



# Oracle8™ Enterprise Edition Release 8.0

*New Features Overview*

*June 1997*

## THE DATABASE FOR NETWORK COMPUTING

As enterprises evolve their information systems to open, network-based architectures for distributed computing, they face the following challenges:

- Maintaining the robustness and performance of existing systems in new systems under development
- Managing the interoperability of existing data and new data
- Identifying the best products to deploy within that environment
- Integrating and managing a multitier, multiplatform environment

Oracle's open, standards-based Network Computing Architecture™ (NCA) allows information technology organizations to spend less time struggling with interoperability and more time focusing on deploying new systems. The Oracle universal data server, a major component of NCA, is designed to meet the performance, reliability, and scalability demands of network-centric computing and object development methods. Whether for traditional enterprise applications or electronic commerce on the Web, Oracle8 and NCA provide the power, performance, robustness, network integration, and flexibility to support the most demanding applications.

Oracle8 includes across-the-board improvements in resource utilization and performance. Whether you need to support tens of thousands of users and hundreds of terabytes of data, or you have a much smaller but no less critical system, everyone benefits from the superior performance of Oracle8. Oracle8 "raises the ceiling" and supports dramatically larger and more demanding on-line transaction processing (OLTP) and data warehousing applications.

Oracle8 sets new standards for availability with features to support nearly 7x24x52 operations. As the first object-relational Oracle server, Oracle8 introduces an object-oriented paradigm providing new capabilities for managing data complexity. Release 8.0 also includes enhancements in nearly all other areas of the Oracle Server, improving overall availability, performance, manageability, multimedia datatype support, and replication functionality.

Oracle8 Enterprise Edition Release 8.0 is designed to address the following industry needs:

- High availability and manageability with partitioned tables and indexes
- Enhanced parallelism
- Improved performance and manageability of data warehouse applications
- Mainframe-caliber OLTP
- Better security administration
- Improved distributed computing
- Object technology and extensibility
- Easy-to-use management tools
- Seamless migration and interoperability with previous releases

## **PARTITIONED TABLES AND INDEXES**

Very large databases present significant challenges for administrators and application designers. Large amounts of data complicate administrative tasks and affect the availability of the database. Most administrative tasks must be performed in a small time window so that they minimally affect the performance of on-line operations. As tables grow larger, administrative operations such as importing and exporting data, back-up and recovery, and defragmenting tables, must be manually broken up into manageable pieces to fit into the required time windows. Also, if one disk containing a portion of a large table becomes unavailable, the entire table is unavailable, and all processing involving that table must halt.

Oracle8 allows tables and indexes to be partitioned or broken up into smaller parts based on a range of key values, this eases administration, improves the availability of mission-critical data, and enhances query and data manipulation language (DML) performance. Partitioning is a “divide and conquer” strategy that improves administration and performance in data warehouse and OLTP systems. Because partitions operate independently of each other, data in a partitioned table is available even if one or more partitions is unavailable. Partitions also make large tables easier to manage by breaking up the administrative tasks into smaller amounts, which in turn can be performed in parallel. Finally, partitioning a table or index can improve performance of operations on the data by eliminating unused partitions from the execution plan of the operation.

Partitioning is transparent to applications and standard DML statements that run against partitioned tables. The Oracle8 optimizer is partition-aware, and partitions that 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 several smaller pieces. You define a table or index partitioning strategy when you create the structure. You pick a column or set of columns to act as a partition key. This key determines which data is placed into each partition. Oracle8 automatically places data from insert operations into the appropriate partitions—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 single partitions of a table to avoid exporting the entire table in one operation. You can also perform the ANALYZE command on each partition concurrently to compute statistics 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, work on data in other partitions can still continue.

### **Greater Control Over Data**

Administrators may 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 may 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. Various local and global index strategies are provided. Partition operations may be performed in parallel. Partitions increase availability by containing media and application failures—applications not requiring data from a down or off-line partition continue to run without impact.

### **Easy-To-Use Administration**

Oracle8 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 contained in the table. To accomplish this, you simply 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 the data among the partitions. Also, stand-alone tables that use Oracle7™ Release 7.3 UNION ALL views may be converted into partitions of a single table. This allows users with very large Oracle7 databases to quickly benefit from Oracle8 partitioning without having to rebuild large tables.

## **IMPROVED DATA WAREHOUSE PERFORMANCE**

Oracle8 improves upon the enterprise-class data warehousing features of Oracle7 Release 7.3. Oracle7 has already proven to be significantly faster and easier to manage than other database products for data warehouse applications. Oracle8 further extends Oracle's lead with the following features:

- Enhanced star query processing
- More parallel operations
- Increased database size

### **Enhanced Star Query Processing**

Oracle8 introduces significant performance improvements to the processing of star queries, which are common in data warehouse applications. A star query or star schema occurs when there is one or more very large tables, often called a fact table, with relationships to multiple smaller tables called dimension tables. Oracle7 introduced the functionality of star query optimization, which provides great performance improvements for these types of queries. In Oracle8, however, star query processing has been significantly improved to provide even better optimization for star queries.

In Oracle8, an innovative new method for executing star queries has been introduced. Using a more efficient algorithm, Oracle8's star query processing provides a significant performance boost to data warehouse applications.

Oracle8 has superior performance with several types of star queries, including star schemas with “sparse” fact tables where the criteria eliminates a great number of the fact table rows. Also, when a schema has multiple fact tables, the optimizer efficiently processes the query. Finally, Oracle8 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 star query optimization algorithm, unlike that of Oracle7, does not produce any Cartesian-product joins. Oracle8 processes a star query in two basic phases. First, it retrieves exactly the necessary rows from the fact table. This retrieval is done via bitmapped 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 leverages 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.

### **New Parallel Options**

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

Oracle8 supports parallel insert, updates, and deletes into partitioned tables. Oracle8 also supports parallel inserts into non-partitioned tables. The parallel insert operation is similar to the direct path load operation available in Oracle7. The parallel insert into a table improves performance by formatting and writing disk blocks directly into the datafiles, bypassing the buffer cache and space management bottlenecks. In this case, each scan process of the query inserts data into a segment above the high watermark of the table. When all of the insert processes are complete, the transaction commits, and the high watermark is moved beyond the new segments.

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

### **Increased Database Size**

Oracle8 supports very large databases, extending into the hundreds of terabytes. Data warehouses often become increasingly large, and subsequently harder to manage because of their vast size. Oracle8 supports 64-bit file systems for significantly improved addressability of files and memory. Also, Oracle now supports unlimited and arbitrarily sized extents for table storage. Of course, storing large amounts of data is meaningless if you do not also have management tools for backing up, loading, and restoring the data. Oracle8 introduces partitioning, enhanced parallel operations, better backup and recovery tools, and other improvements to make managing large data volumes simpler.

## **MAINFRAME-CALIBER OLTP**

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

### **Very Large User Populations**

Numerous enhancements throughout the Oracle8 Server and the networking layer 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. The Oracle Connection Manager leverages multiplexing to forward data from numerous sources to a single server including cross-protocol, and acts as a firewall where needed. Shared database links multiplex many users into a database server with a single connection, reducing resource requirements especially for multitier application architectures.

### **Advanced Queuing**

Advanced queuing adds direct support in the database for deferring transaction execution to a later time and in a particular order. This capability allows you to decouple distributed applications and eliminates the dependency on external systems for applications requiring high scalability. Enqueue and dequeue operations may also be used to shift processing from within a transaction to a background process, thus improving the transaction response time. Also, queuing can be used to implement workflow applications that move data to a system as the state of the data changes. For example, moving orders from an order entry, to a shipping, and then to a billing system during the life of an order. Oracle8's advanced queuing can also work in conjunction with popular transaction processing (TP) monitor queuing systems such as Tuxedo's /Q.

### **Parallel Server Improvements**

Numerous enhancements in the Oracle8 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 Oracle8 Parallel Server significantly faster. Also, new global VS tables improve manageability.

Oracle8 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 Server to run on a given platform. Integrating the DLM within Oracle8 frees the Parallel Server from providing the least-common-denominator functionality across all platforms. This also allows Oracle8 Parallel Server to be available on platforms that previously did not support cross-node sharing of resources.

Oracle8 introduces numerous improvements to 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 10-15 percent improvement in 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 users to increase locality. Finally, Oracle8 reduces the memory overhead associated with fine-grain locking in an Oracle8 Parallel Server. You can now effectively use fine-grain locking of resources for significantly larger databases.

Oracle8 allows you to designate 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 servers. Using separate instances in an Oracle8 Parallel Server ensures that data warehouse queries do not affect the performance of OLTP applications.

Oracle8 introduces global fixed views for Oracle8 Parallel Server. This allows administrators to log into one instance of a Parallel Server and perform monitoring of the entire environment. This makes administration operations significantly simpler and improves the productivity of your database administrators.

### **Transparent Application Failover**

Should a node in the Parallel Server fail, transparent application failover migrates the users' connections and automatically re-establishes their sessions on another node. The users' applications continue to run, and the users 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 wish to incur on your client machine, 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—this eliminates the time required to reconnect to a failover instance.

Transparent application failover is not only useful for availability, but it can be used 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.

### **Improved TP Monitor Support**

Dynamic XA support improves performance for multitier applications with industry-standard XA-compliant TP monitors. Oracle8 improves its support of the XA interface with the following enhancements:

- Support for dynamic registration
- Support for loosely-coupled transaction branches
- Improved transaction recovery for Oracle<sup>®</sup> Parallel Server
- Session caching is no longer used
- SQLLIB is no longer needed for Oracle Call Interface<sup>™</sup> applications
- Availability of XA library with Oracle Parallel Server on all platforms
- No install script needed to run XA applications in Oracle8
- Oracle8 allows both global and local transactions within the same XA connection

## **Extended Backup/Recovery Subsystem**

Oracle8 server-managed backup and recovery provides a higher level of backup and recovery functionality from within the Oracle Server. Oracle8 maintains detailed information 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, Oracle8 analyzes the state of the database and determines the operations necessary to repair the database. Oracle8 then automatically performs those operations, greatly simplifying the recovery for the administrator and reducing the possibility of human error. A simple graphical user interface (GUI) interface within the Oracle<sup>®</sup> Enterprise Manager controls backup and recovery. An application programming interface (API) is also available for third parties who may wish to provide an alternative interface. A media management layer interfaces to popular third-party tape management products, such as Legato.

Multilevel, incremental backups greatly reduce the size of the backups, because only the changed blocks are backed up. This can substantially reduce the time required to backup 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.

Oracle8 also improves crash and instance recovery performance by limiting the number of reads and writes of blocks. This can provide significantly higher availability for mission-critical systems. Oracle allows you to configure the maximum amount of time required for instance recovery with initialization parameters. Thus, you can place stringent requirements on recovery time for situations where the buffer cache is very large.

## **BETTER SECURITY ADMINISTRATION**

Oracle8 includes 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 also contains improved password maintenance and administration. You can define a password profile to enforce a security scheme with Oracle8. Passwords can expire after a certain time, or be checked for validity. You can create your own password 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 also adds support for privileged database links, which make it unnecessary to embed a password in a database link, and data encryption services callable from Oracle Call Interface and PL/SQL<sup>™</sup>.

## REPLICATION

Oracle8 represents the most dramatic improvement in Oracle replication technology since the introduction of the advanced replication option. Oracle8 replication provides improved support for mass-deployment distributed systems, higher throughput failover configurations, and specialized data warehouse systems.

Salesforce automation and other mass deployment systems are now easier to administer and design with Oracle8's subsetting capabilities. 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 salesperson based on the ORDERS table. That snapshot can contain only the orders for each salesperson by selecting the orders based on criteria from an ASSIGNMENTS table. Each salesperson sees only his or her orders and only has to interact with the master table when uploading new orders.

Oracle8 also dramatically increases replication performance. 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 kernel, providing a significant boost in performance. Finally, the amount of data sent over the network for each replicated transaction has been dramatically reduced, and network round-trips are kept to a minimum.

Oracle8 replication also provides numerous improvements in manageability and ease of use. Significant enhancements to Oracle<sup>®</sup> Replication Manager, including wizards, make replicated environments even 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.

## OBJECT-RELATIONAL DATABASE

Oracle8 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 an incredibly 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 to provide a complete object-relational solution, Oracle addresses the need to simplify data modeling and extend the database with new datatypes.

Oracle's object-relational features include the following:

- Object types to extend the server in application-specific ways
- External procedures as object methods
- Client-side support of objects for fast navigation
- Evolution of an existing relational environment
- Development tools for object modeling
- Support for unstructured data (images, video, text, etc.)
- Support of open, industry standards Data Manipulation Java



Objects have been completely integrated throughout Oracle8, in all levels of the Server. All the industrial-strength properties of the Oracle universal data server are provided with objects including Oracle's sophisticated concurrency model, industry-leading performance, scalability, reliability, manageability, and availability.

### **Object Types**

Object types provide a way for extending 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 Zip Code. The object type can also have methods, such as Distance, for computing the distance between addresses. These methods can be written in either PL/SQL, C, or Java. 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 pointers 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 reusable 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.

Oracle8 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.

### **Calling External Procedures From Within the Database**

Oracle provides a safe, fast way for the database to make a call to an external program written in C, C++, or Java. 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 Fourier Transform (FFT). Also, you can use external procedures to interface with other applications or with specialized devices like embedded systems.

### **Client-Side Support of 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.

### **Evolution of Relational Environments**

Oracle8 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 solid foundation as the relational functionality, which means that users do not have to discard or rewrite their existing relational applications before migrating to Oracle8. Unlike other object-relational databases, this design then 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. Oracle8 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 may 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 web client. New and old applications can be based on the same data, but each has its own representation.

### **Development Tools for Object Modeling**

Developer/2000™ will support Oracle8's object model with a new visual object modeling tool. Also, many third-party tool vendors, such as Rational Software Corporation, support object development with Oracle8.

### **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 (or externally stored LOBs), can be replicated and can be an attribute of an object. You can also have more than one LOB per table/object. 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.

### **Support of Open, Industry Standards: Java**

Oracle has two initiatives for support of Java—an Oracle-provided JDBC driver more tightly integrated with Oracle's object types, and JSQL™ for embedding SQL statements into Java code. JDBC is available today for Java clients to access Oracle8. Oracle will provide its own JDBC drivers for better performance with Oracle8. JSQL allows you to include SQL statements in a Java application. The JSQL precompiler then converts the SQL into JDBC calls. This allows you to use existing SQL code in new Java applications.

### **OTHER ENHANCEMENTS**

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, not 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. PL/SQL performance is improved along with the performance of calling SQL from PL/SQL, and vice-versa.

Constraint processing in Oracle8 is 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 the end of transaction execution. 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.

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 new are fixed-length native Unicode 2.0 and Chinese GBK support.

Reverse key indexes reduce the “hot-spots” in indexes, especially primary key indexes. Unbalanced indexes can cause the index to become increasingly deep. Reverse key indexes cause the bytes of leaf-block entries to be reversed, and therefore prevent “sliding indexes.”

Oracle8 introduces several powerful application development features for more complex applications and ease of development. You can now modify data based on any relational view. A new kind of trigger allows you to replace INSERT, UPDATE, and DELETE statements 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. Finally, Oracle8 allows you to call 3GL code written in C from within the database. This allows you to integrate external hardware with database tables, use existing application code, or handle computationally complex processing.

## **MIGRATION AND INTEROPERABILITY**

A simple and fast migration utility rebuilds the data dictionary, converts the control files, log files, and data file headers. The migration utility converts any Oracle7 Release 7.1, 7.2, or 7.3 database into an Oracle8 database. Oracle7 applications run unchanged against Oracle8. Distributed commands from Oracle8 run against Oracle7, and vice-versa.



Oracle Corporation  
World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

Worldwide Inquiries:  
+ 1.415.506.7000  
Fax + 1.415.506.7200  
<http://www.oracle.com/>

Copyright © Oracle Corporation 1997  
All Rights Reserved

This document is provided for informational purposes only, and the information herein is subject to change without notice. Please report any errors herein to Oracle Corporation. Oracle Corporation does not provide any warranties covering and specifically disclaims any liability in connection with this document.

Oracle is a registered trademark, and Enabling the Information Age, Oracle8, Network Computing Architecture, Oracle7, Oracle Call Interface, PL/SQL, Developer/2000, and JSQL are trademarks of Oracle Corporation.

All other company and product names mentioned are used for identification purposes only and may be trademarks of their respective owners.