# Oracle8™

Getting to Know Oracle8™ and the Oracle8™ Enterprise Edition

Release 8.0

December 1997

Part No.  A58228-01

# ORACLE®

Enabling the Information Age™

*Getting to Know Oracle8 and the Oracle8 Enterprise Edition*

Part No.  A58228-01

Release 8.0

Primary Author:  Paul Lane

Contributors:  Dave Colello, Sandy Dreskin, John Frazzini, Joe Garcia, Michael Hartstein, Lefty Leverenz, Mark Niebur, Tom Portfolio, Den Raphaely

Graphic Designer:  Valarie Moore

# Contents

# 5 Deprecated and Desupported Features

# Send Us Your Comments

***Getting to Know Oracle8 and the Oracle8 Enterprise Edition*, Release 8.0**

**Part No.  A58228-01**

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information?  If so, where?
- Are the examples correct?  Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the chapter, section, and page number (if available). You can send comments to us in the following ways:

- electronic mail - infodev@us.oracle.com
- FAX - (650)506-7228
- postal service:
  Oracle Corporation
  Oracle Server Documentation Manager
  500 Oracle Parkway
  Redwood Shores, CA  94065
  USA

If you would like a reply, please give your name, address, and telephone number below.

# Preface

This manual introduces you to the Oracle data server, which is offered in two product configurations, Oracle**8** and the Oracle**8** Enterprise Edition.

Oracle**8** and the Oracle**8** Enterprise Edition have the same basic features. However, several advanced features are available only with the Enterprise Edition, and some of these are available only as separately purchased options. The major objective of this manual is to explain the differences between the features, functionality, and options available with both product configurations.

## How Getting to Know Oracle8 and the Oracle8 Enterprise Edition Is Organized

This manual is organized as follows:

Chapter 1, "Overview"

Contains a brief introduction to Oracle8 and the Oracle8 Enterprise Edition.

Chapter 2, "New Features and Options"

Contains a more detailed explanation of the new features and options offered by Oracle8 and the Oracle8 Enterprise Edition.

Chapter 3, "Differences between Oracle8 and the Oracle8 Enterprise Edition"

Defines the features and options contained in Oracle8 and the Oracle8 Enterprise Edition.

Chapter 4, "How to Use Oracle8 Documentation"

Describes the organization, location, and content of Oracle8 documentation.

Chapter 5, "Deprecated and Desupported Features"

Lists the deprecated and desupported features in Oracle8 and the Oracle8 Enterprise Edition.

# 1

# Overview

## Introducing Oracle8 and the Oracle8 Enterprise Edition

The Oracle data server has been designed to meet the requirements of the network computing era. To ensure that the Oracle data server is appropriate for both small, departmental applications and enterprise-wide computing, it is offered in two configurations:

- Oracle8
- Oracle8 Enterprise Edition

Both Oracle8 and the Oracle8 Enterprise Edition provide reliable and secure data management for applications ranging from small departmental applications to high-volume on-line transaction systems, or query-intensive data warehouse applications. They also provide the tools for systems management, the flexibility to distribute data efficiently, and the scalability for optimal performance from computing resources.

As information systems evolve toward an open, network-based architecture, they must operate in multi-tier, multi-platform environments. Oracle's open, standards-based Network Computing Architecture (NCA) enables companies to spend less time struggling with administration and more time deploying solutions. At the heart of NCA are Oracle8 and the Oracle8 Enterprise Edition, the next generation of the Oracle data server.

## Oracle8

Oracle8 is intended for smaller implementations and provides cost-effective and easy-to-use functionality. Most importantly, because Oracle and the Oracle8 Enterprise Edition are based on the same code, it is easy to migrate an application from Oracle8 to the Oracle8 Enterprise Edition should you need that product's more powerful feature set and greater scalability. Also, Oracle8 and the Oracle8 Enterprise Edition can coexist in the same distributed environment and transparently communicate.

## Oracle8 Enterprise Edition

With many high-end features, the Oracle8 Enterprise Edition is intended for larger systems. It can support tens of thousands of users, with virtually no limits on the type or amount of data stored. The Oracle8 Enterprise Edition can be deployed for local, line-of-business applications, or its new, high-availability features can be used to support 7x24x52 operations. It also supports applications based on both the relational and the object-relational models.

# 2

# New Features and Options

The following topics are included in this chapter:

- Partitioned Tables and Indexes
- Improved Data Warehouse Performance
- Improvements for OLTP Applications
- Replication
- Object-Relational Technology
- Migration and Interoperability
- Other Enhancements

# Partitioned Tables and Indexes

Very large databases present significant challenges for administrators and application developers. Large amounts of data complicate administrative tasks and affect the availability of the database.

To improve availability, ease administration, and enhance query and DML performance, the Oracle8 Enterprise Edition allows tables and indexes to be partitioned, or broken up, into smaller parts based on a range of key values. Because partitions operate independently of each other, data in a partitioned table are available even if one or more partitions are unavailable. Partitions also make large tables easier to manage by breaking up administrative operations into smaller tasks, which in turn can be performed in parallel. Finally, partitioning a table or index can improve performance of operations on the data by eliminating unneeded partitions from the execution plan of the operation.

Partitioning is transparent to both applications and users, so standard SQL statements in existing applications can run against partitioned tables. The Oracle optimizer is partition-aware, and partitions which do not contain any data required by a query are eliminated from the search, often resulting in a substantial performance increase.

## What is Partitioning?

A table or index can be partitioned or divided into smaller pieces. You define a table or index partitioning strategy when creating the structure. Pick a column or set of columns to act as a partition key, and this key will determine which data is placed into each partition. Data from insert operations is automatically placed into the appropriate partitions, so applications do not need to be rewritten to take advantage of partitioning.

All operations you perform on regular tables can be performed on individual partitions of a partitioned table. For example, you can export or back up single partitions of a table to avoid exporting or backing up the entire table in one operation. You can also perform the ANALYZE command on each partition concurrently to compute statistics needed for cost-based optimization more quickly.

Each partition of a partitioned table or index operates independently. Therefore, operations on one partition are not affected by the availability of other partitions. If one partition becomes unavailable because of a disk crash or administrative operations, both query and DML operations on data in other partitions can still continue.

## Greater Control Over Data

Administrators can specify storage attributes for each partition and the placement of the partition within the host file system, increasing the granularity of control for very large databases. Partitions can be individually taken off-line or brought on-line, backed up, recovered, exported and imported, and loaded—thereby limiting the time required for management operations. An individual index partition can be built for one table partition, reducing the time required for index maintenance operations. Partition operations can be performed in parallel. Partitions increase availability by isolating media and application failures—applications not requiring data from an inaccessible partition continue to run without impact.

## Easy-To-Use Administration

The Oracle8 Enterprise Edition provides a set of easy-to-use management commands for partitions. For example, you can implement a table containing a rolling time window of data with a partitioned table so that only one year's worth of data is ever contained in the table. To accomplish this, just add new partitions to the end of the table and drop partitions containing data more than a year old. You can also split and merge partitions easily to balance data among partitions. Also, stand-alone tables that use Oracle7 Release 7.3 UNION ALL views can be converted into partitions of a single table. This allows users with very large Oracle7 databases to quickly benefit from Oracle8 Enterprise Edition partitioning without having to rebuild large tables.

# Improved Data Warehouse Performance

The Oracle8 Enterprise Edition introduces new features that improve data warehousing performance:

- Enhanced Star-Query Processing

- New Parallel Operations

- Increased Database Size

## Enhanced Star-Query Processing

The Oracle8 Enterprise Edition introduces performance improvements to the processing of star queries, which are common in data warehouse applications. A star query, or star schema, occurs when one or more very large tables, often called fact tables, have relationships to multiple smaller tables called dimension tables. Oracle7 introduced the functionality of star query optimization, which provides performance improvements for these types of queries. In the Oracle8 Enterprise

Edition, however, star-query processing has been improved to provide even better optimization for star queries.

In the Oracle8 Enterprise Edition, a new method for executing star queries has been introduced. Using a more efficient algorithm, and utilizing bitmapped indexes, the new star-query processing provides a significant performance boost to data warehouse applications.

The Oracle8 Enterprise Edition has superior performance with several types of star queries, including star schemas with "sparse" fact tables where the criteria eliminate a great number of the fact table rows. Also, when a schema has multiple fact tables, the optimizer efficiently processes the query. Finally, the Oracle8 Enterprise Edition can efficiently process star queries with large or many dimension tables, unconstrained dimension tables, and dimension tables that have a "snowflake" schema design.

The Oracle8 Enterprise Edition's star-query optimization algorithm, unlike that of Oracle7, does not produce any Cartesian-product joins. Star queries are now processed in two basic phases. First, the Oracle8 Enterprise Edition retrieves exactly the necessary rows from the fact table. This retrieval is done via bit mapped indexes and is very efficient. The second phase joins this result set from the fact table to the relevant dimension tables. This allows for better optimizations of more complex star queries, such as those with multiple fact tables. The new algorithm uses bit-mapped indexes, which offer significant storage savings over previous methods that required concatenated column B-tree indexes. The new algorithm is also completely parallelized, including parallel index scans on both partitioned and non-partitioned tables.

For more information, see *Oracle8 Concepts*.

## New Parallel Operations

Insert, update, and delete operations can now be run in parallel in the Oracle8 Enterprise Edition. These operations, known as parallel DML, are executed in parallel across multiple processes. By having these operations execute in parallel, the statement will be completed much more quickly than if the same statement were executed in a serial fashion. Parallel DML complements parallel query by providing parallel transaction execution as well as queries. Parallel DML is useful in a decision support (DSS) or data warehouse environment where bulk DML operations are common. However, parallel DML operations can also speed up batch jobs running in an OLTP database.

The Oracle8 Enterprise Edition supports parallel inserts, updates, and deletes into partitioned tables. It also supports parallel inserts into non-partitioned tables. The

parallel insert operation on a non-partitioned table is similar to the direct path load operation that is available in Oracle7. It improves performance by formatting and writing disk blocks directly into the datafiles, bypassing the buffer cache and space management bottlenecks. In this case, each parallel insert process inserts data into a segment above the high watermark of the table. After the transaction commits, the high watermark is moved beyond the new segments.

To use parallel DML, it must be enabled prior to execution of the insert, update, or delete operation. Normally, parallel DML operations are done in batch programs or within an application that executes a bulk insert, update, or delete. New hints are available to specify the parallelism of DML statements.

For more information, see *Oracle8 Parallel Server Concepts and Administration.*

### Increased Database Size

The Oracle8 Enterprise Edition can manage databases of hundreds of terabytes in size because of partitioning, administrative improvements, and internal enhancements. Many size limitations in earlier versions of Oracle have been raised, such as the number of columns per table, the maximum database size, and the number of files per database.

# Improvements for OLTP Applications

Demanding OLTP applications benefit from a number of new features which improve scalability, performance, and manageability.

### Extended Backup/Recovery Subsystem

Oracle8 and the Oracle8 Enterprise Edition's server-managed backup and recovery provides a better backup and recovery functionality integrated within the Oracle server. Detailed information is maintained on when backups are performed, exactly which parts of the database are backed up, and where the files are stored. Should a recovery be necessary, Recovery Manager analyzes the state of the database and determines the operations necessary to repair the database. It then automatically performs those operations, greatly simplifying the recovery for the administrator and reducing the possibility of human error. A simple GUI interface within Oracle Enterprise Manager controls backup and recovery. An API is also available for third parties who may wish to provide an alternative interface. Media management layer interfaces to popular third-party tape management products are available. Legato Storage Manager is provided free with both Oracle8 and the Oracle8 Enterprise Edition.

The Oracle8 Enterprise Edition provides multilevel, incremental backups that greatly reduce the size of the backups, because only the changed blocks are backed up. This can substantially reduce the time required to back up a datafile.

Tablespace point-in-time recovery allows one or more tablespaces to be recovered to an earlier time, while the remainder of the database is up and running. This allows many types of user errors to be easily corrected. For example, if a user runs a batch job that incorrectly updates many records in a table, the table can be restored to a time previous to the batch job. Also, if a table is accidentally dropped or truncated, it can be restored to a time before this operation.

For more information, see *Oracle8 Backup and Recovery Guide.*

## Very Large User Populations

Numerous enhancements throughout the Oracle8 and Oracle8 Enterprise Edition servers and Net8 increase the utilization of operating system and networking resources. Connection pooling temporarily drops the physical connection for idle users (and transparently re-establishes the connection when needed), thus increasing the number of users that can be supported. Oracle Connection Manager can be used to configure a middle tier that manages the connections of very large user populations. By configuring multiple connection managers, the Oracle8 Enterprise Edition can support tens of thousands of concurrent users. Shared database links multiplex many users into a database server with a single connection, reducing resource requirements, especially for multitier application architectures.

## Advanced Queuing

The Oracle8 Enterprise Edition advanced queuing feature supports database messaging, or queuing, through a set of queue tables and queue functions.

Advanced queuing adds direct support in the database for deferring transaction execution to a later time and executing transactions in a particular order. This capability allows you to decouple distributed applications and eliminates dependency on external systems for applications requiring high scalability. Enqueue and dequeue operations can be used to shift processing from within a transaction to a background process, thereby improving transaction response time. Also, queuing can be used to implement work flow applications that move data to a system as the state of the data changes. An example of this is moving orders from an order-entry application to a shipping application, and then to a billing system during the life of an order. Advanced queuing can also work in conjunction with popular TP monitor queuing systems.

Messages can be enqueued and dequeued by applications or other queues. The propagation feature enables applications to communicate with each other without having to be connected to the same database or to the same queue. Using the familiar database links and Net8, messages can be propagated from one queue to another, irrespective of whether these are local or remote.

For more information, see *Oracle8 Application Developer's Guide.*

## Parallel Server Improvements

Numerous enhancements in Oracle Parallel Server improve performance, scalability, memory usage, and availability. A common and integrated distributed lock manager replaces the lock managers provided by the different operating system vendors, improving performance and portability on most platforms. Several performance enhancements make the Oracle Parallel Server significantly faster. Also, new global V$ tables improve manageability.

Oracle Parallel Server now uses its own integrated distributed lock manager (DLM) for processing inter-node requests for resources. Previously, Oracle relied on operating system vendors to supply DLM capability for the Parallel Sever to run on a given platform. Integrating the DLM within the Oracle8 Enterprise Edition allows the Oracle Parallel Server to be available on platforms that previously did not support cross-node sharing of resources.

The Oracle8 Enterprise Edition introduces numerous improvements to Oracle Parallel Server performance. System change number (SCN) generation is now optimized for providing SCNs across instances more efficiently. This improvement alone can amount to a 10 to 15 percent improvement in Oracle Parallel Server performance. Also, the DLM now caches locks to avoid cross-node communication of lock information. Pinging, or contention for blocks across nodes, is also reduced by improving the algorithm for controlling access to contended blocks. Reverse-key indexes reduce "hot spots" in indexes, especially primary key indexes, by reversing the bytes of the leaf blocks and thus eliminating the contention for leaf blocks across instances. Partitions also help Parallel Server performance by allowing database administrators to map partitions to specific nodes, which can dramatically reduce pinging. Finally, the Oracle8 Enterprise Edition reduces the memory overhead associated with fine-grain locking in an Oracle Parallel Server.

The Oracle8 Enterprise Edition allows designation of groups of instances for parallel query or parallel DML processing. You can specify to which group an instance belongs and then use the group to process statements from certain applications. This is especially useful for separating OLTP from data warehouse processing among your clustered servers. Using separate instances in an Oracle

Parallel Server ensures that data warehouse queries do not affect the performance of OLTP applications.

The Oracle8 Enterprise Edition introduces global fixed views (GV$) for the Oracle Parallel Server. This allows administrators to log into one instance of a Parallel Server and view global views that contain data from all the V$ views in the cluster. This makes administration operations significantly simpler and improves the productivity of database administrators.

## Transparent Application Failover

Should a node in the Oracle Parallel Server fail, transparent application failover will migrate your connections and automatically re-establishe their sessions on another node. Your applications will continue to run, and you may be unaware of the failure. This provides continuous availability in the event of scheduled and unscheduled outages.

Depending on the amount of overhead you are willing to incur on the client and backup machines, the instance failure could result in a completely transparent migration of user connections to the failover node. You can specify that you want all queries to be cached on the client so that they can be reinstantiated on the failover node. Also, you can have a session pre-connected to the failover node, eliminating the time required to reconnect to a failover instance.

Transparent application failover is useful not only for availability, but also for manual load-balancing or orderly shutdown of the system. If too many users connect to an instance, you can terminate their sessions and have them transparently migrate to another node. You can also shut down a node and have users transparently migrate to a failover node after their current transaction completes. Shutdowns no longer need to interrupt users work.

**Note:** To take advantage of this functionality, applications must be written specifically using new Oracle8 OCI calls.

For more information, see *Oracle8 Parallel Server Concepts and Administration.*

## Improved TP Monitor Support

Dynamic XA support improves performance for multitier applications with industry-standard XA-compliant Transaction Processing (TP) monitors. Oracle8 and the Oracle8 Enterprise Edition improve support of the XA interface with support for dynamic registration and loosely-coupled transaction branches. They also offer better performance and recovery when used with Oracle Parallel Server.

For more information, see *Oracle8 Application Developer's Guide* or *Oracle8 Parallel Server Concepts and Administration*.

## Better Security Administration

Oracle8 and the Oracle8 Enterprise Edition include a Security Server which provides a single sign-on environment for centrally administering users and roles. Oracle Security Server is compliant with the X.509 certificate-based security standard for public/private key authentication. Mutual authentication between client and server is also supported for protection against "rogue" databases designed to capture client communication. Also, a digital signatures tool kit is provided for creating applications designed to identify unauthorized tampering with data.

Oracle8 and the Oracle8 Enterprise Edition also contain improved password maintenance and administration. You can now define a password profile to enforce a security scheme. Passwords can expire after a certain time, or be checked for complexity (e.g., minimum length). You can define your own policy or use the standard stored function for checking the length, content, or reuse of new passwords. Also, you can create user accounts so that the user must immediately change the password upon the first access to the system.

Oracle8 and the Oracle8 Enterprise Edition also add support for privileged database links, which make it unnecessary to embed a password in a database link, and data encryption services callable from OCI and PL/SQL.

## Replication

Oracle8 and Oracle8 Enterprise Edition replication provides improved support for mass-deployment distributed systems, higher throughput failover configurations, and specialized data warehouse systems.

Increased replication performance is now possible. Changes to replica sites can be propagated and applied in parallel, effectively removing the throughput limit on replication, while maintaining transactional integrity. Also, much of the replication functionality has been rewritten and moved from PL/SQL triggers into C code inside the Oracle engine, providing a significant boost in performance. Finally, the amount of data sent over the network for each replicated transaction has been reduced, and network round-trips are kept to a minimum.

Salesforce automation and other mass deployment systems are now easier to administer and design with the subsetting capabilities of both Oracle8 and the Oracle8 Enterprise Edition. Complex subquery snapshots can now include a fast

refresh clause which allows you to build a snapshot of select rows based on a query to another table. For example, you can build a snapshot for each salesman based on the ORDERS table. That snapshot can contain only the orders by selecting the orders based on criteria from an ASSIGNMENTS table. Each salesman sees only his orders and only has to interact with the master table when uploading new orders.

Oracle8 Enterprise Edition replication also provides numerous improvements in manageability and ease of use. Significant enhancements to Oracle Replication Manager, including wizards, make replicated environments easier to set up and maintain. Additional improvements to replication manageability include snapshot registration, which provides valuable information about which sites have associated snapshots; primary key snapshots, which allow faster snapshot refresh after a table reorganization; a new security algorithm, which is easier to administer; and fine-grain quiesce, which allows an administrator to make schema-level changes to one replication group while others continue to operate.

For further information, see *Oracle8 Replication*.

## Object-Relational Technology

The Oracle8 Enterprise Edition makes a major leap in data management technology with the introduction of an object-relational paradigm. Database schemas and applications today are becoming increasingly complex. Often, several separate applications with similar data, such as customer information, billing, and shipping, exist in different database schemas and an MIS department must manage the interoperation. Corporate management of the information becomes a difficult task of integrating different relational objects and different applications, possibly from different vendors, into a more coherent end-user data model. By enhancing the relational database with object extensions, Oracle addresses the need to simplify data modeling and extend the database with new datatypes.

The new, object-relational features include the following:

- Object Types and Views

- Calling External Procedures from within the Database

- Client-Side Support for Objects

- Evolution of Relational Environments

- Development Tools for Object Modeling

- Multimedia Data

- Java

■    Extensibility

## Object Types and Views

Object types provide a way to extend Oracle's relational datatype system. Relational databases support three datatypes: characters, numbers, and dates. Object types allow you to define new datatypes and use them as you would regular relational datatypes. For example, you can create a new type called Address. This object type can have data, called attributes, such as Street, City, and Postal Code. The object type can also have methods or stored procedures, such as Distance, for computing the distance between addresses. These methods can be written in either PL/SQL or C. An address can then be used anywhere a regular datatype could, whether in column definitions, in PL/SQL variables, or even as the definition for an object table.

Oracle's object types can use powerful object modeling techniques for complex objects. For example, you can represent collections of similar objects in array structures or nested tables. You can also store references to objects for fast traversal without joining tables.

Object types allow application developers to code application logic in the database or the middle-tier application server, as opposed to using client-side code. All applications can then share the logic of the new datatypes so developers do not need to rewrite the code. This feature provides the advantages of creating re-usable code components and transparent application partitioning so that the code can reside and execute on the tier that will yield the best performance: client, application server, or database server.

The Oracle8 Enterprise Edition follows the emerging SQL3 standard for object type definition and object modeling techniques. SQL3 defines syntax for creating and modifying object types, generating and storing object identifiers (OIDs), creating references or pointers to objects, and modeling collections of similar objects.

For more information, see *Oracle8 Concepts* and *Oracle8 Application Developer's Guide*.

## Calling External Procedures from within the Database

Oracle8 and the Oracle8 Enterprise Edition provide a safe, fast way for the database to make a call to an external program. The call also can be made through open protocols like HTTP or IIOP (a CORBA standard). External procedures allow you to use existing application code, or write highly-optimized code for specific purposes, such as a computationally complex algorithm like Fast Fourrier Transform (FFT). Also, you can use external procedures to interface with other applications or with specialized devices like embedded systems.

For more information, see *PL/SQL User's Guide and Reference* and *Oracle8 Administrator's Guide.*

## Client-Side Support for Objects

The client-side object cache allows user applications to retrieve a complex hierarchy of objects into an application cache. The application can then traverse the objects without performing additional network retrievals. This provides a convenient and fast way to use objects in a client application and write code that is more like the native object-oriented code.

A new utility, the Object Type Translator, has been introduced as a quick way of generating header and implementation files for applications running against object schemas.

## Evolution of Relational Environments

The Oracle8 Enterprise Edition is designed to allow users to easily evolve into using the new object-oriented functionality, as all existing applications are upwardly compatible. The new object-relational extensions are built on the same foundation as the relational functionality, which means that users do not have to discard or rewrite their existing relational applications before migrating to the Oracle8 Enterprise Edition. Unlike other object-relational databases, this design allows the older relational applications, which still read and write rows and columns, to coexist with new object-oriented applications, which read and write objects. The Oracle8 Enterprise Edition provides object views to retrieve relational data and represent the data to a client as if it were an object and vice-versa.

For example, an existing relational order-entry system might need a new front-end for the World Wide Web. The existing applications accessing the relational schema can remain in operation, and a new set of object views can be developed as an object representation for the client. New and old applications can be based on the same data, but each has its own representation.

For more information, see *Oracle8 Concepts.*

## Development Tools for Object Modeling

The Object Database Designer offers object support in a design tool. It helps you design, create, and access object-based Oracle8 Enterprise Edition systems.

For more information, see *Object Database Designer.*

## Multimedia Data

Large Objects (LOBs) handle unstructured data such as images, sounds, video, and text, and have much richer functionality than their predecessors, LONG and LONG RAW. Character LOBs (CLOB or NCLOB), Binary LOBs, and BFILES (externally stored LOBs), can be replicated and can be an attribute of an object. You can also have more than one LOB per table. LOBs also have a greater maximum size than LONGs and have different mechanisms for maintaining read-consistency and random access.

LOB data is indexed for fast access starting at a specified byte. For example, you can read/write at specific byte-offsets. You can also read/write LOBs through the Oracle8 buffer cache or access them directly from disk.

LOB functionality is available with both Oracle8 and the Oracle8 Enterprise Edition and does not require the Objects option to be licensed and installed.

For more information, see *Oracle8 Concepts*.

## Java

Oracle has a long history of supporting cross-platform environments. Oracle8 and the Oracle8 Enterprise Edition build upon that with strong support for Java.

Oracle currently has two methods for accessing Oracle data from Java programs: an Oracle-provided JDBC driver integrated with Oracle's object types, and JSQL for embedding SQL statements into Java code. Oracle provides its own JDBC drivers for better performance. JSQL allows you to include SQL statements in a Java application. The JSQL precompiler then converts the SQL into JDBC calls, which allows you to use existing SQL code in new Java applications.

For more information, see *Oracle8 JDBC Drivers*.

## Extensibility

Extensibility allows you to define your own datatypes. You can extend the capabilities of your current database by creating new datatypes for your specific applications. These new datatypes can be used in the same manner as the ones you have now; the same operations can be performed. Image, Spatial, Time Series, Visual, and Context cartridges are now available as extensions to the data server.

# Migration and Interoperability

A simple and fast migration utility rebuilds the data dictionary and converts the control files, log files, and data blocks. The migration utility converts any Oracle 7.1, 7.2, or 7.3 database into an Oracle8 or Oracle8 Enterprise Edition database. Oracle7 applications run unchanged against either of the Oracle8 products. Distributed commands from either of the Oracle8 products run against Oracle7, and vice-versa.

For more information, see *Oracle8 Migration*.

# Other Enhancements

## Index-Organized Tables

Index-organized tables store the data columns of a table within the leaf nodes of the Oracle8 B-tree index structure. This reduces overall storage requirements when most columns are indexed by storing the columns only once, rather than in both an index and a separate table. Index-organized tables also reduce access time by retrieving all columns from one location instead of two.

For more information, see *Oracle8 Concepts*.

## Reverse Key Indexes

Reverse key indexes reduce the "hot spots" in indexes, especially ascending indexes. Unbalanced indexes can cause the index to become increasingly deep as the base table grows. Reverse key indexes reverse the bytes of leaf-block entries, therefore preventing "sliding indexes".

For more information, see *Oracle8 Concepts*.

## Improved Constraint Processing

Constraint processing has been dramatically improved. You can now use non-unique indexes to enforce UNIQUE and PRIMARY KEY constraints. This eliminates redundant indexes and permits indexes to remain valid while constraints are disabled. Deferred-constraint checking shifts integrity-constraint checking from the end of statement execution to when a commit is issued. This simplifies the coding of certain operations involving integrity constraints. Also, all constraints can be enabled concurrently and in parallel with other constraints, and permit concurrent DML while the enabling continues.

## Two Character Sets in One Database

In both Oracle8 and the Oracle8 Enterprise Edition, a new NCHAR datatype allows a second character set in one database. This improves performance and storage predictability for some Asian language, multibyte character set databases. Also, fixed-length native Unicode 2.0 and Chinese GBK support is new.

## Miscellaneous Improvements

A non-updatable view (i.e., a joined view) can be updatable through the use of a new type of trigger, the INSTEAD OF trigger. This trigger allows you to replace INSERT, UPDATE, and DELETE operations on views with your own logic so that even views based on complex joins can be modified. Also, you can place subqueries in the select list of other queries to return result sets from an arbitrary list of detail tables without performing join operations. By placing a cursor expression in a select list, you can fetch rows from detail tables in a 3GL program without specifying complex join conditions in the FROM clause.

# 3

# Differences between Oracle8 and the Oracle8 Enterprise Edition

The following topics are included in this chapter:

- Feature Factoring
- The V$OPTION Table
- Factoring-Related Error Message

# Feature Factoring

Feature factoring is how Oracle controls which features and options are included with each product configuration. Some features of the Oracle8 Enterprise Edition are available only as options.

If you are upgrading to Oracle8, instead of the Oracle8 Enterprise Edition, you might lose certain options and features that were available to you in the previous release. For further information regarding migration, see *Oracle8 Migration*.

The following tables explain the features and options in Oracle8 and the Oracle8 Enterprise Edition.

*Table 3–1   Availability of Oracle8 Features*

| Feature | Oracle8 | Oracle8 Enterprise Edition | Notes and References |
|---|---|---|---|
| *Systems Management and Availability* | | | |
| Enterprise Manager | Y | Y | |
| Enterprise Manager Performance Pack | N | Y | |
| Server Managed Backup and Recovery | Y | Y | |
| Recovery catalog for online backup | Y | Y | |
| Online recovery | Y | Y | |
| Incremental backup and recovery | N | Y | |
| Parallel backup and recovery | N | Y | Oracle8 can only allocate one recovery manager channel at a time, thus limiting the parallelism to one stream. The Oracle8 Enterprise Edition allows unlimited parallelism. |
| Legato Storage Manager | Y | Y | |
| Point-in-time tablespace recovery | N | Y | |
| Fail Safe for Oracle8 on NT | Y | Y | This feature is available on Windows NT. |

*Table 3–1   Availability of Oracle8 Features (Cont.)*

| Feature | Oracle8 | Oracle8 Enterprise Edition | Notes and References |
|---|---|---|---|
| ***Datawarehouse/VLDB Features*** | | | |
| Bit-mapped indexes | N | Y | |
| Star Query optimization | Y | Y | In Oracle8, star query processing is the algorithm that utilizes B-tree indexes.  In the Oracle8 Enterprise Edition, the parallel bitmap index join algorithm can also be utilized. |
| Parallel Execution | N | Y | |
| Parallel Load | N | Y | |
| Parallel Query | N | Y | |
| Parallel DML (insert, update, and delete) | N | Y | This feature requires the Partitioning Option. |
| Parallel index scans | N | Y | This feature requires the Partitioning Option. |
| Parallel Bitmap Star Joins | N | Y | |
| Parallel Index Build | N | Y | |
| Parallel Analyze | N | Y | This feature requires the Partitioning Option. |
| ***Programming Interfaces*** | | | |
| Oracle Call Interface | Y | Y | |
| Objects for OLE | Y | Y | |
| ODBC driver | Y | Y | |
| Pro*C/C++ | Y | Y | This feature is available at extra cost as part of Oracle Programmer. |
| ***Object Features*** | | | |
| object references (REFs) | Y | Y | |
| object collections | Y | Y | |
| nested tables | Y | Y | |
| variable arrays (varrays) | Y | Y | |
| object views | Y | Y | |

*Table 3–1   Availability of Oracle8 Features (Cont.)*

| Feature | Oracle8 | Oracle8 Enterprise Edition | Notes and References |
|---|---|---|---|
| ***Distributed Features*** | | | |
| Distributed queries | Y | Y | |
| Distributed transactions | Y | Y | |
|    Two-phase commit | Y | Y | |
|    XA | Y | Y | |
| Heterogeneous Services | Y | Y | |
| Basic Replication | Y | Y | |
|    Read-only snapshots | Y | Y | |
|    Subquery subsetting | Y | Y | |
|    Primary key based snapshots | Y | Y | |
|    Internal triggers | Y | Y | |
|    Replicated LOBs | Y | Y | |
| Advanced Replication | N | Y | |
|    Updatable snapshots | N | Y | |
|    Multimaster replication | N | Y | |
|    Conflict detection and resolution | N | Y | |
|    Replication Manager | N | Y | |
|    Parallel propagation | N | Y | |
|    Minimized communication | N | Y | |
| ***Networking Features*** | | | |
| Oracle Names | Y | Y | |
| Oracle Connection Manager | N | Y | |
| Connection Pooling | Y | Y | |
| Connection Multiplexing | Y | Y | |

*Table 3–1   Availability of Oracle8 Features (Cont.)*

| Feature | Oracle8 | Oracle8 Enterprise Edition | Notes and References |
|---|---|---|---|
| MultiProtocol Connectivity | N | Y | |
| Oracle Security Server | Y | Y | |
| *Other Features* | | | |
| Advanced Queuing | N | Y | Advanced Queuing supports two types of message payloads-RAW and structured (where the structure is specified by an object type). Without the Objects option, users can only use RAW queues.  Both single-consumer and multi-consumer RAW queues are available without the objects option. |
| | | | See *Oracle8 Application Developer's Guide* for details. |
| Reverse key indexes | Y | Y | |
| Password management | Y | Y | |
| Index-organized tables | Y | Y | |
| PL/SQL stored procedures, triggers | Y | Y | |
| INSTEAD OF triggers | Y | Y | This feature can be used with relational views as well as object views.  Object capabilities require the Objects option. |
| External procedures | Y | Y | |
| National Language Support | Y | Y | |
| LOB support | Y | Y | |

*Table 3–1   Availability of Oracle8 Features (Cont.)*

| Feature | Oracle8 | Oracle8 Enterprise Edition | Notes and References |
|---|---|---|---|
| *Data Cartridges* (extended datatype support; available at extra cost) | | | |
| ConText Cartridge | Y | Y | The ConText Cartridge offers full text retrieval. |
| Video Cartridge | Y | Y | |
| Image Cartridge | N | Y | For more information, see *Oracle8 Image Cartridge User's Guide.* |
| Visual Information Retrieval Cartridge | N | Y | For more information, see *Oracle8 Visual Information Retrieval Cartridge User's Guide.* |
| Time Series Cartridge | N | Y | **Note:** To add functions or types to the Time Series Cartridge requires the Objects option.<br><br>For more information, see *Oracle8 Time Series Cartridge User's Guide.* |
| Spatial Data Cartridge | N | Y | For more information, see *Oracle8 Spatial Cartridge User's Guide and Reference.* |

*Table 3–2  Availability of Oracle8 Options*

| Feature | Oracle8 | Oracle8 Enterprise Edition | Notes and References |
|---|---|---|---|
| **Oracle8 Options** (available only for the Oracle8 Enterprise Edition, at extra cost) | | | |
| Objects Option | N | Y | The Objects option allows data to be represented, accessed, manipulated, and stored as business objects. |
| | | | It allows the definition of object types and all features that use object types, including object tables, object views, the object cache, extended Oracle Call Interface features, the object features of Pro*C, extended SQL and PL/SQL support of object capabilities. |
| | | | No CREATE TYPE, ALTER TYPE, or CREATE OR REPLACE TYPE statements are possible without this option. |
| | | | Note:  INSTEAD OF triggers are available with the Oracle8 Enterprise Edition, even without the objects option, and can apply to relational views.  Also note that LOBs (large objects) are included in both Oracle8 and the Oracle8 Enterprise Edition. |
| Partitioning Option | N | Y | The Partitioning option allows the definition of partitions of tables and indexes.  This feature is useful for OLTP and Data Warehousing users.  Note that this option is required for parallel index scans and parallel DML (insert, update, and delete).  Note also that partition views, a feature of Oracle7 release 7.3, is included in the Oracle8 Enterprise Edition without this option. |
| | | | No CREATE PARTITION statements are possible without this option. |

*Table 3–2   Availability of Oracle8 Options (Cont.)*

| Feature | Oracle8 | Oracle8 Enterprise Edition | Notes and References |
| --- | --- | --- | --- |
| Advanced Networking Option | N | Y | The Advanced Networking option provides client/server, server/server network security using encryption and data integrity checking as well as enhanced user authentication services. |
| Enterprise Manager Performance Pack | N | Y | The Oracle Enterprise Manager Performance Pack provides an advanced set of tools for managing database environments.  It consists of six applications for advanced diagnostics, monitoring, and tuning of Oracle environments. |
| Parallel Server Option | N | Y | The Parallel Server option allows multiple nodes of a loosely-coupled system (such as a cluster or massively parallel platform) to share access to a single database, for increased scalability and availability.  It includes client-side failover and, on some platforms, inter-node parallel query. |

# The V$OPTION Table

To check the values for different options on your database, query the V$OPTION table:

```
SVRMGR> SELECT * FROM V$OPTION;
```

The response will, of course, vary depending upon the specific installation. A sample result might be:

| PARAMETER | VALUE | Explanation |
|---|---|---|
| Partitioning | FALSE | These options can be purchased |
| Objects | FALSE | individually or together, but |
| Parallel Server | FALSE | only with the Enterprise Edition |
| Advanced replication | TRUE | |
| Bit-mapped indexes | TRUE | |
| Connection pooling | TRUE | All 11 TRUE |
| Connection multiplexing | TRUE | for the Oracle8 |
| Database queuing | TRUE | Enterprise Edition; |
| Incremental backup and recovery | TRUE | |
| Instead-of triggers | TRUE | All 11 FALSE for |
| Parallel backup and recover | TRUE | Oracle8 |
| Parallel execution  (see note below) | TRUE | |
| Parallel load | TRUE | |
| Point-in-time tablespace recovery | TRUE | |

The above example is an Oracle8 Enterprise Edition installation with no extra options.

**Note:** Parallel execution means SQL run in parallel. This includes parallel query, parallel DML, and parallel DDL.

The following example shows an Oracle8 Enterprise Edition installation with Oracle Parallel Server and Partitioning options, but without the Objects option.

| PARAMETER | VALUE |
|---|---|
| Partitioning | TRUE |
| Objects | FALSE |
| Parallel Server | TRUE |
| Advanced replication | TRUE |
| Bit-mapped indexes | TRUE |
| Connection pooling | TRUE |
| Connection multiplexing | TRUE |
| Database queuing | TRUE |
| Incremental backup and recovery | TRUE |
| Instead-of triggers | TRUE |
| Parallel backup and recover | TRUE |
| Parallel execution | TRUE |
| Parallel load | TRUE |
| Point-in-time tablespace recovery | TRUE |

## Factoring-Related Error Message

One new error message deals with product factoring.  It is:

ORA-00439:  feature not enabled:  %s

where %s is the name of the feature as it exists in the V$OPTION view in the database.  For example:

ORA-00439:  feature not enabled: Objects

would occur if you tried to use the CREATE TYPE  ... AS OBJECT syntax without having purchased the Objects option.

# 4

## How to Use Oracle8 Documentation

## Documentation Structure

Oracle8 and the Oracle8 Enterprise Edition are supported by the same documentation set.  There are references throughout the documentation set regarding features and options which are available only with the Oracle8 Enterprise Edition.

This documentation is available in both online and printed book format.

### Online Documentation

Figure 4–1, "Online Documentation Structure" shows the organization of the online Oracle8 and Oracle8 Enterprise Edition documentation.

**Figure 4–1   Online Documentation Structure**
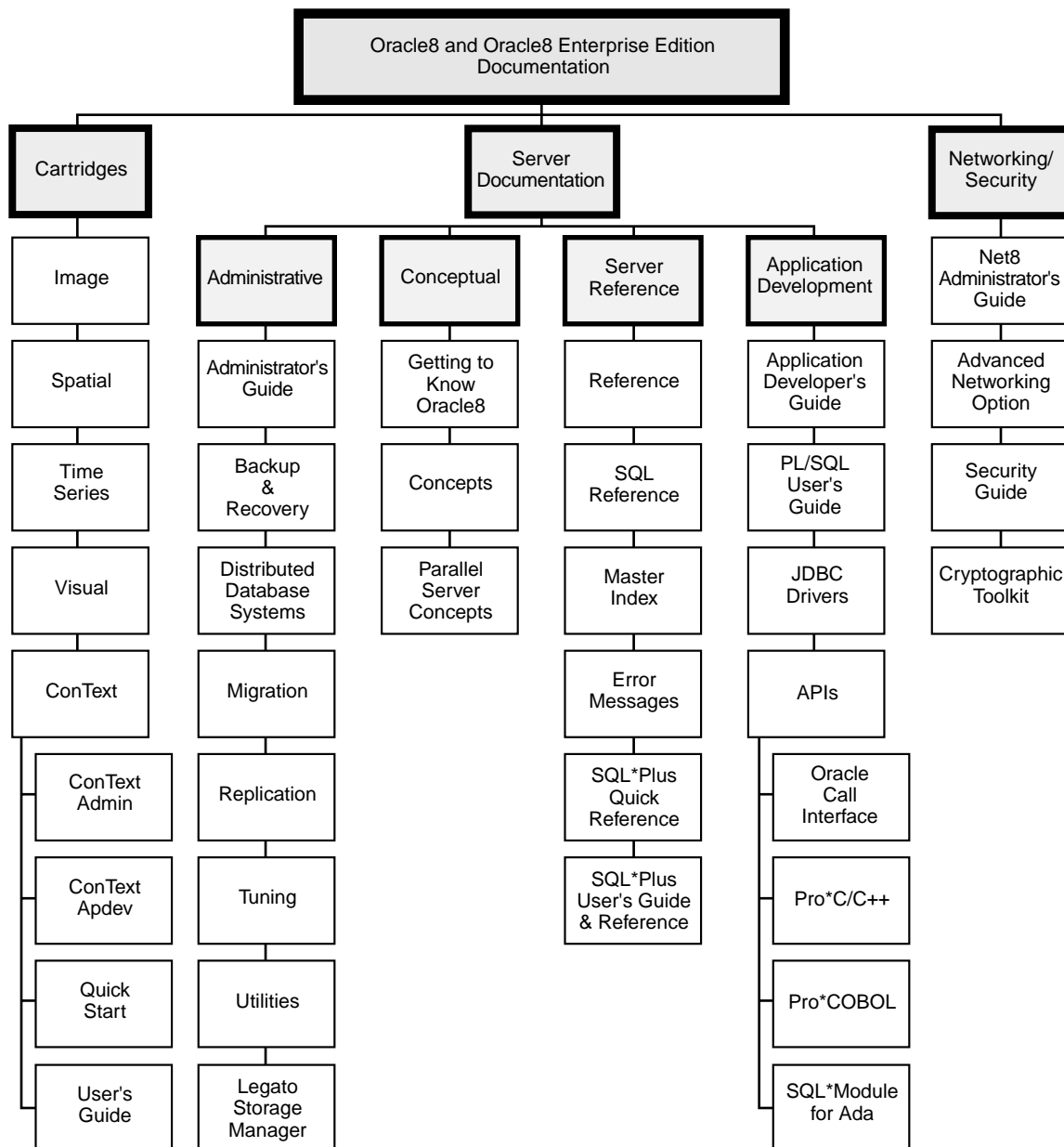
# Printed Documentation

The printed Oracle8 and Oracle8 Enterprise Edition documentation set is divided into three documentation sets:

- Server Documentation Set
- Cartridges Documentation Set
- Networking and Security Documentation Set

### Server Documentation Set

The following list of books and part numbers describes the Server printed documentation set.  The part number is A58405-01.

| | |
|---|---|
| Oracle8 Generic Documentation Master Index | A58495-01 |
| Oracle8 Concepts, Release 8.0 | A58424-01 |
| Oracle8 Administrator's Guide, Release 8.0 | A58397-01 |
| Oracle8 Backup and Recovery, Release 8.0 | A58396-01 |
| Oracle8 Tuning, Release 8.0 | A58246-01 |
| Oracle8 Distributed Database Systems, Release 8.0 | A58247-01 |
| Oracle8 Replication, Release 8.0 | A58245-01 |
| Oracle8 Utilities, Release 8.0 | A58244-01 |
| Oracle8 Migration, Release 8.0 | A58243-01 |
| Oracle8 Reference, Release 8.0 | A58242-01 |
| Oracle8 Application Developer's Guide, Release 8.0 | A58241-01 |
| Oracle8 SQL Reference, Release 8.0 | A58425-01 |
| Oracle8 Parallel Server Concepts and Administration, Release 8.0 | A58238-01 |
| Oracle8 JDBC Drivers, Release 8.0 | A58237-01 |
| Oracle8 Error Messages | A58312-01 |
| PL/SQL User's Guide and Reference, Release 8.0 | A58236-01 |
| Oracle Call Interface Programmer's Guide, Release 8.0 | A58427-01 |
| Pro*C/C++ Precompiler Programmer's Guide, Release 8.0 | A58233-01 |

| | |
|---|---|
| Pro*COBOL Precompiler Programmer's Guide, Release 8.0 | A58232-01 |
| SQL*Module for Ada Programmer's Guide, Release 8.0 | A58231-01 |
| Getting to Know Oracle8 and the Oracle8 Enterprise Edition, Release 8.0 | A58228-01 |
| Legato Storage Manager Administrator's Guide | A58373-01 |
| SQL*Plus Quick Reference | A53718-01 |
| SQL*Plus User's Guide and Reference | A53717-01 |

**Cartridges Documentation Set**

The following list of books and part numbers describes the Cartridges printed documentation set. The part number is A58422-01.

| | |
|---|---|
| Oracle8 Image Cartridge User's Guide, Release 8.0.4 | A55713-02 |
| Oracle8 Spatial Cartridge User's Guide and Reference, Release 8.0.4 | A53264-02 |
| Oracle8 Time Series Cartridge User's Guide, Release 8.0.4 | A57501-01 |
| Oracle8 Visual Information Retrieval Cartridge User's Guide, Release 1.0.1 | A55255-02 |
| Oracle8 ConText Cartridge Administrator's Guide | A58165-01 |
| Oracle8 ConText Cartridge Application Developer's Guide | A58164-01 |
| Oracle8 ConText Cartridge Workbench User's Guide | A57700-01 |
| Oracle8 ConText Cartridge QuickStart | A58161-01 |

**Networking and Security Documentation Set**

The following list of books and part numbers describes the Networking and Security printed documentation set. The part number is A58423-01.

| | |
|---|---|
| Oracle Net8 Administrator's Guide, Release 8.0 | A58230-01 |
| Oracle Advanced Networking Option Administrator's Guide, Release 8.0 | A58229-01 |
| Oracle Cryptographic Toolkit Programmer's Guide | A54082-02 |
| Oracle Security Server Guide | A54088-01 |

# 5

# Deprecated and Desupported Features

## Deprecated and Desupported Features

Many features are no longer advisable for use with Oracle8.

- Strings of Zero Length are not equivalent to a NULL
- The SELECT Privilege
- Date Format Strings are Stricter
- SERIALIZABLE=TRUE is no longer supported
- Non-Deferred Linking
- Single-Task Linking
- CONNECT INTERNAL
- Partition Views
- V6 Compatibility Behavior

## Strings of Zero Length are not equivalent to a NULL

A string of zero length ('') is not equivalent to a NULL.

According to the ANSI SQL 1992 Transitional standard, a zero-length or empty string is not the same as NULL. The Oracle server may comply fully with this aspect of the standard in the future. Therefore, it is recommended that applications ensure that empty strings values and NULL are not treated equivalently.

## The SELECT Privilege

The SELECT privilege will be required on tables that users update.

Always grant the SELECT privilege to a user or role if you grant the UPDATE or DELETE privileges on the table. See also *Oracle8 Reference* for more information about the initialization parameter, SQL92_SECURITY.

## Date Format Strings are Stricter

In Oracle7, a space or punctuation character in the format string caused the corresponding character in the date string to be discarded. This caused incorrect dates to be entered into the database since alphanumeric characters were thrown out. In Oracle8, an error occurs if an alphanumeric character is found in the date string when a punctuation character or space is found in the format string.

Example:

TO_CHAR(TO_DATE('0297','MM/YY'), 'MM/YY')

Oracle7 result: 02/07

Oracle8 result: ORA-1861

## SERIALIZABLE=TRUE is no longer supported

The initialization parameter SERIALIZABLE=TRUE is no longer supported in Oracle8 and beyond. The default behavior henceforth is as if SERIALIZABLE was set to FALSE. Use the SET TRANSACTION ISOLATION LEVEL SERIALIZABLE command to achieve similar transaction isolation behavior.

## Non-Deferred Linking

Application developers are cautioned that Oracle plans to desupport non-deferred mode linking beginning with the release of Oracle9 (it will continue to be supported with all the releases of Oracle8). Recognizing these plans, application

developers should no longer use non-deferred mode linking in developing new applications. Currently Oracle supports two linking modes:

1. Non-deferred linking: The Version 6 OCI (client) only supported non-deferred linking which meant that for each SQL statement, a parse, a bind and a define call were each executed separately with individual round trips between the client and the server. This significantly increased network traffic between the client and the server and reduced both the performance and scalability of OCI applications.

2. Deferred linking: Unlike the Version 6 OCI, the Version 7 OCI supports both non-deferred linking and deferred linking. Deferred mode linking essentially defers the bind and define steps until the statement executes, so it automatically bundles and defers the bind and define calls to execution time. Further, when the application is linked with deferred mode and a special parsing call is used (the OPARSE call with the DEFFLG set to a non-zero value), even the parse call can be deferred to execution time. Note that deferred mode linking does not depend on the specific OCI calls that the application uses, only on the link option that is selected.

Deferred mode linking therefore significantly reduces the number of round trips between the client and the server and as a result improves the performance and scalability of OCI applications. The default behavior of Version 7 OCI connected to the Oracle7 server is deferred mode linking. However, Version 7 OCI also supports non-deferred linking by setting specific link time options.

Further, Version 8 OCI has two types of calls: first, all the Version 7 OCI calls are supported with Version 8 OCI, i.e., they will work with a Version 8 OCI client by relinking the version 8 OCI libraries. Second, there are additional Version 8-specific OCI calls. The default mode with the first type of calls continues to be deferred mode linking; however, non-deferred mode linking is supported for these calls through all releases of Oracle8 by setting link time options. However, Version 8-specific calls use a different paradigm and as a result non-deferred mode linking is not necessary.

The various combinations of client side libraries and server with which non-deferred linking is currently supported are summarized below:

*Table 5–1   Non-deferred Linking:  Server and Client-side Libraries*

| Server | Client-side Libraries | | | | |
|---|---|---|---|---|---|
| | OCIv6 | OCIv7 | OCIv8(v7 calls) | OCIv8(v8 calls) | OCIv9 |
| Oracle9 | Not supported | Default: deferred Non-deferred supported | Default: deferred Non-deferred supported | Not supported | Not supported |
| Oracle8 | Not supported | Default: deferred Non-deferred supported | Default: deferred Non-deferred supported | Not supported | Not supported |
| Oracle7 | Non-deferred mode only | Default: deferred Non-deferred supported | Default: deferred Non-deferred supported | Not supported | Not supported |
| Oracle6 | Non-deferred mode only | Default: deferred Non-deferred supported | Default: deferred Non-deferred supported | Not supported | Not supported |

Oracle will continue to support deferred-mode linking with all the releases of Oracle8 (all 8.* releases). This has the following implications:

### Applications Using Version 6 OCI Libraries
Since the Version 6 OCI library is not supported against the Oracle8 database, applications using the Version 6 library cannot be run against an Oracle8 database.

### Applications Using Version 7 OCI Libraries
Applications using Version 7 OCI libraries can run in two configurations against an Oracle8 database:

- 1. They can be run with Version 7 OCI libraries against an Oracle8 database in non-deferred mode provided link time options are set appropriately.

- 2. They can also be relinked with the Version 8 OCI libraries and run in non-deferred mode provided link time options are set appropriately.

Oracle will support the first configuration through all the releases of Oracle8. However, the second configuration will not be supported in Oracle9. Therefore,

applications that require non-deferred linking will not be able to upgrade to Oracle9 client-side libraries.

### Applications using Version 8 OCI Libraries

Applications using Version 8 specific OCI calls, such as those used to access Oracle8's object types, do not need to use non-deferred mode linking since the Version 8 OCI uses a different paradigm. Applications using only Version 7 OCI calls will be able to use non-deferred mode linking but only through Oracle Release 8.1.

## Single-Task Linking

Single-task linking is a feature used by a limited number of Oracle's customers primarily on the OpenVMS platform. Application developers are cautioned that Oracle will continue to support single-task linking with all the releases of Oracle8 (all 8.* releases) but will desupport it beginning with the first release after Oracle8.

With single-task linking, Oracle supports two configurations to link Oracle products and user-written applications against the Oracle database:

- 1. Single-task linking: In this case, applications are directly linked against the Oracle shareable image making single-task connection to Oracle.

- 2. Two-task linking: In this case, applications linked in a standalone configuration can only connect to Oracle using SQL*Net's two task drivers such as SQL*Net DECnet or SQL*Net VMS Mailbox on the OpenVMS platform. This is the typical configuration used in the large majority of client-server applications. With two task linking applications and tools connect with the Oracle7 database through a programmatic interface that creates a shadow process for each user process. This shadow process runs a copy of the Oracle shareable image on behalf of the user process using SQL*Net protocols to communicate between the user and shadow processes. Therefore, with this interface, user routines that invoke the Oracle7 server functions run as one process or task, and the Oracle7 routines that execute these functions run as the second task.

Oracle will continue to support single-task linking with all the releases of Oracle8 (all 8.* releases) but will desupport it beginning with the first release after Oracle8. Application developers who would like to use single-task linking to run their applications will not be able to do so against the first server release after Oracle8.

## CONNECT INTERNAL

CONNECT INTERNAL is currently supported for backwards compatibility only. CONNECT INTERNAL will be completely desupported in the first maintenance release after Oracle8.1, so that CONNECT INTERNAL may no longer be used to connect to Oracle. If you have not done so already, you should plan to migrate your applications to use other connection syntax. See *Oracle8 Administrator's Guide* for further details.

## Partition Views

In Oracle8, partitioned tables are strongly recommended in preference to partition views. If you must use partition views, please see "Rules and Guidelines for Use" in the Release 7.3.3 Tuning Guide on page 11-2.

## V6 Compatibility Behavior

With Oracle7, Oracle offered a Version 6 [V6] Compatibility flag that allowed application developers developing Oracle7 applications to emulate Oracle6 behavior. With the release of Oracle8, users are cautioned that the Version 6 compatibility flag is being desupported effective immediately in all of the Oracle8 products including PL/SQL8, all the Oracle Precompilers, the Oracle8 Oracle Call Interface, SQL*Module, and SQL*Plus. The desupport of the V6 compatibility flag is consistent with Oracle's policy of supporting backwards compatibility and behavior from one version release upgrade to another, i.e., from Oracle6 to Oracle7 but not for more than one version release upgrade.

Users who do not absolutely need to maintain V6 behavior are encouraged to upgrade their Oracle7 clients to Oracle8. Users who absolutely need to continue to emulate V6 behavior for certain applications need to maintain one $ORACLE_HOME with an Oracle7 client for those applications. They can create a separate $ORACLE_HOME with an Oracle8 client for those applications for which they do not need V6 behavior.

Specifically, the V6 Compatibility flag emulated the following aspects of OracleV6 behavior with Oracle7:

- String literals are fixed length in Oracle7 but are treated as variable  length with the V6 flag.

- PL/SQL Local char variables are fixed length in Oracle7 but are treated as variable length with the V6 flag.

- Return value of SQL Functions (e.g., USER) are fixed length characters in Oracle7 but are treated as variable length characters with the V6 flag.

- Select/Fetch of a NULL with no indicator raises an ORA-1405 error with Oracle7 but returns no error with the V6 flag.

- SQL group function is called at FETCH time with Oracle7 but is called at query execution time with the V6 flag.

- Describe of a fixed length string returns Type=96 with Oracle7 but returns Type=1 with the V6 flag.

All of the above are being desupported with the desupport of the V6 Compatibility Flag with Oracle8.

# Index

## A

## C

## D

## E

## F

## I

## J

## L