DESCRIBE**

Cause: The specified table, view, stored procedure or function was not recognized.

Action: Check spelling and specify an existing table, view, stored procedure,or function.

DBA-00146

DBA-00146   invalid HISTORY value

Cause: The specified HISTORY value was not recognized.

Action: Specify a legal HISTORY value.

DBA-00147

DBA-00147   **invalid LINES value**

Cause: The specified LINES value was not recognized.

Action: Specify a legal LINES value.

DBA-00148

DBA-00148   **invalid TERM switch (use either PAGE or NOPAGE)**

Cause: The specified value was not recognized.

Action: Specify a legal value.

DBA-00149

DBA-00149   **invalid SERVEROUTPUT switch**

Cause: The specified value was not recognized.

Action: Specify a legal value.

DBA-00150

DBA-00150   **invalid FLAGGER switch**

Cause: The specified value was not recognized.

Action: Specify a legal value.

00300-00359: DBA Execution

DBA-00300

DBA-00300   **internal error code, arguments: [*num*], [*str*]**

Cause: An internal error was encountered.

Action: Contact customer support with the circumstances leading to the error and the complete set of error messages.

DBA-00301

DBA-00301   **cannot SET INSTANCE while connected to a database**

Cause: SET INSTANCE was used while currently connected to a database.

Action: To change the current instance, DISCONNECT from the database before using SET INSTANCE.

DBA-00302

DBA-00302   **not connected to a database**

Cause: The requested operation required being connected to the database.

Action: CONNECT to the database using a valid username and password before retrying the operation.

DBA-00304

DBA-00304   **input file I/O error *num* - input aborted**

Cause: A command file used as input to SQL*DBA is corrupt or invalid.

Action: Check the file before retrying the operation.

DBA-00305

DBA-00305 **command size exceeds internal buffer size** *num*

Cause: The size of the SQL statement exceeds SQL*DBA's buffer size.

Action: Shorten the SQL statement by removing extra blanks or by using intermediate statements or views, if necessary.

DBA-00306

DBA-00306   monitor cycle interval time out of range (1 - *num*)

Cause: An invalid number was specified for the cycle interval.

Action: Enter a number between 1 and 3600 for the cycle interval. The numberindicates seconds.

DBA-00307

DBA-00307   **cannot open spool file** *name*

Cause: SQL*DBA tried to open a spool file after the SPOOL filename was specified, but could not open the file. Possible causes are not enough disk space or inadequate privileges to create a file.

Action: Determine why SQL*DBA could not create a new file and retry.

DBA-00308

DBA-00308	**no spool file opened**

Cause: SPOOL OFF was entered when not currently spooling.

Action: To capture session output, use the SPOOL command to open a file, enter the required commands, and close the file with SPOOL OFF.

DBA-00309

DBA-00309	**cannot close spool file** *name*

Cause: SPOOL OFF could not close the currently open spool file.

Action: Check for an operating system reason that the spool file could not be closed.

DBA-00310

DBA-00310   **cannot open parameter file** *name*

Cause: SQL*DBA cannot locate or open the file specified by the PFILE option, because either the file does not exist or SQL*DBA has insufficient privileges to openthe file.

Action: Check that the file exists in a location expected by SQL*DBA and canbe opened.

DBA-00311

DBA-00311   **data size exceeds internal buffer size**

Cause: The results returned by a SQL query exceed the internal SQL*DBA buffer.

Action: Use the SET command to increase MAXDATA or decrease ARRAYSIZE.

DBA-00314

DBA-00314   **unable to attach to default host**

Cause: The default host has not been set up correctly.

Action: Use the SET INSTANCE command with a valid host specification to set up a default host.

DBA-00315

DBA-00315   **cannot open command file** *name*

Cause: SQL*DBA cannot locate the specified command file.

Action: Check the file's name and SQL*DBA's access to it before retrying.

DBA-00316

DBA-00316   **cannot start up while connected to a database**

Cause: An attempt was made to start up an instance while connected to a database.

Action: To start up an instance, first DISCONNECT from the current database.

DBA-00318

DBA-00318   **SQL*DBA command line error** *num*

Cause: A syntax or typing error was made while entering a SQL*DBA command line.

Action: Check the syntax and try again.

DBA-00319

DBA-00319   **cannot locate sqldba configuration file** *name*

Cause: SQL*DBA cannot open the file specified by the ORA_SQLDBAINIT variable, because either the file does not exist or SQL*DBA has insufficient privilege to open the file.

Action: Check that the specified file exists and can be opened.

DBA-00320

DBA-00320   **cannot open sqldba configuration file** *name*

Cause: File mentioned cannot be opened.

Action: Check that the specified file exists and can be opened.

DBA-00321

DBA-00321   **instance name too long**

Cause: An instance name was specified that may be valid, but is too long(exceeds 64 characters).

Action: Check the name of the desired instance and try again.

DBA-00322

DBA-00322   **total size of command line parameters exceeds buffer size**

Cause: Too many command-line arguments were specified to fit in theSQL*DBA buffer.

Action: Specify the arguments over several command lines.

DBA-00325

DBA-00325   pfile too large

Cause: The file specified using PFILE is too large (exceeds 8K).

Action: Reduce the size of the parameter file before specifying it again using PFILE.

DBA-00327

DBA-00327   **command not available in this mode**

Cause: A command was specified that is not available in this mode.

Action: Switch to a mode where the command is allowed.

DBA-00328

DBA-00328   **insufficient privilege for this display**

Cause: An attempt was made to display a MONITOR display without sufficient privileges for the underlying V$ tables.

Action: Contact the database administrator to obtain the required privileges.

DBA-00329

DBA-00329   **insufficient privilege for SHOW SGA**

Cause: An attempt was made to use SHOW SGA without sufficient privileges.

Action: Contact the database administrator to obtain the required privileges.

DBA-00330

DBA-00330   **logsource name too long**

Cause: The specified name may be valid, but is too long (exceeds 127 characters).

Action: Use a shorter name. For example, relocate the file so that its path nameis shorter.

DBA-00331

DBA-00331   **cannot allocate enough memory for SQL buffer**

Cause: There is not enough memory for the current SQL buffer.

Action: Use the SET command to reduce MAXDATA. Re-invoke MONITOR FILEIO.

DBA-00336

DBA-00336   **insufficient privilege for SHOW PARAMETERS**

Cause: An attempt was made to use SHOW PARAMETERS withoutsufficient privileges.

Action: Contact the database administrator to obtain the required privileges.

DBA-00337

DBA-00337   **missing instance name**

Cause: The instance name was not specified in the CONNECT statement.

Action: Issue the CONNECT statement again, using the proper instance name.

DBA-00341

DBA-00341   *name* **is an undefined bind variable**

Cause: The SQL statement refers to an undefined bind variable.

Action: Use the VARIABLE statement to define the bind variable and re-executethe query.

DBA-00342

DBA-00342	**unable to complete internal login**

Cause: Some database error occurred that prevented a successful CONNECT INTERNAL. For example, a SHUTDOWN ABORT command was issued. This message is usually followed by another message that indicates the problem.

Action: Ensure that the instance is started correctly before trying to CONNECT INTERNAL. You may need to use the STARTUP FORCE command.

DBA-00343

DBA-00343   no such parameter

Cause: SHOW PARAMETER was given a parameter name that does not exist.

Action: Use a different argument to SHOW PARAMETER.

DBA-00344

DBA-00344   **no items for the requested operation currently exist**

Cause: A mandatory list element in a dialog contained zero (0) elements.

Action: No action required.

DBA-00345

DBA-00345   no user sessions exist

Cause: An attempt was made to see if users were connected, but no userswere connected.

Action: No action required.

DBA-00346

DBA-00346   no unresolved in-doubt transactions exist

Cause: There were no in-doubt transactions to resolve.

Action: No action required.

DBA-00347

DBA-00347   **no offline tablespaces exist**

Cause: There were no offline tablespaces to bring online.

Action: No action required.

DBA-00348

DBA-00348   no online tablespaces exist

Cause: There were no tablespaces to take offline. You cannot take the SYSTEM tablespace offline.

Action: No action required.

DBA-00349

DBA-00349   **no available rollback segments exist**

Cause: There were no offline rollback segments to drop.

Action: No action required.

DBA-00350

DBA-00350   no offline rollback segments exist

Cause: There were no offline or partly available rollback segments to bring online.

Action: No action required.

DBA-00351

DBA-00351   **no online rollback segments exist**

Cause: There were no online rollback segments to take offline.

Action: No action required.

DBA-00352

DBA-00352   no online tablespaces exist for backup

Cause: There were no online tablespaces to backup.

Action: No action required.

DBA-00353

DBA-00353   **no online tablespaces being backed up exist**

Cause: An attempt was made to end an online tablespace backup, but no tables are being backed up.

Action: No action required.

DBA-00354

DBA-00354   **no offline tablespaces exist**

Cause: There were no offline tablespaces to recover.

Action: No action required.

DBA-00355

DBA-00355   **no offline datafiles exist**

Cause: There were no offline datafiles to recover.

Action: No action required.

DBA-00356

DBA-00356   no users exist

Cause: There were no named users to alter or drop.

Action: No action required.

DBA-00357

DBA-00357   no roles exist

Cause: There were no roles to alter or drop.

Action: No action required.

DBA-00358

DBA-00358   no users or roles exist

Cause: There were no users or roles to grant privileges and roles to or to revoke privileges and roles from.

Action: Create some users before granting roles or privileges.

DBA-00359

DBA-00359   **monitor already active**

Cause: An instantiation of this monitor was already active.

Action: Cycle through the monitor windows until the monitor becomes visible.

00360-00369: DBA Describe Execution

DBA-00360

DBA-00360   **object to be described does not exist**

Cause: The object in a DESCRIBE FUNCTION / PROCEDURE / PACKAGE statement did not exist.

Action: Check that the object name and owner are correct and that the object exists.

DBA-00361

DBA-00361   error during describe

Cause: An unexpected error occurred during describe.

Action: Check the following error and correct the problem.

DBA-00362

DBA-00362   **object *name* is a package; use DESCRIBE *name.<procedure>***

Cause: The named object was a package. DESCRIBE does not currently describe an entire package specification.

Action: Specify an entire package.

DBA-00363

DBA-00363   **procedure or function *name* not found in the package**

Cause: The named package did not contain the procedure of the function specified.

Action: Specify a procedure or function within the package.

DBA-00364

DBA-00364   **object *name* is a remote object, cannot further describe**

Cause: The specified object name contained a database link or was a synonymthat resolved to a name with a database link. Such objects cannot currentlybe described.

Action: Specify a local object.

DBA-00365

DBA-00365 **object *name* is invalid, it may not be described**

Cause: The object must have been successfully compiled.

Action: Fix any errors in the object and recompile.

DBA-00366

DBA-00366   **name *name* is malformed, it must be of form [[a.]b.]c@dblink**

Cause: The name may have at most three parts and a dblink.

Action: Use a well formed object name.

00370-00380: DBA Generic Execution

DBA-00370

DBA-00370   **mandatory field/list needs to be filled in**

Cause: An attempt was made to execute the dialog before filling in all required items.

Action: Fill in all required items and retry.

DBA-00371

DBA-00371   **cannot open/locate input help file,** *name*

Cause: The Help Key was pressed or an item was chosen from the Help Menu, and the help file was not in the expected location.

Action: Put the help file in the location specified.

DBA-00372

DBA-00372   **cannot open/locate input index file,** *name*

Cause: The Help Key was pressed or an item was chosen from the Help Menu, and the help file was not in the expected location.

Action: Put the index help file in the location specified.

DBA-00373

DBA-00373   **cannot allocate memory of size *num* from toolkit**

Cause: The Help Key was pressed or an item was chosen from the Help Menu, and SQL*DBA ran out of memory.

Action: Exit SQL*DBA, restart, and try again.

DBA-00374

DBA-00374 could not set file position in *name* directly

Cause: The Help Key was pressed or an item was chosen from the Help Menu.

Action: Alternate chosen − no action required.

DBA-00375

DBA-00375   unexpected end of file, *name*

Cause: The Help Key was pressed or an item was chosen from the Help Menu, but the help file specified was not installed.

Action: Exit SQL*DBA and install the correct help file.

DBA-00376

DBA-00376   **mandatory field has to be filled in before navigation is possible**

Cause: An attempt was made to navigate to the next/previous item without filling a required field.

Action: Fill in current field before proceeding to another field.

DBA-00377

DBA-00377   **mandatory list item has to be selected before navigation is possible**

Cause: An attempt was made to navigate to the next/previous item without filling a required field.

Action: Select a list item before proceeding to another field.

DBA-00378

DBA-00378   **an item has to be selected before help can be sought for it**

Cause: An attempt was made to obtain help before selecting an item.

Action: Select an item first and re-execute.

DBA-00379

DBA-00379   **variable(s) not defined**

Cause: The SHOW VAR[IABLES] [var-name] command could not find any variables. Either no variables are defined or you specified a non-existent variable.

Action: Check the spelling of the variables that were entered or, if no variables were entered, define the necessary variables.

DBA-00380

DBA-00380   **procedural option required for this statement**

Cause: The attempted statement requires the procedural option.

Action: Refrain from using this statement or contact an Oracle sales representative to purchase the procedural option.

DBA-00381

DBA-00381   **error in the SERVER OUTPUT option**

Cause: Most likely the package DBMS_OUTPUT was not installed.

Action: Install the package DBMS_OUTPUT and check the accompanying messages for more information.

DBA-00382

DBA-00382   value *name* is not a recognizable filename for *name*

Cause: The value for the SQL*DBA initialization variable, which points to the initialization file, is not a recognizable filename.

Action: Specify a valid filename or do not define the variable so that no initialization file is run.

DBA-00383

DBA-00383   **filename *name* pointed to by *name* could not be opened**

Cause: The filename pointed to by the SQL*DBA initialization variable could notbe opened.

Action: Specify an available file or do not define the variable so that no initialization file is run.

DBA-00384

DBA-00384 **could not open SQL*DBA resource file,** *name*

Cause: The file mentioned could not be opened.

Action: Make the file mentioned available for use by SQL*DBA.

DBA-00385

DBA-00385   **could not locate SQL*DBA resource file,** *name*

Cause: The file mentioned could not be located.

Action: Check that the file exists before rerunning SQL*DBA.

DBA-00386

DBA-00386 **could not open toolkit resource file,** *name*

Cause: The file mentioned could not be opened.

Action: Make the file mentioned available for use by SQL*DBA.

DBA-00387

DBA-00387   **could not locate toolkit resource file, *name***

Cause: The file mentioned could not be located.

Action: Check that the file exists before rerunning SQL*DBA.

DBA-00388

DBA-00388   **cannot start SQL*DBA in screen mode; check if values are legal**

Cause: A variable was not properly set.

Action: Check the values listed by SQL*DBA to see if they are correct.

DBA-00389

DBA-00389 **toolkit resource file** *name* **is defined to** *name*

Cause: File mentioned could not be located.

Action: Check that the file specified exists and can be accessed.

DBA-00390

DBA-00390   **terminal type** *name* **is defined by** *name*

Cause: Terminal type specified by may not be valid.

Action: Check that the terminal specification is correct.

DBA-00391

DBA-00391   **value *name* is defined to *name*; legal values: *name* or *name***

Cause: SQL*DBA mode has been incorrectly defined.

Action: Define the value of the variable to one of the legal values.

DBA-00392

DBA-00392   **this field requires a numeric value**

Cause: An attempt was made to navigate to the next/previous item with a non-numeric value in a field requiring a numeric value.

Action: Enter a numeric value in the field and repeat the operation.

00100-00299: LCC Messages

The following messages result from errors occurring in the parameter file. These messages occur when attempting to start an instance.

LCC-00100

LCC-00100   **internal error, argument *num***

Cause: An internal error was encountered.

Action: Call customer support with the circumstances leading to the error and the complete set of messages.

LCC-00111

LCC-00111   **value not in legal range** *range*

Cause: The value of the parameter was outside the valid range.

Action: Check the valid range and retry using a new value.

LCC-00112

LCC-00112	**illegal integer radix specification** *character*

Cause: An invalid character was specified.

Action: Valid characters are 'd', 'h', 'D', and 'H'.

LCC-00113

LCC-00113   **integer conversion error or negative integer** *num*

Cause: A non-integer or negative integer was assigned to an integer parameter.

Action: Retry using a valid integer value.

LCC-00114

LCC-00114   **illegal boolean response** *character*

Cause: A value other than TRUE or FALSE was specified.

Action: Enter a boolean value, either TRUE or FALSE.

LCC-00122

LCC-00122   **unrecognized keyword** *name*

Cause: An invalid or misspelled keyword was specified.

Action: Re-enter the line.

LCC-00128

LCC-00128   **invalid positional parameter value** *num*

Cause: An invalid positional parameter value was specified.

Action: Correct the value.

LCC-00161

LCC-00161   **Oracle error (possible syntax error) parameter** *name*

Cause: A parameter was specified incorrectly, possibly with a syntax error.

Action: Correct the error as described in the accompanying message.

LCC-00201

LCC-00201   **could not open specified filename** *name*

Cause: The specified file did not exist.

Action: Check the name of the file or create a file by that name before retrying.

LCC-00203

LCC-00203   **missing keyword** *name*

Cause: A keyword is expected but none was found.

Action: Add a parameter keyword followed by an equal sign and a parameter value.

LCC-00204

LCC-00204	**left parenthesis and no parameter specified** *name*

Cause: A parameter list was started but no parameter was specified.

Action: Add an appropriate list of values and close the parameter list.

LCC-00205

LCC-00205   **unbalanced parentheses** *character*

Cause: An odd number of parentheses was found, indicating that one is missing or one is extra.

Action: Retry the statement using the correct number of parentheses.

LCC-00206

LCC-00206   **positional parameter entered after keyword** *name*

Cause: An equal sign was missing.

Action: Retry the parameter specification using an equal sign between the parameter name and the value.

LCC-00207

LCC-00207	nested parentheses encountered *character*

Cause: More than one set of parentheses was used. Only one set of parenthesesis permitted.

Action: Remove the nested parentheses and retry.

LCC-00208

LCC-00208   **unexpected keyword in value list** *name*

Cause: A keyword was found instead of a value.

Action: Insert an appropriate value for the keyword.

LCC-00209

LCC-00209   **missing value for keyword at end of string** *str*

Cause: A keyword was specified with no value.

Action: Specify a valid value for the keyword.

LCC-00210

LCC-00210   **illegal assignment operator** *symbol*

Cause: An attempt was made to use a symbol other than an equal sign to assign a value to a parameter.

Action: Retry using an equal sign following the parameter name.

LCC-00211

LCC-00211	**unexpected delimiter** *symbol*

Cause: An invalid delimiter was found between values.

Action: A comma or a space is a valid delimiter.

LCC-00212

LCC-00212	**runaway quoted string** *str*

Cause: The parameter value was quoted on the left side but not on the right.

Action: Balance the quotation marks.

LCC-00215

LCC-00215   **parameter files nested too deep** *num*

Cause: Too many parameter files were nested. The maximum number of files that may be nested is three.

Action: Reduce the number of nested parameter files.

LCC-00217

LCC-00217   failure while processing file parameter

Cause: An error occurred while processing a file parameter.

Action: Other messages will follow with additional information.

LCC-00218

LCC-00218   **error in file** *name*

Cause: An error occurred in the particular file.

Action: Other messages will follow with additional information.

Export/Import Messages

This chapter lists messages generated by the Export/Import utility. For more information on this utility, refer to *Oracle7 Server Utilities*.

The messages listed in this chapter are displayed with the either the prefix EXP, signifying Export, or IMP, signifying Import.

00000-00100: Export Messages

EXP-00000

EXP-00000   **export terminated unsuccessfully**

Cause: Export encountered an Oracle error.

Action: Look up the accompanying Oracle message, and take appropriate action.

EXP-00001

EXP-00001   **data field truncation - column length=*num*, buffer size=*num* fetched=*num***

Cause: Export could not fit a column in the data buffer.

Action: Record the given size parameters and the accompanying messages and report this as an Export internal error to customer support. (Part of the table has been exported. Export will continue with the next table.)

EXP-00002

EXP-00002	error in writing to export file

Cause: Export could not write to the export file, probably because of a device error. This message is usually followed by a device message from theoperating system.

Action: Take appropriate action to restore the device.

EXP-00003

EXP-00003   **no storage definition found for segment (*name, num*)**

Cause: Export could not find the storage definitions for a cluster, index, or table.

Action: Record the accompanying messages and report this as an Export internal error to customer support.

EXP-00004

EXP-00004   **invalid username or password**

Cause: An invalid username or password was specified.

Action: Retry with a valid username and password.

EXP-00005

EXP-00005   **all allowable logon attempts failed**

Cause: Attempts were repeatedly made to log on with an invalid usernameor password.

Action: Shut down the utility, then restart and retry with a valid usernameand password.

EXP-00006

EXP-00006	**internal inconsistency error**

Cause: Export's data structure was corrupted.

Action: Record the accompanying messages and report this as an Export internal error to customer support.

EXP-00007

EXP-00007   **dictionary shows no columns for *name*.*name***

Cause: Export failed to gather column information from the data dictionary. The table may have been dropped.

Action: Retry the export and, if this error recurs, report it as an Export internal error to customer support.

EXP-00008

EXP-00008   **Oracle error *num* encountered**

Cause: Export encountered the referenced Oracle error.

Action: Look up the Oracle message and take appropriate action. (To find information on the error number, click the Help Topics button in this window, click the Index tab, then type the error number, including its three-letter prefix with the hyphen.)

EXP-00009

EXP-00009   **no privilege to export *name*'s table *name***

Cause: An attempt was made to export another user's table. Only a database administrator can export another user's tables.

Action: Ask your database administrator to do the export.

EXP-00010

EXP-00010   *name* **is not a valid username**

Cause: An invalid username was specified.

Action: Shut down the utility, then restart and retry with a valid username.

EXP-00011

EXP-00011   *name.name* **does not exist**

Cause: Export could not find the specified table.

Action: Retry with the correct table name.

EXP-00012

EXP-00012 *num* **is not a valid export mode**

Cause: An invalid export mode was specified.

Action: Retry with a valid export mode.

EXP-00013

EXP-00013   respond with either "Y", "N", RETURN or "." to quit

Cause: An invalid response was entered.

Action: Enter any of the responses shown in the message.

EXP-00014

EXP-00014   **error on row *num* of table *name***

Cause: Export encountered an Oracle error while fetching rows.

Action: Look up the accompanying Oracle message and take appropriate action. (To find information on the error number, click the Help Topics button in this window, click the Index tab, then type the error number, including its three-letter prefix with the hyphen.)

EXP-00015

EXP-00015   **error on row *num* of table *name*, column *name*, datatype *num***

Cause: Export encountered an error while fetching or writing the column. An accompanying message gives more information.

Action: Correct the error and try again.

EXP-00016

EXP-00016   **Oracle error encountered while reading default auditing options**

Cause: Export encountered an Oracle error while reading the default auditing options (those for updates, deletes, and so on).

Action: Look up the accompanying Oracle message and take appropriate action. (To find information on the error number, click the Help Topics button in this window, click the Index tab, then type the error number, including its three-letter prefix with the hyphen.)

EXP-00018

EXP-00018 **datatype (*num*) for column *name*, table *name.name* is not supported**

Cause: Export does not support the referenced datatype.

Action: Retry with an acceptable datatype (CHAR, NUMBER, DATE, LONG, or RAW).

EXP-00019

EXP-00019   **failed to process parameters, type "EXP HELP=Y" for help**

Cause: Invalid command-line parameters were specified.

Action: Check the online help screen for the set of valid parameters, then retry.

EXP-00020

EXP-00020   **failed to allocate memory of size** *num*

Cause: Export failed to allocate the necessary memory.

Action: Decrease the export buffer size so that less memory is required, or increase the runtime memory size for Export.

EXP-00021

EXP-00021   **can only perform incremental export in Full Database mode**

Cause: USER or TABLE mode was specified when doing an incremental export.

Action: Specify FULL database mode (FULL=Y) and retry.

EXP-00022

EXP-00022   **must be SYS or SYSTEM to do incremental export**

Cause: The privileges needed to do an incremental export do not exist. Only users SYS and SYSTEM can do incremental exports.

Action: Ask the database administrator to do the incremental export.

EXP-00023

EXP-00023   **must be a DBA to do Full Database export**

Cause: The privileges needed to do a FULL database export do not exist. Only a database administrator can do a FULL database export.

Action: Ask the database administrator to do the FULL database export.

EXP-00024

EXP-00024   **export views not installed, please notify your DBA**

Cause: The necessary Export views were not installed.

Action: Ask the database administrator to install the required Export views.

EXP-00025

EXP-00025   **dictionary shows no column for constraint *name.num***

Cause: Export failed to gather column information about the referenced constraint from the data dictionary. The constraint may have been altered.

Action: Retry the export and, if this error recurs, report it as an Export internal error to customer support.

EXP-00026

EXP-00026   **only one parameter (TABLES, OWNER, or FULL) can be specified**

Cause: Conflicting export modes were specified.

Action: Specify only one parameter and retry.

EXP-00027

EXP-00027   **failed to calculate Oracle block size**

Cause: Export failed to calculate the Oracle block size.

Action: Report this as an Export internal error to customer support.

EXP-00028

EXP-00028	**failed to open** *name* **for write**

Cause: Export failed to open the export file for writing. This message is usually followed by device messages from the operating system.

Action: Take appropriate action to restore the device.

EXP-00029

EXP-00029   **incremental export mode and consistent mode are not compatible**

Cause: Both consistent and incremental exports were specified.

Action: None. Consistent mode is turned off.

EXP-00030

EXP-00030   **unexpected End-Of-File encountered while reading input**

Cause: Encountered an End-Of-File while reading the user input.

Action: If input to export is being redirected, check the file for errors.

EXP-00031

EXP-00031   **arraysize not in valid range. Using arraysize=***num*

Cause: The arraysize value specified is not in the valid range.

Action: None.

EXP-00032

EXP-00032   **non-DBAs may not export other users**

Cause: Only database administrators can export to other users. A non-database administrator attempted to specify owner=user where exporter is not the user.

Action: Request that this operation be performed by the database administrator.

EXP-00033

EXP-00033   **could not find environment character set**

Cause: The environment character set is missing or incorrectly specified.

Action: Ensure that the environment character set is correctly specified and is present.

EXP-00034

EXP-00034   **error on rowid: file#** *num* **block#** *num* **slot#** *num*

Cause: Identifies the rowid on which an error occurred.

Action: This is an information message. No action is required.

EXP-00035

EXP-00035   **cannot specify Rows=N and Compress=Y**

Cause: Compress applies only to rows of data, so to request compress you must also request rows.

Action: Remove the compress parameter or specify Rows=Y.

EXP-00037

EXP-00037 **Export views not compatible with database version**

Cause: The Export utility is at a higher version than the database version and is thereby incompatible.

Action: Use the same version of Export utility as the database.

00000-00100: Import Messages

IMP-00000

IMP-00000   **import terminated unsuccessfully**

Cause: Import encountered an Oracle error.

Action: Look up the accompanying Oracle message and take appropriate action. (To find information on the error number, click the Help Topics button in this window, click the Index tab, then type the error number, including its three-letter prefix with the hyphen.)

IMP-00001

IMP-00001   **respond with either "Y", "N", "RETURN" or "." to quit**

Cause: An invalid response was entered.

Action: Enter any of the responses shown in the message.

IMP-00002

IMP-00002   **failed to open *name* for read**

Cause: Import failed to open the export file for reading. This message is usually followed by a device message from the operating system.

Action: Take appropriate action to restore the device.

IMP-00003

IMP-00003   **Oracle message code *num* encountered**

Cause: Import encountered the referenced Oracle error.

Action: Look up the Oracle message and take appropriate action. (To find information on the error number, click the Help Topics button in this window, click the Index tab, then type the error number, including its three-letter prefix with the hyphen.)

IMP-00004

IMP-00004   **invalid username or password**

Cause: An invalid username or password was specified.

Action: Retry with a valid username and password.

IMP-00005

IMP-00005   **all allowable logon attempts failed**

Cause: An attempt was repeatedly made to log on with an invalid usernameor password.

Action: Retry with valid username and password.

IMP-00006

IMP-00006   **failed to allocate memory of size** *num*

Cause: Import failed to allocate the necessary memory.

Action: Decrease the import buffer size so that less memory is required, or increase the runtime memory size for Import.

IMP-00007

IMP-00007 **must be a DBA to import objects to another user's account**

Cause: The privileges needed to import into another user's account do not exist. Only a database administrator can import into another user's account.

Action: Ask the database administrator to do the import.

IMP-00008

IMP-00008   **unrecognized statement in the export file:** *str*

Cause: Import did not recognize a statement in the export file. Either the export file was corrupted, or an Import internal error has occurred.

Action: If the export file was corrupted, retry with a new export file. Otherwise, report this as an Import internal error to customer support.

IMP-00009

IMP-00009 **abnormal end of export file**

Cause: The export file is probably from an aborted Export session.

Action: If so, retry the export and import. Otherwise, report this as an Import bug and submit the export file that caused this error to customer support.

IMP-00010

IMP-00010   **not a valid export file, header failed verification**

Cause: Either the file was not generated by Export or it was corrupted.

Action: If the file was indeed generated by Export, report this an Import bug and submit the export file to customer support.

IMP-00011

IMP-00011   **formatted table data not currently supported**

Cause: You tried to import an export file that contains formatted table data. Import only supports export files containing binary table data.

Action: Retry using an export file that contains only binary table data.

IMP-00012

IMP-00012   **invalid export mode *num* in header**

Cause: The Export mode specified in the export file header is invalid.

Action: Check the export file to see if it was corrupted. If it was not, report this as an Import bug and submit the export file to customer support.

IMP-00013

IMP-00013   **only a DBA can import a file exported by another DBA**

Cause: The privileges needed to Import an export file generated by a database administrator do not exist. Only a database administrator can import such files.

Action: Ask the database administrator to do the import.

IMP-00014

IMP-00014   **insufficient Oracle privileges to do import**

Cause: The privileges needed to do an import do not exist. A minimum of CONNECT privilege is required to do an import.

Action: Ask the database administrator to grant the necessary privileges.

IMP-00015

IMP-00015   **following statement failed because the object already exists:** *str*

Cause: Import failed to create an object because it already exists.

Action: Specify IGNORE=Y to ignore such errors and import rows even when tables are not created because they already exist.

IMP-00016

IMP-00016   **required character set conversion (type *num* to *num*) not supported**

Cause: Import could not convert the character format of the export file into the native character format.

Action: Change the user character set by setting the NLS_LANG environment variable to match the character set of the export file.

IMP-00017

IMP-00017   **following statement failed with Oracle message *num*: *str***

Cause: Import failed to execute the statement from the export file because of anOracle error.

Action: Look up the accompanying Oracle message and take appropriate action. (To find information on the error number, click the Help Topics button in this window, click the Index tab, then type the error number, including its three-letter prefix with the hyphen.)

IMP-00018

IMP-00018 **partial import of previous table complete:** *num* **rows imported**

Cause: A table was only partially imported because of an Oracle error.

Action: Look up the accompanying Oracle message and take appropriate action. (To find information on the error number, click the Help Topics button in this window, click the Index tab, then type the error number, including its three-letter prefix with the hyphen.)

IMP-00019

IMP-00019   **row rejected due to Oracle message code** *num*

Cause: Import encountered the referenced Oracle error while trying to import a row.

Action: Look up the accompanying Oracle message and take appropriate action. (To find information on the error number, click the Help Topics button in this window, click the Index tab, then type the error number, including its three-letter prefix with the hyphen.)

IMP-00020

IMP-00020   **long column too large for column buffer size** *num*

Cause: The column buffer is too small. This usually occurs when importingLONG data.

Action: Increase the insert buffer size 10,000 bytes at a time (for example) up to 66,000 or greater. Use this step-by-step approach because a buffer size that is too large may cause a similar problem.

IMP-00021

IMP-00021   **operating system error - error code (dec *num*, hex 0x*num*)**

Cause: An operating system error has occurred.

Action: Look up the referenced message in your operating system-specific Oracle documentation and take appropriate action.

IMP-00022

IMP-00022   **failed to process parameters, type "IMP HELP=Y" for help**

Cause: Invalid command-line parameters were specified.

Action: Check the online help screen for the set of valid parameters, then retry.

IMP-00023

IMP-00023　　**import views not installed, please notify your DBA**

Cause:　　The necessary Import views were not installed.

Action:　　Ask your database administrator to install the required Import views.

IMP-00024

IMP-00024 **full import mode, cannot specify FROMUSER or TABLES parameter**

Cause: Parameters were specified that conflict with the import specification FULL=Y.

Action: Retry, making sure not to specify FULL=Y.

IMP-00025

IMP-00025   **for DBA export files, FROMUSER must be specified with TOUSER**

Cause: When importing a database administrator export file, you specified the TOUSER parameter but not the FROMUSER parameter.

Action: Specify the FROMUSER parameter so that Import knows which user's objects to import.

IMP-00027

IMP-00027	**failed to roll back partial import of previous table**

Cause: Import encountered an Oracle error while trying to roll back a partial import.

Action: Look up the accompanying Oracle message and take appropriate action. Then, log on to Oracle and check that the partial import was not committed.

(To find information on the error number, click the Help Topics button in this window, click the Index tab, then type the error number, including its three-letter prefix with the hyphen.)

IMP-00028

IMP-00028   **partial import of previous table rolled back: *num* rows rolled back**

Cause: Import encountered an Oracle error while trying to import a table.

Action: Look up the accompanying Oracle message and take appropriate action. Then, if necessary, re-import the table.

(To find information on the error number, click the Help Topics button in this window, click the Index tab, then type the error number, including its three-letter prefix with the hyphen.)

IMP-00029

IMP-00029   **cannot qualify table name by owner *name.name*, use FROMUSER parameter**

Cause: A table name was qualified with the name of its owner, as shown in the following example. This is not allowed.

```
IMP SYSTEM/MANAGER TABLES=(SCOTT.EMP,SCOTT.DEPT)
```

Action: Use the FROMUSER parameter to specify the table's owner, as shown in the following example:

```
IMP SYSTEM/MANAGER FROM USER=SCOTT TABLES=(EMP,DEPT)
```

IMP-00030

IMP-00030   **failed to create file *name* for write**

Cause: Import was unable to create the specified file with write enabled.

Action: Check the file name and file system for the source of the error.

IMP-00031

IMP-00031   **must specify FULL=Y or provide FROMUSER/TOUSER orTABLE arguments**

Cause: The database administrator did not specify full or partial import.

Action: The database administrator must specify FROMUSER/TOUSER or table arguments if not a full import.

IMP-00032

IMP-00032	SQL statement exceeded buffer length

Cause: The buffer was too small for the SQL statement being read.

Action: Rerun with a larger buffer. This can also be an indication of a corruption in the import datafile

IMP-00033

IMP-00033   **warning: Table *name* not found in export file**

Cause: A table name was specified that does not exist in export file.

Action: Correct the table specification.

IMP-00034

IMP-00034   **warning: FROMUSER *name* not found in export file**

Cause: The user specified a FROMUSER name that does not exist in export file.

Action: Correct the FROMUSER specification.

IMP-00035

IMP-00035   **warning: TOUSER *name* not found in export file**

Cause: The user specified a TOUSER name that does not exist in export file.

Action: Correct the TOUSER specification.

IMP-00036

IMP-00036   **could not find environment character set**

Cause: An environment character set was specified that was not recognized by the Import utility.

Action: Ensure that the spelling of the character set specified in the command line parameter CHARSET or the environment variable NLS_LANG is correct.

IMP-00037

IMP-00037   character set marker unknown

Cause: The export file is corrupted.

Action: Try to obtain an uncorrupted version of the export file.

IMP-00038

IMP-00038   **could not find environment character set handle**

Cause: An environment character set handle was specified that was not recognized by the Import utility.

Action: Ensure that the spelling of the environment character set handle is correct.

IMP-00039

IMP-00039	no space for owner name in create string

Cause: The string is corrupted.

Action: Try to obtain an uncorrupted version of the export file.

IMP-00040

IMP-00040 warning: **Statement not executed due to invalid session label**

Cause: An ALTER SESSION SET LABEL statement in the export file failed because it specified an invalid label. This caused all subsequent CREATE statements until the next valid ALTER SESSION SET LABEL statement to fail. These failed CREATE statements also generate this warning.

Action: If the invalid label on the ALTER SESSION SET LABEL statement should be valid, define the label within the operating system and re-execute the import. It would be desirable to have an invalid label be valid if an export file is being imported from a Trusted Oracle7 database on a different operating system (one that supports different labels).

If the invalid label must remain invalid, replace it with a valid label in the export file and re-execute the import.

IMP-00041

IMP-00041   **warning: object created with compilation warnings**

Cause: The object in the SQL statement following this error was created with compilation errors. If this error occurred for a view, it is possible that the base table of the view was missing.

Action: This is a warning. The object may have to be recompiled before being used.

IMP-00042

IMP-00042   **CHARSET used, but the export file has specific character set**

Cause: The user requested that a specific character set be used to process an export file that has a specific embedded character set ID. The ID in the export file is accurate and should be used.

Action: Remove the CHARSET keyword from the command line.

IMP-00043

IMP-00043   **export file character set inconsistent**

Cause: The character set of the export file and the environment character set (or the character set specified with the CHARSET keyword) are inconsistent. One character set is ASCII-based while the other is EBCDIC based.

Action: Specify the correct character set with the CHARSET keyword.

IMP-00044

IMP-00044   **unable to allocate enough memory for statement**

Cause: Import was unable to allocate sufficient memory to read in the specified statement and execute it.

Action: Increase the allocation for database buffers, restart the instance, and reexecute the Import command.

IMP-00045

IMP-00045   **unexpected character found on line *num* in mapfile**

Cause: There was a syntax error while processing the mapfile. This message is generated only for Trusted Oracle7.

Action: Examine the specified line and correct the error.

IMP-00046

IMP-00046   **must specify MLS_MAPFILE with MLS_LISTLABELS**

Cause: A map file for output of the list labels option was not specified. This message is generated only for Trusted Oracle7.

Action: Specify a map file using the MLS_MAPFILE option.

IMP-00047

IMP-00047 **label string larger than 255 characters found at line *num* in mapfile**

Cause: A label string was specified in your mapfile that was too large. This message is generated only for Trusted Oracle7.

Action: Shorten label string to less than 255 characters.

IMP-00048

IMP-00048   **the MLS_LISTLABELS option cannot be used with this export file**

Cause: An attempt was made to list the labels on an export file earlier than Release 7.1. This message is generated only for Trusted Oracle7.

Action: Only use the MLS_LISTLABELS option for a Release 7.1 or later export file.

IMP-00049

IMP-00049   **unknown language specified in CHARSET**

Cause: An unknown language was listed in the CHARSET option. This message is generated only for Trusted Oracle7.

Action: Use a known character set.

SQL*Loader Messages

This chapter lists messages you might see when using SQL*Loader. SQL*Loader messages take the form:

`SQL*Loader-`*code number*`:` *message text*

Along with its own messages, SQL*Loader sometimes displays related messages issued by the Oracle7 Server. These messages can be found in Chapter 2. For more information on SQL*Loader, see *Oracle7 Server Utilities*.

00100-00199: Command Line

SQL*Loader-100

SQL*Loader-100   syntax error on command line

Cause: Possible causes for this error include: placing a positional argument after keyword arguments, misspelling a keyword, not balancing parentheses or quotes, or leaving space between the operator, '=', and an argument.

Action: Check the command syntax and the spelling, then retry.

SQL*Loader-101

SQL*Loader-101   invalid argument for *username/password*

Cause: The username/password argument specified on the command line was not recognized. Another argument (not identified by a keyword) could be in the position where username/password is expected.

This error could also result from a spelling mistake. The password, ifpresent, must be separated by a slash (/). No spaces can appear between the slash and username or password. Otherwise, SQL*Loader sees them as multiple arguments.

Action: Check the format of the username/password argument, which is the first argument on the command line, if arguments are given without keywords.

SQL*Loader-102

SQL*Loader-102   invalid control file name on command line

Cause: The control filename specified on the command line was not recognized. It could be misspelled, or another argument (not identified by a keyword) could be in its place. One possible cause is the existence of a space between username and password. See error 101, above.

Action: Check the spelling and position of the arguments on the command line.

SQL*Loader-103

SQL*Loader-103   invalid log file name on command line

Cause: The log file name specified on the command line was not recognized. It couldbe misspelled, or another argument (not identified by a keyword) could be in its place.

Action: Check the spelling and position of the arguments on the command line.

SQL*Loader-104

SQL*Loader-104   invalid bad file name on command line

Cause: The bad file name specified on the command line was not recognized. It could be misspelled, or another argument (not identified by a keyword) could be in its place.

Action: Check the spelling and position of the arguments on the command line.

SQL*Loader-105

SQL*Loader-105   invalid datafile name on command line

Cause: The datafile name specified on the command line was not recognized. It could be misspelled, or another argument (not identified by a keyword) could be in its place.

Action: Check the spelling and position of the arguments on the command line.

SQL*Loader-106

SQL*Loader-106   invalid discard file name on command line

Cause: The discard file name specified on the command line was not recognized. It could be misspelled, or another argument (not identified by a keyword) could be in its place.

Action: Check the spelling and position of the arguments on the command line.

SQL*Loader-107

SQL*Loader-107   invalid maximum number of discards

Cause: The argument's value is inappropriate, or another argument (not identified by a keyword) is in its place.

Action: Check the command line and retry.

SQL*Loader-108

SQL*Loader-108   invalid number of logical records to load

Cause: The argument's value is inappropriate, or another argument (not identified by a keyword) is in its place.

Action: Check the command line and retry.

SQL*Loader-109

SQL*Loader-109   invalid number of logical records to skip

Cause: The argument's value is inappropriate, or another argument (not identified by a keyword) is in its place.

Action: Check the command line and retry.

SQL*Loader-110

SQL*Loader-110   invalid maximum number of errors

Cause: The argument's value is inappropriate, or another argument (not identified by a keyword) is in its place.

Action: Check the command line and retry.

SQL*Loader-111

SQL*Loader-111   invalid number of rows for bind array or data saves

Cause: The argument's value is inappropriate, or another argument (not identified by a keyword) is in its place.

Action: Check the command line and retry.

SQL*Loader-112

SQL*Loader-112   invalid maximum bind array size

Cause: The argument's value is inappropriate, or another argument (not identified by a keyword) is in its place.

Action: Check the command line and retry.

SQL*Loader-113

SQL*Loader-113 invalid silent mode option

Cause: The only valid options for the SILENT command-line argument are ALL, ERROR, FEEDBACK, or HEADER. The argument could be misspelled, or another argument (not identified by a keyword) could be in its place.

Action: Check the command line and retry.

SQL*Loader-114

SQL*Loader-114   error in OPTIONS statement

Cause: Command line options specified in the control file with the OPTIONS clause were found to be incorrect.

Action: Check the format of the OPTIONS clause in the control file.

SQL*Loader-115

SQL*Loader-115   invalid direct path option

Cause: The only valid options for the DIRECT command-line argument are TRUE or FALSE. The argument could be misspelled, or another argument (not identified by a keyword) could be in its place.

Action: Check the command line and retry.

SQL*Loader-116

SQL*Loader-116   error prompting for password

Cause: An internal error has occurred.

Action: Contact customer support.

00200-00299: DDL Syntax

SQL*Loader-200

SQL*Loader-200   FORMAT clause should not be present - flat data files only

Cause: SQL/DS FORMAT clause is not supported.

Action: Remove the FORMAT command from the control file or comment it out.

SQL*Loader-250

SQL*Loader-250   work data sets are not used by SQL*Loader

Cause: The control file contains a WRKDDN statement. SQL*Loader ignoresthis clause.

Action: No action required. This is an informational message.

SQL*Loader-251

SQL*Loader-251   sort devices are not used by SQL*Loader

Cause: The control file contains a SORTDEVT statement. SQL*Loader ignoresthis clause.

Action: No action required. This is an informational message.

SQL*Loader-252

SQL*Loader-252   sort data sets are not used by SQL*Loader

Cause: The control file contains a SORTNUM statement. SQL*Loader ignoresthis clause.

Action: No action required. This is an informational message.

SQL*Loader-253

SQL*Loader-253   DB2 partition number has no significance -- ignored

Cause: The control file contains a PART statement. SQL*Loader ignores this clause.

Action: No action required. This is an informational message.

SQL*Loader-254

SQL*Loader-254   cannot have DISCARDFILE specs here when multiple datafiles

Cause: The control file contained multiple INFILE statements and a DISCARDFILE statement was found below the RESUME clause.

Action: Move the DISCARDFILE statement above the RESUME clause, so it is adjacent to one of the INFILE statements.

SQL*Loader-255

SQL*Loader-255   log file for error recovery not used by SQL*Loader

Cause: The control file contains a LOG statement. SQL*Loader ignores this clause.

Action: No action required. This is an informational message.

SQL*Loader-256

SQL*Loader-256   SORTED INDEXES option allowed only for direct path

Cause: The control file contains a SORTED INDEXES statement, but it was not used in a direct path load.

Action: Specify a direct path load with DIRECT=TRUE on the command line, remove the statement from the control file, or comment it out.

SQL*Loader-257

SQL*Loader-257   index *name* specified in SORTED INDEXES does not exist on table *name*

Cause: A non-existent index was specified in the SORTED INDEXES clause. Either the index does not exist or its name was misspelled.

Action: Create the index, change the spelling, remove the specification, or commentit out.

SQL*Loader-258

SQL*Loader-258   maximum number of sorted indexes *num* exceeded on table *name*.

Cause: There are too many indexes in the SORTED INDEX clause. The message displays the maximum number that are permitted.

Action: Reduce the number of indexes specified in the SORTED INDEX clause or use the conventional path load instead of the direct path load.

SQL*Loader-259

SQL*Loader-259   could not escalate DDL share lock to exclusive on table *name*

Cause: This error occurs when another user has a parse lock on the table, for example, when another user is doing a select on the table. The parse lock should clear momentarily.

Action: Give the parse lock a chance to clear and then retry or else use theconventional path load.

SQL*Loader-260

SQL*Loader-260   index *num* is in an invalid state

Cause: The specified index is in an invalid state.

Action: Drop and re-create the index.

SQL*Loader-262

SQL*Loader-262   PIECED keyword (on column *num*) allowed only when path is direct

Cause: The PIECED keyword cannot be used in a conventional path load.

Action: Remove the PIECED keyword or use the direct path load.

SQL*Loader-263

SQL*Loader-263   PIECED column *num* must be last specified column in table *name*

Cause: A column that is not the last column was specified as PIECED.

Action: Remove the PIECED keyword or place the column last.

SQL*Loader-264

SQL*Loader-264   file mode token *name* parsed but ignored

Cause: An obsolete file mode token was used in the control file. As of Release 1.1 of SQL*Loader, the file-processing options string is used to control file processing, rather than keywords like STREAM, RECORD, FIXED, and VARIABLE.

Action: No action required. This message is informational. Removing the obsolete keywords will eliminate the message without changing the way in which the datafile is processed.

SQL*Loader-265

SQL*Loader-265   unable to get default character set name

Cause: SQL*Loader was unable to locate the default character set name for the environment.

Action: Supply a character set name with the CHARACTERSET keyword.

SQL*Loader-266

SQL*Loader-266   unable to locate character set handle for _name_

Cause: SQL*Loader could not find the character set handle for the namedcharacter set.

Action: Correct the character set name.

SQL*Loader-267

SQL*Loader-267   control file must be first datafile

Cause: The control file is specified as containing data using the INFILE "*" clause, but other datafiles were named first.

Action: Move the INFILE "*" clause so that it is the first datafile declared in thecontrol file.

SQL*Loader-268

SQL*Loader-268   UNRECOVERABLE keyword may be used only in direct path

Cause: The UNRECOVERABLE keyword can only be specified in the direct path load.

Action: Use the direct path load or remove the keyword. (Conventional path loads are always recoverable).

SQL*Loader-269

SQL*Loader-269   Null string not allowed as clause comparison text

Cause: A clause is being compared to a null string.

Action: Modify the clause to compare to at least one character.

SQL*Loader-270

SQL*Loader-270   table *name* has index defined upon it

Cause: Parallel load was specified into a table that has an index defined for it.

Action: Drop the index or indexes defined for the table or do not use parallel load.

SQL*Loader-271

SQL*Loader-271   not a parallel load. Table level OPTIONS statement ignored

Cause: A table-level OPTIONS statement was specified for a non-parallel load.

Action: Remove the OPTIONS statement from the control file.

SQL*Loader-272

SQL*Loader-272   table level OPTIONS statement ignored

Cause: In the parallel load option, the file specified on the command line overrides the file specified in the control file.

Action: Remove the OPTIONS statement from the control file.

00300-00399: DDL Parsing

SQL*Loader-303

SQL*Loader-303   non-hex character encountered where hex char expected

Cause: A non-hexadecimal character was found in a hexadecimal string.

Action: Change it to the intended hexadecimal character.

SQL*Loader-304

SQL*Loader-304   illegal combination of non-alphanumeric characters

Cause: The control file contains a combination of non-alphanumeric characters that SQL*Loader does not recognize. For example, the combination != is recognized as "not equal", but the combination =! is not valid.

Action: Remove the unrecognized characters from the control file.

SQL*Loader-305

SQL*Loader-305   more than one end of file character encountered

Cause: The file contains multiple end-of-file marks.

Action: Remove the excess end-of-file characters.

SQL*Loader-306

SQL*Loader-306   token longer than max allowable length of *num* characters

Cause: The control file contains a single word or combination of characters (a token) that is longer than the maximum permissible value. The maximum possible value is shown. This error could result from missing spaces, so that multiple tokens are joined.

Action: Check that the proper control file is being executed. Separate tokens, if joined, or shorten the token.

SQL*Loader-307

SQL*Loader-307   Warning: conflicting lengths *num1* and *num2* specified for column *name*

Cause: The control file specifies two different lengths for the named column. There are three ways to specify the length of a field:

- • with the POSITION keyword: POSITION(1:3)

- • with the length specifier: CHAR(6)

- • with the implied length of a datatype; for example, INTEGER

A conflict could occur between any two of these specifications (or among all three, but only two are shown).

A common cause of this error is the specification of some numeric datatype, such as INTEGER, when the numeric external form is intended (INTEGER EXTERNAL). The external form consists of character data, so it is considerably longer than the numeric form, which consists of binary data.

Action: No action is necessarily required, because SQL*Loader uses only one of the lengths. Check the log file under the heading "Len" in the table-description section to see which length was used. Adjusting the control file to produce uniform length specifications will remove the warning.

SQL*Loader-308

SQL*Loader-308   optional SQL string of column name must be in double quotes

Cause: A SQL string was found that was not quoted or in single quotes.

Action: Use double quotes for the SQL string, rather than single quotes.

SQL*Loader-350

SQL*Loader-350   syntax error at line *num*

Cause: *num* identifies the line in the control file at which the error occurred. This message is followed by another of the form

Expecting *str1*, found *str2*

where *str1* is a description of what SQL*Loader expected in the control file, and *str2* is what was found. Then SQL*Loader displays the offending line from the control file, indicating the location of the error in the line by a carat (^) or an asterisk (*). An example follows:

SQL*Loader-350: Syntax error at line 28
 Expecting column name, found keyword CHAR
 col3 ENCLOSED BY '"', CHAR ENCLOSED "'",

Action: Compare the DDL syntax against the syntax diagrams in *Oracle7 Server Utilities* and make any necessary corrections.

00400-00499: Columns

SQL*Loader-401

SQL*Loader-401   end field position *num* must be greater than or equal to start *num*

Cause: The named field contains a (*start*:*end*) clause in which *end* is less than *start*.

Action: Modify the clause so that *end* is greater than or equal to *start*.

SQL*Loader-402

SQL*Loader-402   unable to determine length of column *name.name* from specification

Cause: The specified datatype does not have an implied length (for example, a numeric external or RAW datatype), it is not specified with delimiters, no length was specified, and a length cannot be determined from aPOSITION clause.

Action: If a POSITION clause is specified, adding an end location to it produces a length specification. This clause then has the form

POSITION(*start:end*)

A length can also specified after the datatype, as in

INTEGER EXTERNAL (6)

Finally, the field could be specified with delimiters, or the datatype changed to one that has an implied length.

SQL*Loader-403

SQL*Loader-403   referenced column *name* **not present in table** *name*

Cause: The named column is not present in the given table.

Action: Correct the column name or table name.

SQL*Loader-404

SQL*Loader-404   column *name* present more than once in table *name*

Cause: The named column is specified more than once in a singleINTO TABLE statement.

Action: Remove the extraneous column specification.

SQL*Loader-405

SQL*Loader-405   need termination delim with optional enclosure delim: column *name.name*

Cause: The named column was specified with an optional enclosure delimiter, but no termination delimiter. Enclosure delimiters can only be optional when termination delimiters are present.

Action: Specify a termination delimiter or make the enclosure delimiters non-optional.

SQL*Loader-406

SQL*Loader-406   if data is all generated, number to load cannot be ALL

Cause: When only generated data is loaded, a number to load must be given so SQL*Loader knows when to stop.

Action: Specify a number to load.

SQL*Loader-407

SQL*Loader-407   if data is all generated, number to skip is meaningless

Cause: When all data is generated, no file is read, and there are no records to skip.

Action: Remove the number to skip.

SQL*Loader-408

SQL*Loader-408   physical record stack overflow

Cause: An internal error has occurred.

Action: Contact customer support.

SQL*Loader-409

SQL*Loader-409   number to skip must be table-level, not load-level on continued loads

Cause: The SKIP parameter was specified on the command line or in the OPTIONS statement, but the load was specified as continued with CONTINUE_LOAD.

When a multiple-table direct load is interrupted, it is possible that a different number of records were loaded into each table. As a result, the number of records to skip must be specified for each table when continuing the load. In this case, the load is specified as continued with the CONTINUE_LOAD statement, and the number of records to skip is given in eachINTO TABLE statement.

See also messages 410 and 411 for more information.

Action: Check the log file to determine the appropriate number of records to skipfor each table and specify this number in the INTO TABLE statement for each table in a continued load or with the command-line SKIP parameter in a standard load.

SQL*Loader-410

SQL*Loader-410   number to skip must be load-level, not table-level

Cause: A SKIP clause was found in the INTO TABLE statement of a standard (non-continued) load.

In a standard load, specified with LOAD DATA, it is not possible to skip a different number of records for each table. The number of records to skip must be specified for the entire load by using the SKIP parameter on the command line or in the OPTIONS clause.

If table-level skip is required, because a different number of records were loaded into each table (only possible for a multiple-table direct load), then specify a continued load with the CONTINUE_LOAD statement.

See also messages 409 and 411 for more information.

Action: If the load was not a multiple-table, direct path load, then move the SKIP clause from the INTO TABLE statements to the command line or to the OPTIONS clause. Otherwise, specify the load as continued with CONTINUE_LOAD.

SQL*Loader-411

SQL*Loader-411   only a direct path load may be continued

Cause: The load is specified with CONTINUE_LOAD, but DIRECT=FALSE. CONTINUE_LOAD is only possible for a direct path load and is only necessary for a multiple-table, direct path load when a different number of records have been loaded into each table.

See also messages 409 and 410 for more information.

Action: If CONTINUE_LOAD is necessary, specify a direct load and put the number of records to skip in each INTO TABLE statement. Otherwise, use the command line or OPTIONS clause to specify the number of records to skip and use LOAD DATA instead of CONTINUE_LOAD.

SQL*Loader-412

SQL*Loader-412   more columns specified for table *name* than the maximum *num*

Cause:　　　More columns were specified for the table than the maximum number allowed by the database.

Action:　　　Remove the extraneous columns.

SQL*Loader-413

SQL*Loader-413   maximum number of constraints *num* exceeded on table *name*

Cause: An internal error has occurred.

Action: Contact customer support.

SQL*Loader-414

SQL*Loader-414   maximum number of triggers *num* exceeded on table *name*

Cause: An internal error has occurred.

Action: Contact customer support.

SQL*Loader-415

SQL*Loader-415   **unimplemented database column type *num* on column *name***

Cause: An internal error has occurred.

Action: Contact customer support.

SQL*Loader-417

SQL*Loader-417   SQL string (on column *name*) not allowed in direct path

Cause: Because the direct path bypasses SQL processing, the SQL string cannotbe used.

Action: Remove the SQL string or use the conventional path.

SQL*Loader-457

SQL*Loader-457   comparison text *str* of CONTINUEIF LAST must have length 1 not *num*

Cause: The comparison text is too long. It can only be one character.

Action: Reduce the comparison text to one character.

SQL*Loader-458

SQL*Loader-458   comparison text of CONTINUEIF LAST must be non-whitespace

Cause: The comparison text is a whitespace character (blank or tab).

Action: Change the comparison text to a non-whitespace character.

00500-00599: File I/O and Operating System

SQL*Loader-500

SQL*Loader-500   unable to open file *name*

Cause: SQL*Loader could not open the named file.

Action: Check the operating system messages below this one in the log file.

SQL*Loader-501

SQL*Loader-501   unable to read file *name*

Cause: SQL*Loader could not read the named file.

Action: Check the operating system messages below this one in the log file.

SQL*Loader-502

SQL*Loader-502   unable to open log file for write *name*

Cause: SQL*Loader could not open the named file.

Action: Check that the file location is specified correctly and that write privileges for the file have been granted.

SQL*Loader-503

SQL*Loader-503   error appending extension to file *name*

Cause: SQL*Loader could not append the default extension to create the filename. The given name could be too long or contain illegal characters.

Action: Check the specified filename.

SQL*Loader-504

SQL*Loader-504   error skipping records in file *name*

Cause: SQL*Loader could not open the file or could not read from it.

Action: Check that the file is where it is expected and that read access has been granted.

SQL*Loader-506

SQL*Loader-506   unable to open bad file *name*

Cause: SQL*Loader could not open the named file.

Action: Check the errors below it in the log file for more information.

SQL*Loader-507

SQL*Loader-507   unable to open discard file *name*

Cause: SQL*Loader could not open the named file.

Action: Check the errors below this message in the log file for more information.

SQL*Loader-510

SQL*Loader-510   physical record in datafile *name* is longer than the maximum *num*

Cause: The datafile has a physical record that is too long.

Action: Use CONCATENATE or CONTINUEIF. Break up the physical records.

SQL*Loader-511

SQL*Loader-511   unable to initialize read functions

Cause: SQL*Loader could not initialize the read functions.

Action: Check the errors below this message in the log file for more information.

SQL*Loader-512

SQL*Loader-512   unable to free read buffer

Cause: An internal error has occurred.

Action: Contact customer support.

SQL*Loader-513

SQL*Loader-513   unable to close file *name*

Cause: SQL*Loader could not close the named file.

Action: Check the errors below this message in the log file for more information.

SQL*Loader-514

SQL*Loader-514 error getting elapsed time

Cause: SQL*Loader could not get the elapsed time from the system.

Action: No action required.

SQL*Loader-515

SQL*Loader-515   error getting CPU time

Cause: SQL*Loader could not get the CPU time from the system.

Action: No action required.

SQL*Loader-516

SQL*Loader-516   control file *name* has no contents

Cause: The named file was found to be empty.

Action: Check that the intended file was referenced and that it is not empty.

SQL*Loader-517

SQL*Loader-517   error decomposing filename *name*

Cause: SQL*Loader could not break down the filename into its component parts.

Action: Check the filename for illegal characters.

SQL*Loader-518

SQL*Loader-518   error reassembling filename *name*

Cause: SQL*Loader could not put the filename back together again fromits components.

Action: Check the filename for illegal characters.

SQL*Loader-519

SQL*Loader-519   error *num* writing to filename *name*

Cause: SQL*Loader could not write to the named file. The error number displayed in the message is the one returned by the C language FWRITE function. More information can be obtained by accessing a C language reference manual forthe system.

Action: Check that the file's location is where it is expected to be and that write privileges on it have been granted. Be sure that adequate disk space is available and that the disk quota is not exceeded.

00600-00619: Errors Occurring During the Load

SQL*Loader-600

SQL*Loader-600   specified max. bind size *num* bytes must be *num* bytes to hold 1 row

Cause: The space needed to contain a single row is longer than the specified bind size.

Action: One solution is to increase the amount of space allowed for the bind array, using the BINDSIZE keyword on the command line or in the OPTIONS statement in the control file.

If the row contains VARCHAR or VARGRAPHIC data, specifying a maximum length for these fields can also reduce the amount of memory needed to buffer a row.

SQL*Loader-601

SQL*Loader-601   for INSERT option, table must be empty. error on *name*

Cause: A non-empty table is being loaded with the INSERT option.

Action: Use the REPLACE keyword to empty the old table and store the new data in its place. Use the APPEND keyword to leave the table's contents intact and add the new data to it.

SQL*Loader-602

SQL*Loader-602   **maximum converted length *num* of column *name.name* is too big forbind array**

Cause: The named column cannot undergo the specified character set conversion and still be placed in a contiguous piece of memory on the system.

Action: Specify a shorter data column or eliminate the conversion.

SQL*Loader-603

SQL*Loader-603   maximum length *num* of column *name.name* is too big for bind array

Cause: The named column cannot be put in a contiguous piece of memory onthe system.

Action: Specify a shorter data column.

SQL*Loader-604

SQL*Loader-604   error occurred on an attempt to commit

Cause: An error occurred while trying to commit changes to the database.

Action: Look at the message listed below this one in the log file for more information.

SQL*Loader-605

SQL*Loader-605   non-data dependent Oracle error occurred load discontinued

Cause: An error occurred that is independent of the data. For example, an out-of-space condition. Because all further rows will be rejected, theload is discontinued. (If the error were data dependent, then otherrows might succeed.)

Action: See the errors below this one in the log file for more information.

SQL*Loader-606

SQL*Loader-606 **synonym** *name* **refers to an object on a remote database**

Cause: The synonym specified in the INTO TABLE clause in the SQL*Loader control file specifies a remote object via a database link. Only a synonym for an existing local table can be specified in the INTO TABLE clause.

Action: Ensure that a local table name or a synonym for a local table is fully specified in the INTO TABLE clause. The table must already exist.

00620-00649: Formatting

These are errors in data described by free-format (TERMINATED, ENCLOSED) statements in the control file. Format errors occur when the data does not meet format specifications and might cause the loading of incorrect data into the database. A format error causes the record to be written to the bad file because an attempt to insert data failed.

SQL*Loader-620

SQL*Loader-620 initial enclosing character not found

Cause: A mandatory initial enclosure delimiter was not present. Either it is missing on the current field or the previous field is missing an identical closing delimiter.

Action: Supply the missing delimiters.

SQL*Loader-621

SQL*Loader-621   field in datafile exceeded maximum specified length

Cause: Delimited data was specified with a maximum length, and the data value exceeded that length.

Action: Check for missing delimiters and/or shorten the field.

SQL*Loader-622

SQL*Loader-622   column not found before end of logical record (use TRAILING NULLCOLS)

Cause: The logical record ended before all specified fields were found.

Action: If the missing fields should be loaded as null, use the TRAILING NULLCOLS clause. Otherwise, correct the data.

SQL*Loader-623

SQL*Loader-623   logical record ended − second enclosure character not present

Cause: The logical end of record occurred before a second enclosure delimiterwas found.

Action: Supply the missing delimiter.

SQL*Loader-624

SQL*Loader-624   no terminator found after TERMINATED and ENCLOSED field

Cause: No termination delimiter was found after the enclosed field ended.

Action: Supply the missing termination delimiter.

SQL*Loader-625

SQL*Loader-625   multi-byte character error in control file

Cause: Incomplete multi-byte character strings were found in the control file.

Action: Edit the control file to check that all multi-byte character data is valid.

SQL*Loader-640

SQL*Loader-640   variable length field was truncated

Cause: The end of the logical record was encountered before the end of a variable length field. The length of each variable-length field is embedded in the field, so SQL*Loader knows that more data should have been present.

Action: Check the data for inadvertent truncation and verify the control file specifications against the log file − the field may be starting in thewrong place.

SQL*Loader-641

SQL*Loader-641   invalid zoned decimal nibble

Cause: Each byte (character) in a zoned decimal field contains two 4-bit nibbles. Each nibble must have a valid value.

Action: Check the control file's specifications against the log file to ensure that the field location was specified correctly. Check the contents of the field.

SQL*Loader-642

SQL*Loader-642   relative start position > absolute field end position

Cause: A field specified as POSITION(*+n:y) had its relative start occur after the absolute position y.

Action: Check that a delimiter is not missing and that the values of *n* and *y* are correct.

00700-00799: Fatal errors

SQL*Loader-700

SQL*Loader-700   out of memory while performing essential allocations *num*

Cause: SQL*Loader could not allocate the memory it needs.

Action: Check the operating system and process memory. Increase memory available to SQL*Loader if possible. Otherwise, wait until more memory becomes available.

SQL*Loader-701

SQL*Loader-701   out of memory while allocating bind array

Cause: SQL*Loader could not allocate memory for the bind array.

Action: Check the operating system and process memory. Increase memory available to SQL*Loader if possible. Otherwise, wait until more memory becomes available.

SQL*Loader-702

SQL*Loader-702   internal error - *str*

Cause: An internal error has occurred.

Action: Make a note of the message and contact customer support.

SQL*Loader-703

SQL*Loader-703   internal error: argument *num*

Cause: An internal error has occurred.

Action: Make a note of the message and the number, then contact customer support.

SQL*Loader-704

SQL*Loader-704   internal error: *str num*

Cause: An internal error has occurred.

Action: Make a note of the message and the number, then contact customer support.

SQL*Loader-705

SQL*Loader-705   internal error

Cause: An internal error has occurred.

Action: Contact customer support.

SQL*Loader-706

SQL*Loader-706   bad argument to ulerr *num*

Cause: An internal error has occurred.

Action: Contact customer support.

00900-00949: Direct Path Load Preparation

SQL*Loader-901

SQL*Loader-901   error logging on to Oracle

Cause: An attempt was made to log on to Oracle in direct path load mode.

Action: Ensure that the instance is up and running. Check the Oracle messages below this one in the log file for more information.

SQL*Loader-902

SQL*Loader-902   error opening cursor: *num*

Cause: An internal error has occurred.

Action: Check the Oracle messages below this one in the log file and contactcustomer support.

SQL*Loader-903

SQL*Loader-903   database must be at least version *num* for direct path

Cause: The direct path load mode is being used with an incompatible database.

Action: Upgrade the database to the specified version or else use the conventionalpath load.

SQL*Loader-904

SQL*Loader-904   you cannot use the direct path when linked with v5 of the database

Cause: The direct path mode is being used after SQL*Loader has been linked with a Version 5 database.

Action: Use the conventional path.

SQL*Loader-905

SQL*Loader-905   error parsing SQL statement for upi: *num*

Cause: Header message.

Action: Check the message below this one in the log file for more information.

SQL*Loader-906

SQL*Loader-906   error executing SQL statement for upi: *num*

Cause: An internal error has occurred.

Action: Check the message below this one in the log file for more information.

SQL*Loader-907

SQL*Loader-907   error parsing insert statement on table *name*

Cause: Loader cannot insert into the specified table.

Action: Check that the table exists and that insert privileges on it have been granted.

SQL*Loader-908

SQL*Loader-908   unable to lock table *name* in exclusive mode due to Oracle error *num*

Cause: SQL*Loader could not get a DML exclusive lock on the table it needed.

Action: Check the message below this one in the log file for more information.Find the object number of the table in the catalog view USER_OBJECTS.Use the SQL*DBA MONITOR LOCK command to list all processes with outstanding locks. Identify the process that has the lock on the table.Then use the MONITOR PROCESS command to find out who is connected as the Oracle process holding the lock and get them to relinquish it, or simply wait and retry.

SQL*Loader-909

SQL*Loader-909   loader views must be loaded (as SYS) for direct path to be used

Cause: Database views required for the direct path mode are not present.

Action: The *Oracle7 Server Administrator's Guide* details how to run the required scripts as user SYS for loading the required views.

SQL*Loader-910

SQL*Loader-910   error binding input variables of upi: *num*

Cause: Header message.

Action: Check the message below this one in the log file for more information.

SQL*Loader-911

SQL*Loader-911   error defining output variables for upi: *num*

Cause: Header message.

Action: Check the message below this one in the log file for more information.

SQL*Loader-912

SQL*Loader-912   tables loaded through the direct path may not be clustered

Cause: A direct path load mode is being used to put data into a clustered table.

Action: Use the conventional path.

SQL*Loader-913

SQL*Loader-913   error fetching results of select statement (upi): *num*

Cause: Header message.

Action: Check the message below this one in the log file for more information.

SQL*Loader-914

SQL*Loader-914   error locking table *name* in exclusive mode

Cause: The direct path load mode could not lock the named table.

Action: Check the message below this one in the log file for more information. Check that the necessary privileges have been granted.

SQL*Loader-915

SQL*Loader-915   error closing cursor: *num*

Cause: Header message.

Action: Check the message below this one in the log file for more information.

SQL*Loader-916

SQL*Loader-916   error checking path homogeneity

Cause: To do a direct path load load when the front end and back end of the database are on separate hardware systems, the systems must be identical, using the same machine, operating system, and character set. This error occurs when SQL*Loader detects a difference between the systems that will prevent a direct load from operating properly.

Action: Use a conventional path load for this configuration.

SQL*Loader-917

SQL*Loader-917   error parsing insert statement on table *name*

Cause: For security purposes, SQL*Loader generates a SQL INSERT statement and uses it to verify that the load specifications make sense and that the proper access privileges exist. This message occurs when the INSERT statement cannot be used.

Action: Check the Oracle messages below this one in the log file for more information. Check that all the columns to be loaded exist and that insert privileges on the table exist. Finally, check any SQL strings defined for the named table.

SQL*Loader-918

SQL*Loader-918   the catalog must be loaded (as SYS) for SQL*Loader to be used

Cause: The catalog is not loaded.

Action: Logon as user SYS and load the Oracle7 catalog and the views in thescript ULVIEW.SQL.

SQL*Loader-919

SQL*Loader-919   error during upi fetch: *num*

Cause: An internal error has occurred.

Action: Check the message below this one in the log file for more information.

SQL*Loader-920

SQL*Loader-920   error deleting data from table *name* (due to REPLACE keyword)

Cause: The REPLACE option is specified in the control file but SQL*Loader could not delete the data from the table.

Action: Check the message below this one in the log file for more information.

SQL*Loader-921

SQL*Loader-921   NOT NULL column *name.name* has a NULLIF clause

Cause: A column that is NOT NULL in the database has a NULLIF clause in the control file.

Action: Remove the NULLIF clause.

SQL*Loader-922

SQL*Loader-922   NOT NULL column *name.name* has a default if clause which may evaluate to null

Cause: A column that is NOT NULL in the database has a NULLIF clause that may evaluate to null.

Action: Remove the NULLIF clause.

SQL*Loader-923

SQL*Loader-923   column *name* in table *name* is NOT NULL but is not being loaded

Cause: A column that is NOT NULL in the database is not being loaded, so every row will be rejected.

Action: Load the column.

SQL*Loader-925

SQL*Loader-925   Error while *str*

Cause: An internal error has occurred.

Action: Check the message below this one in the log file for more information. The table or column referenced may not be accessible.

SQL*Loader-926

SQL*Loader-926   OCI error while executing *name* for table *name*

Cause: An OCI error has occurred.

Action: Check the message below this one in the log file for more information. The table or column referenced may not be accessible.

SQL*Loader-927

SQL*Loader-927   table *name* does not exist

Cause: SQL*Loader could not find the named table. The specified table may not exist. On some systems, this message will also appear if the necessary privileges have not been granted.

Action: Check that that the table exists, its name is spelled properly, and that the necessary privileges on it have been granted.

SQL*Loader-928

SQL*Loader-928   column *name.name* does not exist

Cause: SQL*Loader could not find the named table.

Action: Check that the column exists, its name is spelled properly, and the necessary privileges on it have been granted.

SQL*Loader-929

SQL*Loader-929   error parsing insert statement for table *name*

Cause: The table's insert statement caused a parse error.

Action: Check all SQL strings that were used.

SQL*Loader-930

SQL*Loader-930   error parsing insert statement for column *name*

Cause: The named column's INSERT statement caused a parse error.

Action: Check the message below this one in the log file for more information. Check the SQL string used for this column.

SQL*Loader-931

SQL*Loader-931   OCI error while binding variable for column *name*

Cause: An OCI error has occurred.

Action: Check the errors below it for more information. The SQL string for this column may be in error.

SQL*Loader-932

SQL*Loader-932   could not truncate table *name*

Cause: Truncation of the table was attempted and failed.

Action: Check the following errors for more information. Truncation due to referential constraints should not occur.

SQL*Loader-933

SQL*Loader-933   specified file *name* not part of database

Cause: The specified filename to load the table is not a database file.

Action: Specify a valid datafile.

SQL*Loader-934

SQL*Loader-934   incorrect datafile *name* specified for table *tabnam*

Cause: A datafile name was given to load the table that is not part of the tablespace in which the table resides.

Action: Specify a valid datafile.

SQL*Loader-935

SQL*Loader-935   error verifying required option for parallel load

Cause: An error was encountered because a required option was not found orwas invalid.

Action: Correct the problem as described in the accompanying Oracle message.

00950-00999: Direct Path Load Error-Header Messages

These messages are informational. They describe the kind of error that has occurred. Check the messages below them in the log file for moredetailed information.

SQL*Loader-950

SQL*Loader-950   error shipping data structures across 2-task

SQL*Loader-951

SQL*Loader-951   error calling once/load initialization

SQL*Loader-952

SQL*Loader-952   error calling once/datafile initialization

SQL*Loader-953

SQL*Loader-953   error shipping read buffer across 2-task

SQL*Loader-954

SQL*Loader-954   error shipping index to read buffer across 2-task

SQL*Loader-955

SQL*Loader-955   error calling col array build, block formatter

SQL*Loader-956

SQL*Loader-956   error calling once/datafile finishing

SQL*Loader-957

SQL*Loader-957   error shipping text string across 2-task

SQL*Loader-958

SQL*Loader-958   error popping data relocation address stack

SQL*Loader-959

SQL*Loader-959   error shipping buffer across 2-task

SQL*Loader-960

SQL*Loader-960   error escalating DDL share lock to exclusive

SQL*Loader-961

SQL*Loader-961   error calling once/load finishing

SQL*Loader-962

SQL*Loader-962   error setting user-side Ctrl-C handler

SQL*Loader-963

SQL*Loader-963   error clearing user-side Ctrl-C handler

SQL*Loader-964

SQL*Loader-964   error logging off Oracle

SQL*Loader-965

SQL*Loader-965   error *num* disabling constraint *name* of table *name*

SQL*Loader-966

SQL*Loader-966   error *num* disabling all triggers on table *name*

SQL*Loader-967

SQL*Loader-967   **Oracle error:** *str*

SQL*Loader-968

SQL*Loader-968   error disconnecting from Oracle

SQL*Loader-969

SQL*Loader-969   error performing filename translation on kernel side

SQL*Module Messages

This chapter lists messages you might see when using SQL*Module. For more information on SQL*Module, see the SQL*Module User's Guide and Referenceorg. SQL*Module issues error messages like the following:K

MOD-F-0037: End of SQL statement or PL/SQL block not found

The error code consists of a prefix, severity code, and sequence number. The prefix MOD shows that the error was issued by SQL*Module. There are four severity codes; the following list gives their meanings:K

CodeMeaning#

W   Warning ¾ despite an error, a compilable output file was created.

S   Severe error ¾ despite an error, an output file was created. However, it might not be compilable. (Note: the file is created only if RPC_GENERATE=NO.)

F   Fatal error ¾ no output file was created because of an internal problem or because a resource (such as memory) was unavailable or ran out.

U   Unrecoverable error ¾ no output file was created because an input requirement was not met.   In addition to the errors generated by SQL*Module, you might encounter errors at runtime. These errors have the prefixes SQL- and ORA-. SQL errors are generated by the routines in the SQLLIB library; ORA errors are generated by the server.

00000-00100: SQL*Module Messages

SQL*Module Messages

Main:0126

 error messages, coping with

 error messages, interpreting codes

ada_sql_typ

00000-00100: SQL*Module Messages

Main:0127

m0075

MOD-00009

MOD-00009   unable to open input file name

Cause: SQL*Module is not able to open the input file that was specified with the INAME option. This error can result from any of the following:

- The file does not exist.
- The file exists, but in a directory not on the path specified.
- You do not have the permissions or access rights required to read the file.

Action: Check that the appropriate permission to read the file has been granted.Check that the name and directory hierarchy have been specified correctly. Rerun the compiler.

MOD-00010

MOD-00010   unable to open listing file name

Cause: SQL*Module is not able to create the listing output file. This usually happens when write permissions do not exist on the current directory or on the directory specified for the LNAME option. It can also happen if the name or directory path is not legal or a directory on the path does not exist.

Action: Check the permissions on the specified path and file, or on the current directory if the LNAME option is not used.

MOD-00011

MOD-00011   **unable to open output file name**

Cause: SQL*Module is not able to create an output file, such as the specification file or the output code file. This usually happens when write permissions on the current directory or on the directory specified for the ONAME or SNAME options do not exist. It can also happen if the name or directory path is not legal or a directory on the path does not exist.

Action: Check the permissions on the specified path and file, or on the current directory if the ONAME or SNAME options are not used.

MOD-00012

MOD-00012 **encountered the symbol name when expecting one of the following...**

Cause: This is caused by a syntax error in the Module Language file. For example, if the SQL statement

```
SELECT ename INTO :name FORM emp WHERE ...
```

is in a module procedure, the following message appears:

MOD-S-00012, Encountered the symbol FORM where expecting one of the following:

,:INDICATOR FROM

This message is not triggered by semantic errors. See MOD-00075.

Action: Correct the syntax error, then recompile.

MOD-00015

Cause: This message results from an internal error: an invalid value for the mode of a bind variable.

Action: Contact customer support.

MOD-00016

MOD-00016   **out of space - unable to allocate num bytes**

Cause: SQL*Module could not allocate enough memory to run or to perform some required action.

Action: Try killing some non-essential processes, then rerunning SQL*Module. On some operating systems, there may be a quota for the amount of memory the processes can allocate. See the system administrator. If necessary, increase memory for the machine.

MOD-00017

MOD-00017   unable to connect to Oracle with conn_str:username/password. Oracle error number ora-nnnnn

Cause: The database connection or the username or the password that was specified was not correct. This message can result from an incorrect command-line specification or an incorrect specification of the username and/or password in the module AUTHORIZATION clause.

Action: Look up the error number for more specific information on the failure, correct the identifiers, then rerun SQL*Module. (To find information on the error number, click the Help Topics button in this window, click the Index tab, then type the error number, including its three-letter prefix with the hyphen.)

MOD-00027

MOD-00027   no input file name specified

Cause: The STORE_PACKAGE=YES option was entered, but an input file using the INAME option was not specified.

Action: Use INAME on the command line.

MOD-00036

MOD-00036   internal programming exception

Cause: An internal error occurred.

Action: Contact customer support.

MOD-00038

MOD-00038   module name required when output language is Ada

Cause: A MODULE clause was not specified in the module source file, or a name for the module was not provided.

Action: Specify the module name in the module file.

MOD-00039

MOD-00039   unable to open specifications file name

Cause: SQL*Module is not able to create the specifications (header) output file. This usually happens when write permissions on the current directory or on the directory specified for the SNAME option do not exist. It can also happen if the name or directory path is not legal, a directory on the path does not exist, or the disk or file system is full.

Action: Check the permissions on the specified path and file or on the current directory if the SNAME options are not used. Check that the disk or file system is not full.

MOD-00041

MOD-00041   ANSI violation at line num, column colnam: text

Cause: FIPS=YES was specified, and part of the module code was not compliant with the ANSI SQL92 Entry Level standard.

Action: Specify FIPS=NO (the default) or correct the code.

MOD-00042

MOD-00042   cursor name is multiply declared

Cause: The cursor named in the message was declared more than once in themodule file.

Action: Check that each cursor names (not just in the module file, but in the entire application) is unique.

MOD-00043

MOD-00043   **procedure name is multiply declared**

Cause: A procedure named X was declared more than once in the module file.

Action: Check that each procedure name (not just in the module file, but in the entire application) is unique.

MOD-00044

MOD-00044   **parameter param in procedure name is multiply declared**

Cause: In the procedure named in the message, a parameter was declared with a name that duplicates another parameter's name.

Action: Check that each parameter name in a procedure is unique.

MOD-00046

MOD-00046 **cursor name has not been declared**

Cause: In the module file, an OPEN, FETCH, or CLOSE was attempted on a cursor that had not yet been declared.

Action: Insert an appropriate cursor declaration in the module file. Remember that cursors must be declared in the file before they are acted upon.

MOD-00047

MOD-00047   an open statement already exists for cursor name

Cause: More than one procedure in this module contains an OPEN command on the named cursor.

Action: Remove one of the OPEN commands, as there is no need for redundancy.

MOD-00048

MOD-00048   cursor name was never opened

Cause: No OPEN command exists in the module for the cursor named in this message.

Action: Add an OPEN command or remove the cursor if it is not needed.

MOD-00049

MOD-00049   unable to initialize PL/SQL

Cause: This message usually results when SQL*Module cannot connect to the database, using the username and password mentioned either in the module's AUTHORIZATION clause or on the command line with the USERID option.

Action: If connection is made using SQL*Net, check that a listener process is running and that the service name specified in the AUTHORIZATION clause or on the command line is in the tnsnames.ora file. If the connect parameters are correct and this problem persists, contact customer support.

MOD-00050

MOD-00050   formal parameter param is not used in procedure

Cause: A formal parameter was declared in a procedure, but never used.

Action: Remove the parameter declaration or use it in the procedure.

MOD-00054

MOD-00054   name is not a valid cursor reference

Cause: An attempt was made to OPEN a cursor that had not been declared.

Action: Declare the cursor or remove the erroneous reference to it.

MOD-00059

MOD-00059   **INTO variable var was not listed as a formal parameter**

Cause: The target of an INTO clause in a SELECT statement is not a parameter ofthe procedure

Action: Declare the parameter using the correct datatype in the procedure'sparameter list.

MOD-00060

MOD-00060 **PL/SQL error in compiling package specification**

Cause: A semantic error occurred in the PL/SQL parse phase when running SQL*Module.

Action: Check that all database objects are valid. To get more information, rerun SQL*Module with OUTPUT=PACKAGE, and check the output PL/SQL code specification file (it has a .pks extension). See the PL/SQL User's Guide and Reference for more information.

MOD-00061

MOD-00061   PL/SQL error in compiling package body

Cause: A semantic error occurred in the PL/SQL parse phase when running SQL*Module.

Action: Check that all database objects are valid. To get more information, rerun SQL*Module with OUTPUT=PACKAGE, and check the output PL/SQL code file (it has a .pkb extension). See the PL/SQL User's Guide and Reference for more information.

MOD-00062

MOD-00062   host variable for indicator is not recognized

Cause: A host variable that was used as an indicator variable was not declared.

Action: Declare the host variable or, if possible, remove the indicator variable from the SQL statement.

MOD-00063

MOD-00063   **procedure name, parameter param: illegal interface type type**

Cause: When attempting to generate an interface procedures (stubs) file, the WITH INTERFACE PROCEDURE clause in the referenced stored package or stored procedure contained a parameter having an illegal datatype.

Action: Correct the datatype. Datatypes in a WITH INTERFACE clause must be SQL datatypes, not PL/SQL datatypes.

MOD-00064

MOD-00064   procedure name: extra parameter param in interface

Cause: An unused parameter was declared in the WITH INTERFACE clause of a stored package or a stored procedure.

Action: Correct the WITH INTERFACE clause in the package.

MOD-00065

MOD-00065   **interface definition missing for procedure name**

Cause: An attempt was made to generate stubs for a package that does not contain a WITH INTERFACE clause for one or more procedures.

Action: Add the WITH INTERFACE clause to the package spec.

MOD-00066

MOD-00066   **unable to load package name from database**

Cause: The package name that was specified using the PNAME option is not correct, or SQL*Module could not access the database schema required.

Action: Check that the specified package exists and has a valid status. Check the STATUS column of the USER_OBJECTS table for that schema to check that the package is valid.

MOD-00067

MOD-00067   package name is required but not specified

Cause: The options and values RPC_GENERATE=YES and STORE_PACKAGE=YES were specified, but a package name using the PNAME option was not specified.

Action: Add the PNAME option.

MOD-00068

MOD-00068   could not find or open system configuration file

Cause: This is a warning message. SQL*Module is not able to find the system configuration file or is not able to open the file. The location of the system configuration file is system specific. See your operating system-specific documentation for the location on your system.

Action: Check that the file exists and that read permission or the appropriate access rights are set for it. The file must exist to avoid this warning message; however, it can be empty.

MOD-00069

MOD-00069   Oracle error N while attempting to store package

Cause: SQL*Module's attempt to store the package in the database failed.

Action: Look up the error number in this manual and correct the problem.

MOD-00070

MOD-00070   procedure name interface is missing declaration for indvar

Cause: An indicator variable was specified in the WITH INTERFACE clause, but the associated main variable was not declared there.

Action: Modify the WITH INTERFACE clause. Either remove the indicator variable or declare a host variable for it.

MOD-00072

MOD-00072   PNAME option is specified but not meaningful

Cause: The PNAME option was used when there was no need for it. The PNAME option is only required when you specify RPC_GENERATE, STORE_PACKAGE, or both.

Action: Correct the command line.

MOD-00073

MOD-00073   **INAME option is specified but not meaningful**

Cause: The INAME option was used when it was not required. For example, INAME is not required when RPC_GENERATE=YES and the PNAME option are specified.

Action: Do not use the INAME option.

MOD-00074

MOD-00074   **ONAME option is specified but not meaningful**

Cause: The ONAME option was used when it was not required. For example, the ONAME option is not required when RPC_GENERATE=NO, and STORE_PACKAGE=YES, or the OUTPUT values list does not contain theCODE value.

Action: Do not use the ONAME option.

#

MOD-00075

MOD-00075   error at line N. PLS-N: text

Cause: A semantic error occurred in the PL/SQL parse phase when running SQL*Module.

Action: Check that all database objects are valid. To get more information, rerun SQL*Module with OUTPUT=PACKAGE, and check the output PL/SQL code file (it has a .pkb extension). See the PL/SQL User's Guide and Reference for more information

MOD-00076

MOD-00076   ANSI violation at line N. PLS-N: text

Cause: FIPS=YES was specified and part of the module code is not compliant with the ANSI SQL92 Entry Level standard.

Action: Specify FIPS=NO (the default) or correct the code.

MOD-00077

MOD-00077   mandatory parameter SQLCODE or SQLSTATE is missing

Cause: Include a status parameter ¾ SQLCODE, SQLSTATE, or both ¾ in every parameter definition.

Action: Include SQLCODE or SQLSTATE.

MOD-00078

MOD-00078   must specify a user if generating stubs from stored package

Cause: Because no USERID was specified, SQL*Module cannot connect to the server to create stubs (an interface procedures file) from the stored package.

Action: Specify the service name (if not connecting to the default database), and the username and password. Do this either on the command line or in a configuration file.

MOD-00079

MOD-00079   SQLCHECK must be SEMANTICS when RPC_GENERATE=yes

Cause: SQLCHECK=SYNTAX (or SQLCHECK=NONE) cannot be specified when creating a stubs file.

Action: Remove the SQLCHECK=SYNTAX or SQLCHECK=NONE specification from the command line or the configuration file.

MOD-00080

MOD-00080   error extracting stored package source.

Cause: OUTPUT=PACKAGE was requested on the command line or in a configuration file, and a package of the name specified does not exist in the schema or is marked as INVALID.

Action: Check that the named package exists and is valid.

MOD-00081

MOD-00081   failed assertion

Cause: Some part of the input to SQL*Module caused an error or warning condition. This message always follows a specific error or warning message.

Action: No action required for this message. Correct the previous errors.

MOD-00086

MOD-00086   **error: couldn't install username. Aborted...**

Cause: The server rejected a username.

Action: If the username is valid, contact customer support.

MOD-00088

MOD-00088   feature not implemented yet

Cause: SQL*Module encountered a SQL statement that it can compile, but that uses a feature not yet implemented.

Action: Contact Oracle Languages Product Management.

MOD-00093

MOD-00093 **SNAME option is specified but not meaningful**

Cause: The SNAME option was used when it was not required. For example, no SNAME option is required if the values list of the OUTPUT option does not include the value SPECIFICATION.

Action: Do not use the SNAME option.

MOD-00094

MOD-00094   extension in PNAME option is specified but not meaningful

Cause: A filename extension was included in the value of a PNAME option. Packages are database objects, not files.

Action: Remove the extension.

MOD-00095

MOD-00095 **Datatype "X" is not valid for the host language**

Cause: A datatype was used that is not supported for the host language. For example, NUMERIC is not a supported datatype for Ada and C.

Action: Refer to the Programmer's Guide to the Oracle Precompilers for a list of the datatypes supported in each host language.

MOD-00096

MOD-00096   LNAME option is specified but not meaningful

Cause: The OUTPUT option list does not contain the value LIST.

Action: Add LIST to the list of values of the OUTPUT option.

MOD-00097

MOD-00097 **WITH INTERFACE must be specified for this host language**

Cause: A PL/SQL package or function specification did not have a WITH INTERFACE clause. Currently, MODADA requires the WITH INTERFACE clause to be present in order to create calls to a PL/SQL function or package.

Action: Rewrite PL/SQL package or function specification to include a WITH INTERFACE clause. Alternatively, write a PL/SQL package or function specification which has a WITH INTERFACE clause and have it call the original PL/SQL package.

MOD-00098

MOD-00098 WITH INTERFACE may only be defaulted when bind=late

Cause: A Command line option, "bind=early" was specified while creating calls to a PL/SQL package or function without a WITH INTERFACE clause.

Action: Change the command line option to "bind=late", or include a WITH INTERFACE clause in the PL/SQL specification (see error 97 action).

MOD-00099

MOD-00099   **type name is invalid when storing a module (store=yes)**

Cause: The module contained a type for which there is currently no equivalentPL/SQL type; it may not be stored in the database.

Action: Set the command line option to STORE=NO or move the procedurescontaining the offending type to another module and compile that modulewith STORE=NO.

Oracle Database Messages (ORA-09200 to ORA-09499)

These are Oracle database messages. Each is triggered by an Oracle operating-system-dependent (OSD) message specific to Windows. When you receive one of these messages, look up the accompanying OSD message number for an explanation of the error. (To find information on the error number, click the Help Topics button in this window, click the Index tab, then type the error number, including its three-letter prefix with the hyphen.)

ORA-09200   sfccf: error creating file

ORA-09201   sfcopy: error copying file

ORA-09202   sfifi: error identifying file

ORA-09203   sfofi: error opening file

ORA-09204   sfotf: error opening temporary file

ORA-09205   sfqio: error reading or writing to disk

ORA-09206   sfrfb: error reading from file

ORA-09207   sfsrd: error reading from file

ORA-09208   sftcls: error closing file

ORA-09209   sftget: error reading from file

ORA-09210   sftopn: error opening file

ORA-09211   sfwfb: error writing to file

ORA-09212   sfwfbmt: error writing to file

ORA-09213   slgfn: error fabricating file name

ORA-09214   sfdone: I/O error detected

ORA-09215   sfqio: error detected in I/O completion routine

ORA-09216   sdnfy: bad value `%s' for parameter %s

ORA-09217   sfsfs: failed to resize file

ORA-09218   sfrfs: failed to refresh fie size

ORA-09240   smpalo: error allocating PGA memory

ORA-09241   smsalo: error allocating SGA memory

ORA-09242   smscre: error creating SGA

ORA-09243   smsget: error attaching to SGA

ORA-09244   smprset: error setting memory protections

ORA-09245   smcstk: error switching stacks

ORA-09246   sfsmap: unable to map SGA

ORA-09247   smsdes: error destroying the SGA

ORA-09260   sigpidu: error obtaining process id

ORA-09261   spdcr: error creating detached (background) process

ORA-09262   spdde: error terminating detached (background) process

ORA-09263   spini: error initializing process

ORA-09264   sptpa: error flagging process

ORA-09265   spwat: error temporarily suspending process

ORA-09266   spawn: error starting an oracle process

ORA-09270   szalloc: error allocating memory for security

ORA-09271   szlon: error verifying user name

ORA-09272   remote OS login is not allowed

ORA-09273   szrfc: error verifying role name

ORA-09274   szrfc: insufficient role name buffer space

ORA-09280   sllfcf: error closing file

ORA-09281	sllfop: error opening file
ORA-09282	sllfrb: error reading records
ORA-09283	sllfsk: error skipping records
ORA-09290	sksaalo: error allocating memory for archival
ORA-09291	sksachk: invalid device specification for LOG_ARCHIVE_DEST
ORA-09292	sksabln: unable to build archive file name
ORA-09293	sksasmo: unable to send message to console
ORA-09300	osncon: unable to connect, DPMI not available
ORA-09301	osncon: local kernel only supported in standard mode
ORA-09310	sclgt: error freeing latch
ORA-09311	slsleep: error temporarily suspending process
ORA-09312	slspool: error spooling file to printer
ORA-09313	slsprom: error prompting user
ORA-09314	sltln: error translating logical name
ORA-09315	sql2tt: two-task error translating ORACLE_EXECUTABLE
ORA-09316	szrpc: unable to verify password for role
ORA-09317	szprv: insufficient privileges
ORA-09318	slkhst: error hosting out to operating system
ORA-09319	slgtd: unable to obtain the current date and time
ORA-09320	szrfc: unable to obtain the list of valid OS roles
ORA-09321	slzdtb: unable to convert zoned decimal to binary
ORA-09322	slpdtb: unable to convert packed decimal to binary
ORA-09330	session terminated internally by Oracle or by an Oracle DBA
ORA-09331	scgcan: unable to process lk_sync_cancel return code
ORA-09332	scgcc: unable to process lk_close return code
ORA-09333	scggc: unable to process lk_open_convert return code
ORA-09334	scggc: unable to process lk_convert return code
ORA-09335	scgcm: unabel to process return code in completion procedure
ORA-09340	specified ORACLE_SID is either invalid or too long
ORA-09341	scumnt: unable to mount database
ORA-09350	Windows NT two-task driver unable to allocate context area
ORA-09351	Windows NT two-task driver unable to allocate shared memory
ORA-09352	Windows NT two-task driver unable to spawn new Oracle task
ORA-09353	Windows NT two-task driver unable to open event semaphore
ORA-09354	Windows NT two-task driver: Oracle task unexpectedly died
ORA-09360	Windows 3.1 Two-Task driver unable to allocate context area
ORA-09361	Windows 3.1 Two-Task driver unable to lock context area
ORA-09362	Windows 3.1 Two-Task driver unable to deallocate context area
ORA-09363	Windows 3.1 Two-Task driver invalid context area
ORA-09364	Windows 3.1 Two-Task driver unable to create hidden window
ORA-09365	Windows 3.1 Two-Task driver unable to destroy hidden window
ORA-09366	Windows 3.1 Two-Task driver unable to allocate shared memory
ORA-09367	Windows 3.1 Two-Task driver unable deallocate shared memory
ORA-09368	Windows 3.1 Two-Task driver unable to spawn ORACLE

ORA-09369   Windows 3.1 Two-Task driver bad instance handle

ORA-09370   Windows 3.1 Two-Task driver ORACLE task timed out

Memory Errors (01100 to 01199)

OSD-01100

OSD-01100   **Malloc() failure, unable to allocate memory**

Cause: Out of memory

Action: Free up memory, terminate processes, install more memory in machine

OSD-01102

OSD-01102	Free() failure, unable to free allocated memory

Cause: Unexpected return from Windows system service free()

Action: Check OS error code and consult Windows documentation

OSD-01103

OSD-01103   Unable to allocate memory

Cause: Out of memory

Action: Free up memory, terminate processes, install more memory in machine

OSD-01104

OSD-01104   **Unable to attach to SGA: SGA does not exist**

Cause: No SGA exists to connect to

Action: Startup ORACLE instance

OSD-01105

OSD-01105   Invalid SGA: SGA not initialized

Cause: The SGA has been allocated but not yet initialized

Action: Try again later

OSD-01106

OSD-01106   **Unable to attach to variable part of SGA**

Cause: Shared memory containing variable part of SGA not available

Action: Verify that SGA has been created w/o error

OSD-01107

OSD-01107   **Unable to attach to database buffers area of SGA**

Cause: Shared memory containing database buffers not available

Action: Verify that SGA has been created w/o error

OSD-01108

OSD-01108   **Unable to attach to redo buffer area of SGA**

Cause: Shared memory containing redo buffers not available

Action: Verify that SGA has been created w/o error

OSD-01109

OSD-01109   **Invalid memory protection mode specified**

Cause: A mode other than read-only and read-write was specified for memory protection

Action: Contact your Oracle Customer Support Representative

OSD-01110

OSD-01110   **VirtualProtect() failure, unable to set memory protections**

Cause: Unable to set the memory protections for a database block in the SGA

Action: Check OS error code and consult Windows documentation

OSD-01204

OSD-01204   Invalid pid

Cause: Process ID not recognized by system, process previously terminated

Action: Verify that process ID is correct and that process is active

OSD-01205

OSD-01205   **Unable to spawn process**

Cause: Scheduler could not start background process

Action: Check OS error code

Miscellaneous Errors (01500 to 01599)

OSD-01500

OSD-01500   **Illegal option specified**

Cause: Internal error

Action: Contact your Oracle Customer Support Representative

OSD-01501

OSD-01501   Internal buffer overflow

Cause: Internal error

Action: Contact your Oracle Customer Support Representative

OSD-01502

OSD-01502   **Translations nested too deep**

Cause: A CONFIG variable translation caused too many intermediate translations

Action: Simplify CONFIG variables

OSD-01503

OSD-01503   **Text contains no translatable elements**

Cause: No recognized variables are contained in the text to be translated

Action: This message is mostly informational

OSD-01505

OSD-01505   **Stdin not responding**

Cause: Unable to receive input from the standard input stream

Action: Verify that process has access to an input device

OSD-01506

OSD-01506   **Unable to spawn process via system()**

Cause: Out of resource or invalid executable name

Action: Free up memory, terminate processes, install more memory in machine, verify executable name

OSD-01507

OSD-01507   Password for 'internal' is incorrect

Cause: An attempt was made to connect as 'internal' with an invalid password

Action: Verify that the password is correct and try again

OSD-01508

OSD-01508   No password given

Cause: An attempt was made to connect as 'internal' without a password,

Action: Enter a valid password when connecting as internal

OSD-01509

OSD-01509   **No password found**

Cause: Oracle was unable to locate and retrieve the password for 'internal'

Action: Verify that Oracle is installed and configured correctly

OSD-01510

OSD-01510   Operating system roles are not supported

Cause: An attempt was made to use an OS Role

Action: Only use roles that were created 'IDENTIFIED BY password' as opposed to 'IDENTIFIED EXTERNALLY'

OSD-01511

OSD-01511   Unable to get date and time from the operating system

Cause: Unexpected return from localtime() call

Action: Verify that the system time is correct on the computer

OSD-01512

OSD-01512   Unable to translate the 'USERNAME' configuration variable

Cause: In order for os logins to work, the 'USERNAME' configuration variable must be set on the database machine

Action: Verify that the 'USERNAME' variable is set

OSD-01513

OSD-01513   **'Remote_os_authent' init.ora variable not set to true**

Cause: In order for os logins to work, 'REMOTE_OS_AUTHENT' must be set to 'TRUE' in INIT.ORA.

Action: Shutdown and startup the instance with 'REMOTE_OS_AUTHENT = TRUE' in INIT.ORA

Codes 4000-4999: Windows NT-Specific Oracle Messages

The messages in this section are Oracle operating-system-dependent (OSD) messages that issue in response to an error condition in Windows NT. Each message in this section triggers an Oracle database message (listed in the preceding section).

OSD-04000

OSD-04000   **logical block size mismatch**

Cause: The database block size specified in the initialization parameter file does not match the block size of the actual database files.

Action: Use matching logical block sizes.

OSD-04001

OSD-04001   **invalid logical block size**

Cause: The logical block size is not a multiple of 512 bytes, or it is too large.

Action: Change the value of DB_BLOCK_SIZE in the initialization parameter file.

OSD-04002

OSD-04002   **unable to open file**

Cause: The specified path or filename is invalid, or the destination device is full. This error can also be caused by insufficient Windows NT file handles.

Action: Make sure that the path and file exist, and that the device has free space. If this fails, increase the number of Windows NT file handles.

OSD-04003

OSD-04003   **unable to read file header block**

Cause: The media has been damaged.

Action: Recover the file if necessary, and verify that Windows NT is functioning correctly.

OSD-04004

OSD-04004 invalid file header

Cause: The file is corrupted.

Action: Recover the file.

OSD-04005

OSD-04005 **SetFilePointer() failure, unable to read from file**

Cause: There was an unexpected return from the Windows NT system service, SetFilePointer().

Action: Check the operating system error code and consult the Windows NT documentation.

OSD-04006

OSD-04006   ReadFile() failure, unable to read from file

Cause: There was an unexpected return from the Windows NT system service, ReadFile().

Action: Check the operating system error code and consult the Windows NT documentation.

OSD-04007

OSD-04007   truncated read

Cause: The system encountered an unexpected end-of-file, which is due to damaged media.

Action: Verify that the file is not damaged.

OSD-04008

OSD-04008   WriteFile() failure, unable to write to file

Cause: There was an unexpected return from the Windows NT system service, WriteFile().

Action: Check the operating system error code and consult the Windows NT documentation.

OSD-04009

OSD-04009   truncated write

Cause: The destination device is full or the media is damaged.

Action: Verify that the device has free space and the file is not damaged.

OSD-04010

OSD-04010   option specified, file already exists

Cause: The file that you attempted to create already exists.

Action: Delete the existing file or use the REUSE option in the SQL statement.

OSD-04011

OSD-04011   GetFileInformationByHandle() failure, unable to obtain file info

Cause: There was an unexpected return from the Windows NT system service, GetFileInformationByHandle().

Action: Check the operating system error code and consult the Windows NT documentation.

OSD-04012

OSD-04012   file size mismatch

Cause: The file to be reused is either too large or too small

Action: Specify the correct file size or delete the existing file.

OSD-04013

OSD-04013   **unable to read line from file**

Cause: This error is caused by an operating system error or by damaged media.

Action: Check the operating system error code (if available) and consult the Windows NT documentation. If no operating system error code is presented, verify that the media is not damaged.

OSD-04014

OSD-04014   **unable to close file**

Cause: The media has been damaged.

Action: Recover the file, if necessary, and verify that Windows NT is functioning correctly.

OSD-04015

OSD-04015   asynchronous I/O request returned an error

Cause: There was an unexpected return from the Windows NT system service.

Action: Check the operating system error code and consult the Windows NT documentation.

OSD-04016

OSD-04016   **error queuing an asynchronous I/O request**

Cause: There was an unexpected return from the Windows NT system service.

Action: Check the operating system error code and consult the Windows NT documentation.

OSD-04017

OSD-04017   unable to open the specified RAW device

Cause: An invalid path or filename was specified or the device is full.

Action: Make sure the file exists and/or device is not full; verify that the operating system is functioning correctly.

OSD-04018

OSD-04018   **unable to access the specified directory or device**

Cause: An invalid path name was specified.

Action: Make sure the directory or device exists and is accessible.

OSD-04100

OSD-04100   malloc() failure, unable to allocate memory

Cause: The program is out of memory.

Action: Shutdown all unnecessary processes or install more memory in the machine.

OSD-04101

OSD-04101   **invalid SGA: SGA not initialized**

Cause: The System Global Area (SGA) has been allocated but not initialized.

Action: Wait until the STARTUP has completed before attempting to connect.

OSD-04102

OSD-04102   **unable to open/create file for shared memory object**

Cause: There was an unexpected return from the Windows NT system service, CreateFile().

Action: Check the operating system error code and consult the Windows NT documentation.

OSD-04103

OSD-04103   unable to attach to SGA: SGA does not exist

Cause: The System Global Area (SGA) does not exist.

Action: Start up an Oracle instance.

OSD-04104

OSD-04104   **unable to map shared memory (SGA) into the address space**

Cause: There was an unexpected return from the Windows NT system service, MapViewOfFileEx().

Action: Check the operating system error code and consult the Windows NT documentation.

OSD-04105

OSD-04105   **shared memory (SGA) mapped to wrong address**

Cause: There was an unexpected return from the Windows NT system service, MapViewOfFileEx().

Action: Check the operating system error code and consult the Windows NT documentation.

OSD-04106

OSD-04106   **unable to allocate memory with VirtualAlloc**

Cause: The program is out of memory.

Action: Shutdown all unnecessary processes; install more memory in the machine.

OSD-04107

OSD-04107   **unable to deallocate memory with VirtualFree**

Cause: This is an internal error, not normally expected to occur.

Action: Contact Oracle Customer Support.

OSD-04200

OSD-04200   unable to begin another thread

Cause: The program has run out of system resources.

Action: Shutdown all unnecessary processes; install more memory in the machine.

OSD-04201

OSD-04201   no pid structure supplied to spdcr()

Cause: This is an internal error, not normally expected to occur.

Action: Contact Oracle Customer Support.

OSD-04202

OSD-04202   DosSetPriority() failure, unable to set process priority

Cause: There was an unexpected return from the Windows NT system service, DosSetPriority().

Action: Check the operating system error code and consult the Windows NT documentation.

OSD-04203

OSD-04203   DosKillProcess() failure, unable to kill process

Cause: There was an unexpected return from the Windows NT system service, DosKillProcess().

Action: Check the operating system error code and consult the Windows NT documentation.

OSD-04204

OSD-04204   invalid pid

Cause: Process id not recognized by system, process previously terminated.

Action: Verify that process ID is correct and that process is active.

OSD-04205

OSD-04205   **CreateProcess() failure, unable to spawn process**

Cause: There was an unexpected return from the Windows NT system service, CreateProcess().

Action: Check the operating system error code and consult the Windows NT documentation.

OSD-04207

OSD-04207   **invalid priority specified in CONFIG parameter ORACLE_PRIORITY**

Cause: The priority specified is invalid or out of range.

Action: Specify a valid setting for ORACLE_PRIORITY.

OSD-04208

OSD-04208   **OpenProcess() failure, unable to open process handle**

Cause: There was an unexpected return from the Windows NT system service, OpenProcess().

Action: Check the operating system error code and consult the Windows NT documentation.

OSD-04209

OSD-04209   **Incorrect or unknown background image name given to spdcr()**

Cause: There was an unexpected background name given to spdcr().

Action: Contact Oracle Customer Support.

OSD-04210

OSD-04210   Timeout waiting for thread semaphore

Cause: An Oracle7 thread died holding the semaphore.

Action: Restart Oracle7 instance.

OSD-04211

OSD-04211   **Thread information not found**

Cause: An Oracle7 thread died without deleting its information.

Action: Restart Oracle7 instance.

OSD-04212

OSD-04212   **Maximum number of Oracle threads reached**

Cause: The maximum number of Oracle7 threads for the instance is reached.

Action: Wait until some connections exit before trying again.

OSD-04213

OSD-04213   **Oracle thread unable to DuplicateHandle()**

Cause: This is an internal error, not normally expected to occur.

Action: Contact Oracle Customer Support.

OSD-04214

OSD-04214   **Oracle thread unable to CreateEvent()**

Cause: This is an internal error, not normally expected to occur.

Action: Contact Oracle Customer Support.

OSD-04215

OSD-04215   Bad function code supplied to ssthreadop

Cause: This is an internal error, not normally expected to occur.

Action: Contact Oracle Customer Support.

OSD-04216

OSD-04216   **Unable to find file handle for that thread**

Cause: This is an internal error, not normally expected to occur.

Action: Contact Oracle Customer Support.

OSD-04300

OSD-04300   **unable to read complete record from data file**

Cause: The data file ended in the middle of a record. This error occurs when loading files with a fixed record length.

Action: Verify that the data file is of the correct length and contains complete records.

OSD-04301

OSD-04301   record size too large

Cause: The specified record size is too large to load.

Action: Reduce record size and reload the data.

OSD-04302

OSD-04302   **invalid record type and/or load options**

Cause: The control file's Windows NT file processing options string contains an invalid option or keyword.

Action: Set the Windows NT file processing options string to an acceptable value.

OSD-04400

OSD-04400   **unable to acquire internal semaphore for process**

Cause: Oracle7 Workgroup Server for Windows NT has exceeded the maximum number of connections.

Action: Delete any unused connections and try again.

OSD-04401

OSD-04401 **WaitForSingleObject() failure, unable to obtain semaphore**

Cause: There was an unexpected return from the Windows NT system service, WaitForSingleObject().

Action: Check the operating system error code and consult the Windows NT documentation.

OSD-04500

OSD-04500   **illegal option specified**

Cause: This is an internal error, not normally expected to occur.

Action: Contact Oracle Customer Support.

OSD-04501

OSD-04501   **internal buffer overflow**

Cause: This is an internal error, not normally expected to occur.

Action: Contact Oracle Customer Support.

OSD-04502

OSD-04502   **translations nested too deep**

Cause: The program encountered too many intermediate translations while attempting to translate a configuration variable.

Action: Simplify the values of configuration parameters to include fewer intermediate translations.

OSD-04503

OSD-04503   **text contains no translatable elements**

Cause: The program cannot recognize variables in the text to be translated.

Action: Check and, if necessary, correct the text to be translated.

OSD-04505

OSD-04505   stdin not responding

Cause: The system is unable to receive input from the standard input stream.

Action: Verify that the process has access to an input device.

OSD-04506

OSD-04506   **unable to spawn process via system()**

Cause: The system is out of memory or the executable is invalid.

Action: Shutdown unnecessary processes; install more memory in machine. Verify the name of the executable.

OSD-04507

OSD-04507   **password for 'internal' is incorrect**

Cause: An attempt was made to connect as 'internal' with an invalid password.

Action: Verify that the password is correct and try again.

OSD-04508

OSD-04508   no password given

Cause: An attempt was made to connect as 'internal' without a password.

Action: Enter a valid password when connecting as internal.

OSD-04509

OSD-04509   no password found

Cause: Oracle was unable to locate and retrieve the password for 'internal'.

Action: Verify that Oracle is installed and configured correctly.

OSD-04510

OSD-04510   **operating system roles are not supported**

Cause: An attempt was made to use an operating system role.

Action: Only use roles that were created 'IDENTIFIED BY password' as opposed to 'IDENTIFIED EXTERNALLY'.

OSD-04511

OSD-04511   **unable to get date and time from the operating system**

Cause: There was an unexpected return from GetLocalTime() call.

Action: Verify that the system time is correct on the computer.

OSD-04512

OSD-04512   unable to translate the 'USERNAME' configuration variable on server

Cause: The 'USERNAME' configuration parameter variable on the host is not properly set.

Action: Verify the 'USERNAME' variable is set.

OSD-04513

OSD-04513   **'REMOTE_OS_AUTHENT' variable not set to TRUE**

Cause: For remote operating system logins to function, the 'REMOTE_OS_AUTHENT' parameter must be set to TRUE.

Action: Shut down and start up the instance with 'REMOTE_OS_AUTHENT = TRUE' in the initialization parameter file.

SQL Loader Errors (01300 to 01399)

OSD-01300

OSD-01300   Unable to read complete record from data file

Cause: With fixed length records, the data file ended in the middle of a record

Action: Verify that the data file is of the correct length and contains complete records

OSD-01301

OSD-01301   **Record size too large**

Cause: The record size specified is too large to load

Action: Reduce record size and retry the data load

OSD-01302

OSD-01302   **Invalid record type and/or load options**

Cause: The OS-dependent file processing specifications string contains an invalid option or keyword

Action: Set the OS-dependent string to an acceptable value

OSD-01303

OSD-01303   Unable to open Btrieve file

Cause: The Btrieve file specification was incorrect, or Btrieve is not functional

Action: Verify the Btrieve file name, and verify that Btrieve is operational

OSD-01304

OSD-01304   **Unable to read Btrieve file**

Cause: The Btrieve file was corrupted or truncated

Action: Verify that the Btrieve is valid and complete

OSD-01305

OSD-01305   **Variable length Btrieve files not supported**

Cause: The Btrieve file contains variable length records

Action: Convert the file to fixed length records and reload

Windows-Specific Messages (01000 to 01999)

OSD-01000

OSD-01000   **Logical block size mismatch**

Cause: File created with logical block size different from that requested

Action: Use matching logical block sizes

OSD-01001

OSD-01001   **Invalid logical block size**

Cause: Logical block size is not a multiple of 512 bytes, or is too big

Action: Change requested logical block size

OSD-01002

OSD-01002   Unable to open file

Cause: Invalid path or filename specified, device full, insufficient file handles

Action: Make sure file exists and/or device is not full, verify OS is functioning correctly

OSD-01003

OSD-01003   Unable to read file header block

Cause: Damaged media

Action: Recover file if necessary, verify OS is functioning correctly

OSD-01004

OSD-01004   **Invalid file header**

Cause: Corrupted file

Action: Recover file

OSD-01005

OSD-01005   Lseek() failure, unable to read from file

Cause: Unexpected return from lseek() call

Action: Check OS error code, verify file is not damaged

OSD-01006

OSD-01006   Read() or ReadFile() failure, unable to read from file

Cause: Unexpected return from call to read from file

Action: Check OS error code, verify file is not damaged

OSD-01007

OSD-01007   **Truncated read**

Cause: Unexpected end of file, damaged media

Action: Verify file is not damaged

OSD-01008

OSD-01008   Writefile() failure, unable to write to file

Cause: Unexpected return from Windows system service WriteFile()

Action: Check OS error code and consult Windows documentation

OSD-01009

OSD-01009 Truncated write

Cause: Unable to write to device, device full, damaged media

Action: Verify device is not full, file is not damaged

OSD-01010

OSD-01010   <Create> option specified, file already exists

Cause: Asked to create already existing file

Action: Delete file or use <reuse> option

OSD-01011

OSD-01011   **GetFileSize() failure, unable to obtain file size**

Cause: Unexpected return from GetFileSize()

Action: Check OS error code

OSD-01012

OSD-01012   **File size mismatch**

Cause: File to be reused is too large or too small

Action: Specify new file size

OSD-01013

OSD-01013 **Unable to read line from file**

Cause: Check OS error code

Action: Remedy OS error

OSD-01014

OSD-01014   **Unable to close file**

Cause: Damaged media

Action: Recover file if necessary, verify OS is functioning correctly

OSD-01015

OSD-01015   **Unable to delete file**

Cause: File is write protected

Action: Remove protections from file, verify OS is functioning correctly

OSD-01016

OSD-01016   DosQueryPathInfo() failure, unable to obtain path info

Cause: Unexpected return from DosQueryPathInfo()

Action: Check OS error code

OSD-01017

OSD-01017   **The path specified for archive destination is not a directory**

Cause: A filename was specified as the archive destination

Action: Specify a valid directory as the archive destination

OSD-01018

OSD-01018   Out of space on specified device

Cause: Not enough space on the device to write to or create the file

Action: Free up space for the file or specify another device

OSD-01019

OSD-01019   SetFilePtr() failure, unable to read from file

Cause: Unexpected return from SetFilePtr() call

Action: Check OS error code, verify file is not damaged