IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

ORACLE CORPORATION and ORACLE U.S.A. INC.,	)	
Plaintiffs/Counterclaim Defendants,	)	C.A. No. 06-414-SLR
v.	)	**JURY TRIAL DEMANDED**
EPICREALM LICENSING, LP,	)	**PUBLIC VERSION**
Defendant/Counterclaim Plaintiff.	)	

APPENDIX IN SUPPORT OF EPICREALM'S MOTION FOR PARTIAL SUMMARY JUDGMENT OF LITERAL INFRINGEMENT

VOLUME II of III

EXHIBITS 14 to 20

Richard L. Horwitz (#2246)
David E. Moore (#3983)
POTTER ANDERSON & CORROON LLP
Hercules Plaza, 6th Floor
1313 N. Market Street
Wilmington, DE 19899
Tel: (302) 984-6000
rhorwitz@potteranderson.com
dmoore@potteranderson.com

Attorneys for Defendant/Counterclaim Plaintiff epicRealm Licensing, LP

OF COUNSEL:

Harry J. Roper
George S. Bosy
Aaron A. Barlow
Patrick L. Patras
David R. Bennett
Paul D. Margolis
Benjamin J. Bradford
Emily C. Johnson
JENNER & BLOCK
330 N. Wabash Avenue
Chicago, IL 60611-7603
Tel: (312) 923-8305

Dated: July 31, 2008
Public Version Dated: August 7, 2008
876825 / 31393 / Oracle

TABLE OF CONTENTS

Exh.	Document
1.	U.S. Patent 5,894,554, Lowery et al.
2.	U.S. Patent 6,415,335, Lowery et al.
3.	Excerpt from Report of Plaintiffs' Expert Michael I. Shamos Concerning Invalidity, May 2, 2008
4.	Excerpt from Oracle WebServer User's Guide, Release 1.0 A34986-2, ORCL00941713-00941960
5.	The Oracle WebServer: A Technical Discussion, ORCL01042968-01042974
6.	Oracle WebServer Release 2.0, Feature List, 10/30/95, ORCL00937853-00937862
7.	Excerpt from Expert Witness Rebuttal Report on Non-Infringement By Dr. Paul C. Clark, June 6, 2008
8.	Excerpt from Oracle Database Application Guide – Fundamentals, 10g Release 2 (10.2), B14251-01
9.	Excerpt from Oracle Application Server Containers for J2EE User's Guide 10g (9.0.4) B10322-01, ORCL00338416-00338661
10.	Parties' Proposed Element-by-Element Claim Construction Chart, June 30, 2008
11.	Excerpt from Oracle Application Server Web Cache Administrator's Guide, 10g Release 2 (10.1.2), B14046-05, ORCL00513887-00514436
12.	Excerpt from Oracle XML DB Developer's Guide 10g Release 2 (10.2), B14259-02.
13.	Excerpt from Oracle Real Application Clusters Administration and Deployment Guide, 11g Release 1 (11.1), B28254-04, ORCL01972334-01972630
14.	What's New for Java DB, JDBC, and Database Web Servers in Oracle Database 10g, an Oracle White Paper, May 2005
15.	Excerpt from Oracle Database New Features Guide, 11g Release 1 (11.1) B28279-02
16.	Functional Specification for Runtime Load Balancing, RDBMS, 10gR2, ORCL01060148-01060193
17.	Excerpt from Plaintiff's Responses to Defendants' First Set of Interrogatories (1-6)
18.	Oracle World, ORCL01050220-01050280

Exh.	Document
19.	Excerpt from Oracle Application Server 10g Concepts, 10g (9.0.4), B10375-02, ORCL00370687-00370914
20.	Oracle9i Application Server, An Oracle White Paper, September 2002, ORCL00010730-00010792
21.	Excerpt from Oracle Call Interface Programmer's Guide, 11g Release 1 (11.1), B28395-03
22.	Design Specification for OCI Runtime Connection Load Balancing in 11g, 12/14/08, ORCL01310468-01310498
23.	Functional/Design Specification for Runtime Work Request Load Balancing using RAC Service Metrics, JDBC, 10gR2, 5/1/04, ORCL01054844-01054858
24.	Oracle HTTP Server, mod_plsql User's Guide, 10g Release 2 (10.2), B14337-01
25.	Excerpt from deposition transcript of Paul C, Clark, July 11, 2008
26.	Excerpt from deposition transcript of Steve Harris, April 18, 2008
27.	Excerpt from deposition transcript of Pushkar Kapasi, February 27, 2008
28.	Excerpt from deposition transcript of Matthew Mayerson, April 15, 2008
29.	Excerpt from deposition transcript of Katrina Montinola, February 12, 2008
30.	Excerpt from deposition transcript of Mark Nelson, February 22, 2008
31.	Excerpt from deposition transcript of Douglas Surber, February 28, 2008
32.	Oracle10iAS Planning Update, ORCL01066383-01066413

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

CERTIFICATE OF SERVICE

I, David E. Moore, hereby certify that on August 7, 2008, the attached document was electronically filed with the Clerk of the Court using CM/ECF which will send notification to the registered attorney(s) of record that the document has been filed and is available for viewing and downloading.

I further certify that on August 7, 2008, the attached document was Electronically Mailed to the following person(s):

Mary B. Graham
Morris, Nichols, Arsht & Tunnell LLP
1201 North Market Street
P.O. Box 1347
Wilmington, DE 19899
MGraham@MNAT.com

James G. Gilliland
Igor Shoiket
Townsend and Townsend and Crew LLP
Two Embarcadero Center
Eighth Floor
San Francisco, CA 94111-3834
OracleEpicrealm@townsend.com

Theodore T. Herhold
Robert J. Artuz
Eric M. Hutchins
Eric A. Mercer
Joseph A. Greco
Nitin Gupta
Townsend and Townsend and Crew LLP
379 Lytton Avenue
Palo Alto, CA 94301
OracleEpicrealm@townsend.com

Chad E. King
Townsend and Townsend and Crew LLP
1200 Seventeenth Street
Suite 2700
Denver, CO 80202
OracleEpicrealm@townsend.com

/s/ David E. Moore
Richard L. Horwitz
David E. Moore
Potter Anderson & Corroon LLP
Hercules Plaza – Sixth Floor
1313 North Market Street
P.O. Box 951
Wilmington, DE 19899-0951
(302) 984-6000
rhorwitz@potteranderson.com
dmoore@potteranderson.com

Exhibit 14

What's New for Java DB, JDBC, and Database Web Services in Oracle Database 10*g*

An Oracle White Paper
May 2005

What's New for Java DB, JDBC, and Database Web Services in Oracle Database 10*g*

Introduction 3
What's NEW FOr JDBC 3
Reminder - JDBC Features in Oracle Database 10*g* R1 3
New JDBC Features in Oracle Database 10*g* R2 4
What's NEW FOr JAVA-IN-THE-DATABASE 14
Tell me Again - Why Java In the Database? 14
Reminder – Java DB Features in Oracle Database 10*g* R1 14
New Java DB Features in Oracle Database 10*g* R2 14
WHAT'S NEW FOR JPUBLISHER & DATABASE WEB SERVICES 15
Tell Me Again -- Why Database Web Services 15
Reminder -- JPublisher and Database Web Services Features in Oracle Database 10*g* R1 15
New JPublisher and Database Web Services Features in Oracle Database 10*g* R2 16
Conclusions 16

What's New for Java DB, JDBC, and Database Web Services in Oracle Database 10*g*

INTRODUCTION

Oracle Database 10*g* Release 1 brought major enhancements to the Java runtime in the database (OracleJVM), to the Oracle JDBC, and to the Oracle Database Web Services. Release 2 goes further and brings completeness in terms of standard support, integration with the Oracle database, manageability, and performance. In this paper I will give you an overview of Java, JDBC and Web Services features across Oracle Database 10*g*, (Release 1 and Release 2). Through these capabilities, database developers will: (i) reduce their costs through productivity gain; (ii) reduce their risks through increase scalability, load balancing, high-availability; and (iii) extend the outreach and reach-in of their Oracle databases. The white paper " *Java and Web Services Devlpers Perspective on Oracle Database 10g*" looks at those features from the developer's perspective.

WHAT'S NEW FOR JDBC

Reminder - JDBC Features in Oracle Database 10*g* R1

Standard Support

The Oracle9i Release 2 JDBC drivers initiates JDBC 3.0 support with the following features: *transaction savepoints, toggling between local and global transaction, reuse of PreparedStatement, and JDK 1.4.x support for client JDBC drivers.* The Oracle Database 10*g* Release 1 JDBC brought the following JDBC 3.0 features:
Named Parameters: named parameters in CallableStatement and PreparedStatement, will enable JDBC applications to pass parameters by name as well as registering, and retrieving output parameters by name.
New Ref interface and Datalink: a DATALINK value references a file outside of the underlying data source that the data source manages. The JDBC 3.0 specification maps this new JDBC data type to the Java type java.net.URL.
J2EE Connector Architecture Resource Adapter: the Oracle JDBC driver can function as a JCA resource adapter for Oracle databases.
Connection Pool: the Implicit Connection Cache (see below) is Oracle's implementation of JDBC 3.0 connection pool. Furthermore this implementation brought advanced capabilities such as connection tagging for faster search and retrieval, weight and attribute-based search

JDBC Web RowSet (JSR-114): this standard API allows applications such as Web Services clients or J2EE components to fetch a collection of rows from database tables (or other data sources) and emit the result set in XML format, disconnected from the Data-Source. The Oracle Database 10g R1 JDBC implementation is based on the draft specification for JSR-114.

Manageability / Scalability/ High-Availability/ Ease of Use / Performance

End-to-end Tracing: JDBC drivers to propagate the middle-tier Web client id to the RDBMS engine so as to trace resource consumption from a Web Browser to the SQL corresponding operations.

Enhanced Oracle JDBC Datum support: the separation of datum classes from the rest of jdbc for "small" download size for datum users. Better (finer-grain) SQLException handling; enhanced conversion and arithmetic operations methods on datum to-and-from other datum.

Instant Client JDBC-OCI drive: a downloadable, low footprint bundling for simplifying JDBC-OCI install

New database types: IEEE DOUBLE, and IEEE FLOAT: Java and J2EE applications will retrieve and perform faster arithmetic calculations without loss of information, using reduced database storage space.

LONG-to-LOB conversion: conversion function for CLOBs and BLOB to/from LONG, RAW and LONG RAW, will simplify their manipulation and improve their performance.

Unlimited size LOB: ever wanted to manipulate more than 4 Giga binary data?

Enhanced support for VARRAY: data mining applications will perform aggregation, collection and general set operations, more efficiently.

PL/SQL Index table: this feature lets you send and receive PL/SQL tables in the type-4 "thin" JDBC driver.

New encryption algorithms: the type-4 "thin" JDBC driver now supports `3DES112` and `3DES168`.

Direct XA: performance optimization of JDBC XA operations when using the type-4 "thin" JDBC driver.

JDBC-Thin Proxy authentication: multiple middle-tier users/threads can share the same database connection under the same authenticated user.

New JDBC Features in Oracle Database 10*g* R2

Standard Support

Oracle Database 10g JDBC Release 2 achieves 100% JDBC 3.0 compatibility by adding support for Auto-Generated Key Retrieval, Result Set Holdability as described below.

JDBC 3.0 Retrieval of Auto-Generated Keys: Oracle database sequences and triggers can be used to generate a unique key when a new row is inserted. JDBC 3.0 introduce three interfaces that support the retrieval of the auto-generated keys; namely

`java.sql.DatabaseMetaData, java.sql.Statement` and `java.sql.Connection`

- JDBC also offers "DML in the RETURNING clause" ss Oracle extension to support Retrieval of Auto Generated Keys.

JDBC 3.0 Result Set Holdability: this feature enables applications to decide whether the ResultSet object should be open or closed, when a commit operation is performed. The commit operation could be either implicit or explicit. The Oracle JDBC supports `ResultSet.HOLD.CURSORS_OVER_COMMIT` which is the default option for the Oracle database itself.

In addition, Oracle Database 10g JDBC Release 2 supports the final specification of *JDBC RowSet JSR-114,* a JDK 5.0 requirement. All 5 types of RowSets as JavaBeans: JdbcRowSet, CachedRowSet, WebRowSet, FilteredRowSet, and JoinRowSet are supported. In addition JAXP 1.2 (SAX 2.0 and DOM), and W3C XML schema, are also supported.

Manageability / Scalability/ High-Availability/ Ease of Use / Performance

JDBC-OCI support for Proxy Authentication: parity with JDBC-Thin Proxy Authentication (see above

JDBC-Thin support for SSL: JDBC leverages JSSE to provide a secure communication between client and server when specified (presence of (PROTOCOL=tcps) in the connection description).

JDBC-Thin support for TNSNAMES.ORA Lookup: similar to JDBC-OCI, both client-side and server-side JDBC-Thin can now read the local TNSNAMES.ORA file to resolve service name or TNS alias. Consequently Java in the database will also be in position to reference external Oracle database, using service name. See the JDBC documentation for more details.

Caching XAConnections: The JDBC connection cache has been enhanced to cache XAConnections, since this is a first implementation of such capability and we are looking for feedback (mostly Application Servers and ERP vendors).

New JDBC Connection Services

The Oracle Database 10g JDBC drivers introduce new connection services in term of performance, scalability and high-availability. In this section I'll describe the new Implicit Connection Cache (performance, scapalability), the Fast Connection Failover, (scalability, high-availability), and Runtime Load Balancing of JDBC Connection Requests (performance, scalability) in RAC and GRID environments,

Connection Cache Manager

The Connection Cache Manager provides a centralized way to manage one or more connection caches. It consists in a single instance of Connection Cache Manager per VM. Each Connection Cache Manager instance manages all connection caches

existing in the JDK/JVM. This instance of the Connection Cache Manager must be obtained before using any of the Connection Cache Manager functionality.

The Connection Cache Manager plays two major roles: cache management and binding connection caches to data sources.

Cache Management and Maintenance

The Connection Cache Manager is responsible for creating, destroying and maintaining the sanity of each cache. A programmatic API allows interaction with the Connection Cache Manager (i.e., explicitly create a cache). A connection cache may also be created transparently during the DataSource access. In fact, the first connection request on a DataSource attempts to create the cache, however, if it already exists then, the cache creation is skipped and the connection request is simply routed to the cache, which retrieves a connection.

Binding a Connection Cache to the DataSource

The Connection Cache Manager associates a connection cache with its DataSource, during the creation and re-initialization of the cache. The associations ends when the cache is removed/destroyed. This ensures an efficient access and connections retrieval when `getConnection()` request is invoked on the DataSource object.

Multi-cache Support: The Connection Cache Manager supports coexistence of more than one cache. Each cache identified by a unique name, is tightly bound to a DataSource. Each cache is either created transparently when getConnection() requests are made on a cache enabled DataSource or is created explicitly in the middle tier via the Connection Cache Manager API. Using multiple cache enabled DataSources, provides the following benefits:

Request connections to more than one DataSource, especially when each of these DataSources point to a separate underlying database.

Manage all caches configured, via the Connection Cache Manager. This makes managing caches easy, without the burden of unnecessary book keeping in Application Server code (code using Connection Cache Manager).

There are no limits imposed by the Connection Cache Manager on the number of caches that can be created. This number is limited by the resources available on the JVM and the Oracle database server.

Once a cache is created, it may either be explicitly removed via the Connection Cache Manager, or is removed when the DataSource is closed after use, via the DataSource close() API.

Connection Cache Properties

It is easy to set limits and tuned the connection cache using a set of connection cache properties. Cache properties are set either while creating the cache for the first time, on the DataSource, or when the re-initializing the cache, via the Connection Cache Manager.

MinLimit: this sets the minimum number of PooledConnections the cache maintains. This guarantees that the cache will not shrink below this minimum limit. This property does not initialize the cache with minimum number of connections[1]. Refer to ***InitialLimit*** property for priming of cache information. The default value is 0.

MaxLimit: this sets the maximum number of ***PooledConnections*** the cache can hold. The default value is unbound, meaning that there is no MaxLimit assumed. Since the connection cache does not assume the maximum limit, the connection cache is only limited by the number of database sessions configured for the database. In other words, connections in the cache could reach as much as the database allows.

InitialLimit: this sets the size of the connection cache when the cache is initially created or reinitialized. When this property is set to a value greater than 0, that many connections are pre-created and ready to use. This property is typically used to reduce the "ramp up" time in priming the cache to its optimal size. The default value is 0.

Dynamic Reconfiguration Support

Implicit Connection Cache allows dynamic reconfiguration of the cache properties by reinitializing the cache using the specified new set of cache properties. The new properties take effect on all ***PooledConnections*** newly created as well as *PooledConnections* that are not in use. For connections that are in use, the new properties take effect only after they are returned to the cache.

MONITORING THE CONNECTION CACHE

These APIs may be invoked on the Connection Cache Manager to obtain cache information, such as the number of connections checked out, or the number of connections available in the cache.

int getNumberOfAvailableConnections(String cacheName)

This API returns the number of connections in the connection cache, that are available for use. The value returned is a snapshot of the number connections available in the connection cache at the time the API was processed and hence a statistical value.

int getNumberOfActiveConnections(String cacheName)

This API returns the number of checked out connections. These are connections that are active or busy, and hence not available for use. The value returned is a snapshot of the number of checked out connections in the connection cache at the time the API was processed and hence a statistical value.

java.util.properties getCacheProperties(String cacheName)

[1] As was the case with OracleConnectionCacheImpl

Retrieves the properties of the specified cache.

`String[] getCacheNameList()`

This API returns all the connection cache names that are known to the Connection Cache Manager. The cache name(s) may then be used to manage connection caches using the Connection Cache Manager APIs.

Implicit Connection Cache

The Implicit Connection Cache provides a rich set of features including: transparent or implicit access to the connection cache, support for heterogeneous authenticated connection, the ability to refresh or recycle stale connections from the cache, connection retrieval based on user defined attributes, connection retrieval based on attributes and weights.

Abandoned Connection Support

Abandoned or orphaned connections are those that have been checked out of the connection cache, but never returned to the cache, for numerous reasons. Implicit Connection Cache automatically detects and reclaims these connections back to the cache, based on the `AbandonedConnectionTimeout` property on the connection cache. Setting this property to a valid timeout value, starts a heartbeat monitoring process[2] on this connection that is checked out.. When a heartbeat is not registered on the connection for the specified period of time, the connection is considered abandoned and is automatically claimed and put back to the cache.

Statement Caching Support

The `MaxStatementsLimit` connection cache property sets the maximum statements to keep open for a connection cache. If cache is reinitialized with this property and if more cursors are cached than specified, the extra cursors are closed. The default is 0.

Connection Recycling Support

Over a period of time, connection cache accumulates stale connections. There are two modes for recycling or refreshing stale connections in the cache: `REFRESH_INVALID_CONNECTIONS` and `REFRESH_ALL_CONNECTIONS`.

- With `REFRESH_INVALID_CONNECTIONS` each PooledConnection in the cache is checked[3]; if invalid, the connection's resources are removed and replaced with a new PooledConnection.
- WIth `REFRESH_ALL_CONNECTIONS`, all the available connections in the cache are closed and replaced with new valid physical connections.

Connection Retrieval based on User defined Attributes

---

[2] Any SQL activity on the connection
[3] The validity test is as simple as: *select 1 from dual;*

The implicit connection cache supports the idea of striping (i.e., tagging) a connection with user-defined attributes before returning it to the cache. These attributes can later be used to retrieve the same connection from the cache.

Example 1: Retrieval based on Connection Attribute NLS_LANG

```
// Look up the datasource object
javax.sql.DataSource ds = (javax.sql.DataSource)
ctx.lookup(MyOracleDataSource);

// get a connection from MyCache
java.util.Properties connAttr = null;
connAttr.setProperty("NLS_LANG", "ISO-LATIN-1");
conn = ds.getConnection(connAttr); // retrieve
connection - NLS_LANG
…
conn.applyConnectionAttributes(connAttr); // apply
attributes

Statement stmt = conn.createStatement();
stmt.execute("select empname from emp");
…
conn.close(); // release the connection back to MyCache
```

Example 2: Retrieval based on Connection Attribute Isolation level

```
…
java.util.Properties connAttr = null;
connAttr.setProperty("TRANSACTION_ISOLATION",
"SERIALIZABLE");
conn = ds.getConnection(connAttr); // retrieve
connection that matches Transaction Isolation
…
conn.close(connAttr); // another way to  apply
attributes to the connection
```

Example 3: Retrieval based on Connection Attribute, Connection tags (Reserved connection)

```
…
java.util.Properties connAttr = null;
connAttr.setProperty("CONNECTION_TAG",
"JOE'S_CONNECTION");
conn = ds.getConnection(connAttr); // retrieve
connection that matches Joe's connection
…
conn.close(connAttr); // apply attributes to the
connection
…
conn = ds.getConnection(connAttr); // This will retrieve Joe's connection
```

Connection Retrieval Based on Attributes and Weights

The cache property `CacheAttributeWeights` are ***`java.util.Properties`*** that allows setting attribute weights. Weights are assigned to each Key in a ***`ConnectionAttribute`***. Each weight is an integer value that defines the cost of the Key. Once the weights are specified on the cache, connections may be retrieved using

`getConnection(connectionAttributes)` which searches for a connection that satisfies a combination of:

- Key/Value match on a connection from the cache
- Maximum total weight of all the keys of the `connectionAttributes` that were matched on the connection

Consider the following example.

A cache is configured with `CacheAttributeWeights` as follows:

```
java.util.properties cacheProps = new Properties();
java.util.properties cacheWeights = null;

cacheWeights.setProperty("NLSLANG", "10");
cacheWeights.setProperty("SecurityGroup", "8");
cacheWeights.setProperty("Application", "4");
…
// set weights on the cache
cacheProps.put(CacheAttributeWeights, cacheWeights);
                    …
```

Once the weights are set, a connection request could be made as:

```
java.util.properties connAttr = null;
connAttr.setProperty("NLSLANG", "ISO-LATIN-1");
connAttr.setProperty("SecurityGroup", "1");
connAttr.setProperty("Application", "HR")

// Request connection
ds.setCacheName("MyCache");
// First retrieval of connection from myCache
conn = ds.getConnection(connAttr);
…
conn.close(connAttr); // apply attributes on the
connection
…
// Next retrieval finds the connection in the cache
conn = ds.getConnection(connAttr);
…
```

This *`getConnection()`* request tries to retrieve a connection from the cache, *`MyCach`e*. The connection matching and retrieval from the cache involves the following:

- An exact match: a connection that satisfies the same attribute values and all the Keys (NLS_LANG, SecurityGroup, and Application).
- An exact match is not found: if the *`ClosestConnectionMatch`* property is set, a closest match based on the attribute key/value and their associated weights are used. For example, a closest match may be a connection that contains attribute match of NLS_LANG and APPLICATION, but not SECURITY_GROUP. It is also possible to find connections that match some keys of the original list, but their combined weights are the same. For example, connection1 could have a match of NLS_LANG with its associated weight of 10, where as connection2 may have an attribute match of SECURITY_GROUP and APPLICATION with their combined weight of 12. In this case, it is desired that connection2 is returned. In other words,

connection2 is the closest match and more expensive to reconstruct (from the caller's perspective) as opposed to connection1. When none of the *connectionAttributes* match a new connection is returned. The new connection is created using the user and password set on the DataSource.

Once the connection is returned, the user can invoke `getUnMatchedConnectionAttributes()` API on the connection object to return a set of attributes (`java.util.Properties`) that did not match the criteria. These unmatched attribute list is used by the caller (or application) to re-initialize these values before using the connection.

Applying Connection Attributes to a Cached Connection

There are two ways to apply a connection attribute to a connection. (1) By calling `applyConnectionAttributes(java.util.properties connAttr)` API on the connection object. This simply sets the supplied attributes on the connection object. Its possible to apply attributes incrementally using this API, allowing users to apply connection attributes over multiple calls. For example, *NLS_LANG* may be applied by calling this API from module A. The next call from module B, may then apply the *TXN_ISOLATION* attribute, and so on. (2) By calling `close(java.util.properties connAttr)` API on the connection object. This API closes the logical connection and then applies the supplied connection attributes on the underlying PooledConnection (physical connection). The attributes set via this `close()` API overrides the attributes, if any, set using the `applyConnectionAttributes()` API.

The following example shows a call to `close(connectionAttributes)` API on the connection object, that lets the cache apply the matched `connectionAttributes` back on the PooledConnection, before returning it to the cache. This ensures that when a subsequent connection request with the same connection attributes is made, the cache would find a match.

// Sample connection request and close

```
java.util.properties connAttr = null;
connAttr.setProperty("NLSLANG", "ISO-LATIN-1");
conn = ds.getConnection(connAttr); // request connection
based on attributes
java.util.properties unmatchedAttr =
conn.getUnMatchedConnectionAttributes();
…
--- App Server code applies unmatched attributes to the
connection, either by
    calling PL/SQL procedures or SQL, before using the
connection. ---
--- work ---
…
conn.close(connAttr); // apply attributes to connection
```

Fast Connection Fail-Over

Fast Connection Failover works in conjunction with the Implicit connection caching mechanism and a RAC database.

When a connection cache is setup for a multi-instance RAC database, database connections may end up on any of the RAC instances, as distributed by the database listener. When an instance goes down, due to an instance or host failure, it brings down all the connections with it. For example, in a two node, two instance RAC cluster, consider a scenario where each of the instances were to have 50 connections each, for a connection cache primed up to 100 connections in the cache. If one of RAC instances goes down, there would instantly be 50 bad connections in the cache. This could potentially result in multiple bad connections being returned from the cache, which in turn could result in application or browser page errors.

The Fast Connection Failover support enables an automatic detect-and-fix mechanism to handle any instance or host failures in a RAC environment. The mechanism is enabled on a cache enabled DataSource, by simply flipping the DataSource property *FastConnectionFailoverEnabled* to *true*.

```
// Example to show binding of OracleDataSource to JNDI
// with relevant cache properties set  on the
DataSource.

import oracle.jdbc.pool.*; // import the pool package

Context ctx = new IntialContext(ht);
OracleDataSource ods = new OracleDataSource();

// Set DataSource properties
ods.setUser(“Scott”);
ods.setPassword(“tiger”);
ods.setConnectionCachingEnabled(True);
ods.setConnectionCacheName(“MyCache”);
ods.setConnectionCacheProperties(cp);
setURL("jdbc:oracle:thin:@(DESCRIPTION=
   (LOAD_BALANCE=on)
   (ADDRESS=(PROTOCOL=TCP)(HOST=host1) (PORT=1521))
   (ADDRESS=(PROTOCOL=TCP)(HOST=host2)(PORT=1521))
   (CONNECT_DATA=(SERVICE_NAME=service_name)))");
ods.setFastConnectionFailoverEnabled(true); // Enable
fast connection failover
ctx.bind(“MyDS”, ods);
…
ds = lookup(“MyDS”);  // lookup DataSource from the
cache
// implicitly create connection cache, that is set up
for fast connection failover
conn = ds.getConnection();
…
conn.close();  // return connection to the cache
…
ods.close() // close datasource and cleanup the cache
```

The Fast Connection Failover mechanism works by handling Service DOWN or UP events and Host DOWN events. The DOWN event processing always cleans up the bad connections from the cache. The UP event processing does Load Balancing of connections in the cache.

The Fast Connection Failover functionality provides the following advantages:

Fast shutdown of connections in the connection cache, when RAC instance/node failures are detected: this prevents bad or invalid connections being handed out to application connection requests. For an application that depends on the Implicit connection cache for total connection management, the Fast Connection Failover mechanism provides maximum connection availability.

Load Balancing of connections when a RAC UP event is generated: in this model, connections are established and load balanced to all active RAC instances, without waiting for application connection retries/requests. Note that for RAC setups, the service is almost always instantly available, except when the entire service is down.

Runtime Load-Balancing of Connection Requests

In Oracle Database Release 1, connections to RAC instances were randomly established by the TNS Listener, without any consideration of how busy an instance already was. In Oracle Database 10g Release 2, RAC nodes emit Service Metrics every 30 seconds (i.e., RLB events), for each service on each instance. Each event payload contains a state or health of the service in question and the percentage of idleness (in other words, the percentage of additional work related to the service in question that the instance can handle). The RAC Automatic Workload Repository (AWR) contains the directives for the workload management goal (i.e., throughput or service time). The Oracle JDBC connection cache and the Connection Cache Manager have been extended to exploit RAC service metrics. When RLB is enabled (see configuring RLB, later), a connection belonging to the least loaded instance for the service is retrieved from the cache and returned to the requestor. By routing connections requests based on workload feedback, the workload will be evenly distributed across all nodes that furnish the service in question; thereby optimizing resource utilization (which is the whole purpose of Enterprise Grid Computing).

Configuring Runtime Load-Balancing

The connection pool Manager routes the connection request to the least loaded RAC instance using different policies.

WHAT'S NEW FOR JAVA-IN-THE-DATABASE

Why Java In the Database?

Java in the database is a response to customers in quest for portability, reuse and the ability to do things that cannot be done in PL/SQL (a lot of things!). Java is indeed portable across hardware platform, across RDBMS (no vendor lock-in), and across tiers – furthermore Java can be migrated to/from middle-tier (J2EE, POJOS, JDBC) to the database (Java stored procedures). Other important benefits Java brings are the ability to just reuse, the huge Java class libraries produced over years by millions of developers; and also the ability to reuse the pervasive Java skills.

Reminder – Java DB Features in Oracle Database 10*g* R1

On standard support front, the OracleJVM runtime achieved full compatibility with J2SE 1.4 including stringent Java Security (JCE, JSSE, etc), headless AWT, JAXP, etc. On the performance front, OracleJVM implements self-tuning Java pool; optimized memory management (end-of call migration) for dedicated processes; implemented a new, faster server-side JDBC driver; rewrote a new, faster bytecode verifier (loadjava). Fixed EJB Call-out, which completed the set of call-out capabilities: HttpClient, RMI/JRMP, RMI/IIOP, and SOAP Client. JPublisher utility now offers the generation of client-side stub for direct invocation of Java in the database, without a user-supplied Call Spec (PL/SQL Wrapper).

New Java DB Features in Oracle Database 10*g* R2

Optimize OracleJVM Garbage Collector behavior

A new parameter PGA_AGGREGATE_TARGET influences OracleJVM GC behavior at the end of the SQL call. Low values will optimize memory at the expense of the speed while high values will optimize execution speed at the expense of memory.

Java Audit

With pre-Release 2 of Oracle Database 10g, only Java classes/methods published to SQL through PL/SQL wrapper can be audited through SQL Audit. With this enhancement, java sources, java classes or java resources can be directly/explicitly audited. The following table summarizes the options and scope of Java audit.

Java Audit Option	SQL Statement to be Audited
CREATE JAVA SOURCE	CREATE JAVA SOURCE CREATE OR REPLACE JAVA SOURCE
ALTER JAVA RESOURCE	ALTER JAVA RESOURCE
DROP JAVA SOURCE	DROP JAVA SOURCE

CREATE JAVA CLASS	CREATE JAVA CLASS CREATE OR REPLACE JAVA CLASS
ALTER JAVA CLASS	ALTER JAVA CLASS
DROP JAVA CLASS	DROP JAVA CLASS
CREATE JAVA RESOURCE	CREATE JAVA RESOURCE CREATE OR REPLACE JAVA RESOURCE
ALTER JAVA RESOURCE	ALTER JAVA RESOURCE
DROP JAVA RESOURCE	DROP JAVA RESOURCE

WHAT'S NEW FOR JPUBLISHER & DATABASE WEB SERVICES

Why Database Web Services

Database Web Services bring three benefits to database users:

- Database as Web Services Provider extends database's client-base to Web Services clients by allowing the execution of database operations and data retrieval through Web Services mechanisms.
- Database as Web Services Consumer extends database's reach. The inclusion of external data as part of a SQL query or database batch processing is not new (see gateways and other related mechanisms), what's new is when the external data is only available as dynamic data (produced on demand), typically available as a Web services (i.e., temperature, stock price, IRS table, etc).
- Reusing the Web Services framework of Oracle Application Server with the database, allows instant interoperability, consistent Web services development and deployment; furthermore it integrates the Database in the Service Oriented Architecture beyond basic SOAP/HTTP; Database Web Services inherit Web Services Interoperability, WS-Security, WS-Reliability, and WS Management.

Reminder -- JPublisher and Database Web Services Features in Oracle Database 10*g* R1

JPublisher

Complete JDBC types support: new supported types including: NCHAR, Timestamp, SQLJ objects, and SQL Opaque. Easier access to native PL/SQL types through predefined type conversions. And the generation of client stub for invoking Java-in-the-database.

Database Web Services

Support for Database as Services Consumer

For a given WSDL, generate the Java proxy classes, the PL/SQL wrappers and load the appropriate files in the database.

Support for Database as Web Services Provider

- Publishing PL/SQL, Java Stored Procedure, SQL Queries, and SQL DML as Web Services
- Mapping PL/SQL Types (CLOB, BLOB)
- Mapping of REF CURSORs and Result Sets

New JPublisher and Database Web Services Features in Oracle Database 10*g* R2

- Support JDBC types for server-side Java calling
- Publishing Streams/Advance Queue as Web Services
- Support complex types in Web Services callout
- Handle connection loss through data sources
- Generate SQLJ runtime free code (works as pure JDBC code)

CONCLUSIONS

This paper gave you a brief overview of Java, JDBC and Database Web services enhancements in Oracle Database 10g release 1 and release 2. The net benefits are: faster application development, low cost application development, faster applications execution, low cost platform integration, dynamically scaling up and down RAC/Grid configurations, extending the reach of the database, extending the capabilities of the database, reuse of existing database entities as Web Services.



What's New for Java DB, JDBC, and Database Web Services in Oracle Database 10*g*
May 2005
Author: Kuassi Mensah

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

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Copyright © 2005, Oracle. All rights reserved.
This document is provided for information purposes only and the contents hereof are subject to change without notice.
This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.
Oracle, JD Edwards, and PeopleSoft are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Exhibit 15

Oracle® Database

New Features Guide

11*g* Release 1 (11.1)

B28279-02

September 2007

Oracle Database New Features Guide is addressed to people familiar with previous versions of Oracle Database who would like to become familiar with features, options, and enhancements that are new in this release of the database.

Oracle Database New Features Guide, 11*g* Release 1 (11.1)

B28279-02

Copyright © 2001, 2007, Oracle. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software—Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

1

Oracle 11*g* Database New Features

This chapter contains descriptions of all of the features that are new to Oracle 11*g* Database Release 1 (11.1). This chapter contains the following sections:

- Application Development
- Availability
- Database Overall
- Business Intelligence and Data Warehousing
- Clustering
- Content Management Services
- Fault Diagnosability
- Information Integration
- Location Services and Specialty Data
- Manageability
- Performance
- Security
- Windows

1.1 Application Development

The following sections describe new Application Development features for Oracle Database 11*g* Release 1 (11.1).

1.1.1 APIs and Precompilers

The following sections describe new Oracle Database 11*g* features in application programmatic interfaces. These features improve performance and scalability of applications and enable easy deployment of the Oracle client-side stack.

1.1.1.1 Client-Side Query Cache

This feature enables caching of query result sets in client memory. The cached result set data is transparently kept consistent with any changes done on the server side.

Applications leveraging this feature see improved performance for queries which have a cache hit. Additionally, a query serviced by the cache avoids round trips to the server for sending the query and fetching the results. It also reduces the server CPU that

would have been consumed for processing the query, thereby improving server scalability.

See Also:

Oracle Call Interface Programmer's Guide for details

1.1.1.2 OCI Client Session NLS Character Set Status Information

This new feature adds client session NLS and character set information to the `V$SESSION_CONNECT_INFO` and `GV$SESSION_CONNECT_INFO` views.

This enhancement provides DBAs with access to client-side session character set information for all clients. This information is especially useful in troubleshooting problems.

See Also:

Oracle Database Globalization Support Guide for details

1.1.1.3 Precompilers: Additional Array INSERT and SELECT Syntax Support by Pro*C/C++ and Pro*COBOL

In this Oracle Database 11*g* release, there is added support for IBM DB2 array `INSERT` and array `SELECT` syntax for Pro*C/C++ and Pro*COBOL.

Adding support for IBM DB2 array `INSERT` and array `SELECT` syntax helps with the migration of IBM DB2 installations to Oracle.

See Also:

*Pro*C/C++ Programmer's Guide* for details

1.1.1.4 Precompilers: Dynamic SQL Statement Caching in Pro*C/C++ and Pro*COBOL

Application developers can now precompile Pro*C/C++ and Pro*COBOL applications with the new statement caching option.

The potential benefit of this feature is an increase in performance by enabling statement caching in precompiled programs. The performance improvement is achieved by removing the overhead of parsing the dynamic statements on reuse.

See Also:

*Pro*C/C++ Programmer's Guide* for details

1.1.1.5 Precompilers: Fix Execution Plan in Pro*C/C++ and Pro*COBOL

This new feature focuses on fixing an execution plan for a Pro*C/C++ and Pro*COBOL application by using the outline feature.

By using the outline feature, you can ensure that the performance of the application is not affected when modules are integrated and deployed into different environments.

1.1.1.6 Precompilers: Flexible B Area Length for Pro*COBOL

The length of B Area for the Pro*COBOL application can now be from 8 to 253 columns.

The length of B Area for a Pro*COBOL program was limited to 72 when the format is set to `ANSI`. COBOL compilers now can support B Area length up to 253. This

provides a programmer with the flexibility to type a line that is longer than 72 columns.

1.1.1.7 Precompilers: Implicit Buffered Insert in Pro*C/C++ and Pro*COBOL

Some applications that require periodic insertion of new data use a single `INSERT` statement within a loop. When queried by a command-line argument, the precompiler buffers the rows being inserted, reducing network round trips.

The benefit is that an implicit buffered `INSERT` provides performance improvement that is comparable to using the precompiler array interface.

See Also:

*Pro*C/C++ Programmer's Guide* for details

1.1.1.8 Precompilers: Platform Endianness Support by Pro*COBOL

Oracle always stores Unicode data (UTF16) in Big Endian representation. Linux and Windows have Little Endian representation and Solaris has Big Endian representation. With this enhancement, Unicode data can be fetched in Little Endian format on Windows and Linux platforms using a Pro*COBOL application.

When a UTF16 (UCS2) string is fetched in `PIC N` variable in a Pro*COBOL application running on Little Endian platforms, the application can now be precompiled with the appropriate command-line option to have the correct platform endianness.

See Also:

*Pro*COBOL Programmer's Guide* for details

1.1.1.9 Precompilers: Scrollable Cursor Support in Pro*COBOL

Pro*COBOL now supports scrollable cursors to fetch data in a nonsequential manner.

A scrollable cursor enables you to access the rows of a database result set in a forward, backward, and random manner. This enables the program to fetch any row in the result set.

1.1.1.10 Precompilers: SQL99 Syntax Support by Pro*C/C++

With this enhancement, the SQL99 syntax for `SELECT`, `INSERT`, `DELETE`, and `UPDATE` statements and the body of the cursor in a `DECLARE CURSOR` statement is supported by Pro*C/C++.

Oracle Database supports most of the SQL99 syntax. With this enhancement, Pro*C/C++ also supports the SQL99 syntax that is supported by Oracle Database.

See Also:

*Pro*C/C++ Programmer's Guide* for details

1.1.2 Application Express

The following sections describe new features for Oracle Application Express. Application Express provides an application development tool that is built into the database and is installed in the database by default which is different from Oracle Database 10*g* implementation.

1.1.2.1 Application and Schema Comparison

New application and schema comparison capabilities enable you to identify differences between two selected applications.

You can also identify differences between objects in two different schemas.

See Also:

Oracle Database Application Express User's Guide for details

1.1.2.2 Drag and Drop Item Layout

The Drag and Drop Layout page enables you to interactively reorder items within a given region, change select item attributes, create new items, and delete existing items.

See Also:

Oracle Database Application Express User's Guide for details

1.1.2.3 Flash Charts

Oracle Application Express includes support for 18 different types of Flash charts. You can create Flash charts when creating a new page or by defining a Chart region.

This release also includes a utility that enables you to convert existing Scalable Vector Graphics (SVG) charts to Flash.

See Also:

Oracle Database Application Express User's Guide for details

1.1.2.4 Friendly URL Syntax to Facilitate Bookmarks

By specifying zero as the session ID, you can now make it easier for application users to bookmark pages within an application.

This features makes all public page links consistent, making them easier to bookmark.

See Also:

Oracle Database Application Express User's Guide for details

1.1.2.5 Improved Web Services

Oracle Application Express now supports more loosely defined Web services types and document styles. Both of these enhancements enable support for Web services created with JDeveloper and Oracle BPEL synchronous processes. You can now interact with Web services over a Secure Sockets Layer (SSL) and work with Web services that require basic authentication.

Finally, this release also enables you to manually create a Web services reference even if the Application Express engine cannot correctly parse a WSDL document.

See Also:

Oracle Database Application Express User's Guide for details

1.1.2.6 Improved Workspace Management

This release also includes a number of workspace management enhancements. For users requesting new workspaces or requesting additional storage, you can control what sizes display.

You can also request an e-mail of your workspace names. And, finally, you can view a log of login attempts.

See Also:

Oracle Database Application Express User's Guide for details

1.1.2.7 Microsoft Access Migration

Application Migration enables you to migrate a Microsoft Access application and generate an Oracle Application Express application.

This feature provides an easy way to migrate applications from Microsoft Access to Oracle Express.

See Also:

Oracle Database Application Express User's Guide for details

1.1.2.8 New Password and Account Controls

You can now specify rules for password expiration, mandate strong passwords (that is, specify the minimum number of and type of characters), require that users change their passwords on first use, and lock accounts.

With this new feature, you can configure account login controls and password policies across a development instance in Oracle Application Express Administration Services. You can also specify controls for a specific account under Account Controls on the Edit User page.

See Also:

Oracle Database Application Express User's Guide for details

1.1.2.9 Page and Region Caching

You can improve the performance of your application by taking advantage of page and region caching. New Cache attributes have been added to the Page attributes and Region Definition pages.

Page caching works well for static pages and region caching is a good choice for regions such as lists that do not have any conditions or regions containing HTML text.

See Also:

Oracle Database Application Express User's Guide for details

1.1.2.10 PDF Printing

You can now include the ability to print a report by exporting a report region to PDF. You can also define report queries and print attributes for report regions as Shared Components making them accessible to all pages within an application.

The advantage of this feature is that it provides production of high quality, easily printable reports.

See Also:

Oracle Database Application Express User's Guide for details

1.1.3 Extensible Indexing

The following sections describe extensible indexing features for Oracle Database 11*g* Release 1 (11.1).

1.1.3.1 Create Rules and Actions for DML Events With Rules Manager

Rules Manager now adds the ability to treat DELETE and UPDATE operations on a table as events that drive the rule evaluation and action execution (INSERT operations are already supported). In addition to pre-commit evaluation of rules based on trigger, this release supports post-transaction evaluation through use of the Change Notification feature of Oracle Database.

Because rules can now be processed during UPDATE and DELETE statements, developers can now model the entire business process for OLTP workloads in addition to the current "append-only" workloads characteristic of data warehouse and sensor-based applications.

See Also:

Oracle Database Rules Manager and Expression Filter Developer's Guide for details

1.1.3.2 Enhanced Rules Manager

Rules Manager now includes a PL/SQL API to store reusable and shareable rule conditions in SQL `WHERE` clause format. Complex rule conditions can be assembled from these shared elements.

In addition, rules can now be enabled and disabled through standard SQL.

Application developers can reuse rule conditions to speed the rule definition process. Since these shared conditions are stored once, changes propagate to all the complex rules referring to them. For example, if a rule component defines a "suspicious bank transaction" as involving an international bank transfer over $10,000, a change to the amount, location, or other condition needs to be made only once and it is updated in all complex rules using "suspicious bank transaction" in their event scenario.

Rules can be turned on and off to facilitate rule testing and control introduction into the system as well as to reflect changes in business conditions.

See Also:

Oracle Database Rules Manager and Expression Filter Developer's Guide for details

1.1.3.3 New Capabilities for Domain Index Operations

A new domain index callout updates table and column name metadata for a domain index during an `ALTER TABLE ... RENAME` or `ALTER TABLE ... RENAME COLUMN` command to ensure domain index metadata is kept up-to-date.

A partition-specific parameter string provides control over placement of domain index data during partition maintenance operations.

The SQL `MERGE` statement can now be used with a domain index.

See Also:

Oracle Database SQL Language Reference for details

1.1.3.4 Oracle Text Predicates and Aggregate Operators are Included in Rule Conditions

The Rules Manager feature of Oracle Database now supports the use of the Oracle Text `CONTAINS` operator with the SQL `WHERE` clause rule conditions.

Rules Manager can now compute aggregate values (`SUM`, `AVG`, `MIN`, `MAX`, and `COUNT`) using moving windows semantics. The window size can be based on time or number of events. The aggregate values for a group of events can be computed and tested against predefined or relative thresholds in rule conditions.

Rules can respond to real-world trends identified from a collection of events occurring over a particular length of time or a sliding window of time, such as for Business Activity Monitoring and Business Intelligence applications. For example, you can offer a promotion to a customer if the sum of purchases in the last three days is over $1000.00.

See Also:

Oracle Database Rules Manager and Expression Filter Developer's Guide for details

1.1.3.5 Oracle Text Predicates Included in Stored Expressions

The Expression Filter feature of Oracle Database now supports the use of the Oracle Text `CONTAINS` operator with the SQL `WHERE` clause stored expressions.

Stored expressions can evaluate text and XML data.

See Also:

Oracle Database Rules Manager and Expression Filter Developer's Guide for details

1.1.3.6 System-Managed Partitioning of Extensible Indexes

System-managed domain indexes (both non-partitioned and local partitioned) use Oracle Database to track and maintain the domain index storage tables during DDLs and partition maintenance operations on the base table. System-partitioned tables can be used as index storage tables for local domain indexes. System-managed domain indexes with range partitioning are supported in this release.

System-managed partitioning of domain indexes enhances local domain index manageability and performance.

See Also:

Oracle Database Data Cartridge Developer's Guide for details

1.1.4 Globalization and Unicode

This section lists improvements to globalization features and Unicode support of Oracle Database.

1.1.4.1 Unicode 5.0 Support

The NLS data files for AL32UTF8 and AL16UTF16 character sets have been updated to match version 5.0 of the Unicode Standard character database.

With this enhancement, Oracle Database conforms to the newest version of the standard.

See Also:

Oracle Database Globalization Support Guide for details

1.1.5 Java in the Database

The major enhancements to the Java runtime in this release include performance with a new JIT compiler, ease-of-use with a JDK-like command-line interface, and tooling.

1.1.5.1 OracleJVM Java/JDK 5.0 Compatibility

OracleJVM supports Sun's JDK 1.5.

This feature provides portability of J2SE applications (stand-alone JDBC or middle-tier) into Oracle Database and the ability to take advantage of the key benefits of J2SE 5.0 such as the "declarative" programming style enabled by Java Annotations.

This feature allows sharing metadata of Java classes loaded by user class loader among different database user sessions.

See Also:

Oracle Database Java Developer's Guide for details

1.1.5.2 OracleJVM JIT Configuration

The JIT is enabled by default (out-of-the-box) and does not need further configuration. However, it can be disabled or reenabled by setting `JAVA_JIT_ENABLED` to `TRUE` or `FALSE` in the parameter file or dynamically by altering the instance or the session, for example:

```
ALTER SYSTEM SET JAVA_JIT_ENABLED=true
or
ALTER SESSION SET JAVA_JIT_ENABLED=true
```

In addition, the `OJVM_JIT_PROPERTIES` are used to specify the list of properties, for example:

```
OJVM_JIT_PROPERTIES="key0-value0","key1=value1","keyword0"
```

Currently, supported properties are:

- `DISABLE_OPTIM_*` which are various switches to disable optimization passes.
- New dictionary table and Oracle Runtime methods for checking the compilation status of classes or methods.
- New column `IS_COMPILED` in `USER_JAVA_METHODS`, `ALL_JAVA_METHODS`, and `DBA_JAVA_METHODS` views set to `YES` or `NO`.

This feature allows fine-grained control on the behavior of the JIT. In addition, you have the ability to query the status of compiled classes and methods.

See Also:

Oracle Database Java Developer's Guide for details

1.1.5.3 OracleJVM JIT (Transparent Native Java Compilation Without a C Compiler)

Enabled by default and similar to the JDK JIT, this feature compiles Java in the database natively and transparently without the need of a C compiler.

The JIT runs as an independent session in a dedicated Oracle server process. There is at most one compiler session per database instance and it is Oracle RAC-aware and amortized over all Java sessions.

The compilation session uses profiling feedback data gathered by all Java sessions to compile the most often used methods, so as to get the best overall performance. The executable binary persists in the database.

Java sessions just use the executable binaries of already compiled methods or discover newly compiled methods through an event polling performed by OracleJVM at regular intervals. When there is a compiled code for a given method, future executions of that method use the compiled version.

Conversely, when a Java class is invalidated, the persisted compiled code of the class or method is removed.

This feature brings two major benefits to Java in the database:

- Increased performance of pure Java execution in the database resulting from the transparent compilation of Java into native machine code, similar in behavior to a JDK JIT.
- Ease-of-use. Unlike the Static Native Java compiler (also known as NCOMP or JAccelerator) in previous releases, the new JIT Compiler (when enabled at database configuration level) is activated transparently, without the need of an explicit command, when Java is executed in the database.

In addition, this feature removes the need for a C compiler.

See Also:

Oracle Database Java Developer's Guide for details

1.1.5.4 OracleJVM Ease-of-Use: JDK-Like Interface

Similar to the JDK VM, this feature provides a command-line interface to OracleJVM through new PL/SQL interfaces, for example:

```
FUNCTION runjava (cmdline VARCHAR2) RETURN VARCHAR2;
or
FUNCTION runjava_in_current_session (cmdline VARCHAR2) RETURN VARCHAR2;
```

Analogous to the JDK/JRE Java shell command, the JDK-like interface takes a subset of the JDK/JRE command-line syntax including:

```
[-options] classname [arguments...]
or
[-options] -jar jarfile [arguments...]
```

The supported options include:

- {cp|classpath} *< ":" separated list of file system directories/JAR/ZIP to search for class files>*
- D*<name>*=*<value>* set a system property
- Xbootclasspath:*<":" separated directories/zip/jar files to search for bootstrap classes and resources>*
- Xbootclasspath/a:*<":" separated directories/zip/jar files to append to the end of bootstrap class path>*
- Xbootclasspath/p:*<":" directories/zip/jar files to prepend in front of bootstrap class path>*

This feature makes Java easy to use using command-line syntax, which is similar to the JDK JAVA command line. In addition to database resident Java classes, this feature allows the execution in the database runtime environment of file system resident Java class, using the standard `classpath` syntax. In addition, it allows setting system properties using the standard `-D` syntax.

See Also:

Oracle Database Java Developer's Guide for details

1.1.5.5 OracleJVM Ease-of-Use: Property Management Interface

This feature provides `DBMS_JAVA` functions to emulate JDK System property setting for the duration of the session. The settings are `set_property`, `get_property`, `remove_property`, and `show_property`.

Java developers can now emulate familiar JDK behavior such as system properties support when using Java in the database.

1.1.5.6 OracleJVM Ease-of-Use: Output Redirect

This feature provides `DBMS_JAVA` functions and procedures for controlling the redirection of Java output, for example:

```
{set | remove | enable | disable | query}_output_to_{sql | java | file}
```

And for enabling or disabling sending output to trace, for example:

```
trace {enable | disable}_output_to_trc
```

Java developers can now emulate familiar JDK behavior such as output redirect. In addition, they can now disable sending output to a trace file, which was not possible in previous releases.

1.1.5.7 OracleJVM Ease of Use: Database Resident Jars

This feature allows supporting JAR as a unit of operations (and as a result of support for signed JARs) using new `loadjava` options, `-jarsasdbobjects` and `-prependjarnames`.

When these new options are not used, `loadjava` explodes the jar into individual classes, while `dropjava` goes back to the original jar (from which the classes are loaded) and drop the classes each at a time.

Java developers using OracleJVM benefit from the same JDK features and behavior such as JAR support.

1.1.5.8 OracleJVM Ease of Use: Two-Tier Java Session Termination

This feature allows the termination of Java within a database session, with and without retention of system properties, using `endsession` and `endsession_and_related_state`.

Java developers can terminate Java in a session while keeping the property settings so as to speed up Java reinitialization within the same database session. This feature also allows immediate termination of Java sessions, without requiring the use of `System.exit`.

1.1.5.9 OracleJVM Utilities Enhancements

Enhancements to existing OracleJVM utilities (`loadjava`, `dropjava`, and `ojvmjava`) include the deprecation of least-frequently-used options of `loadjava` and the ability to load Java classes over the web (HTTP URL support). For example:

```
loadjava -u scott/tiger -r
-v http://my.server.com/this/is/the/path/my.jar
```

Enhancements to `dropjava` include:

- list-based deletion
- exception handling
- ability to connect to a different user
- invalid command reporting

`ojvmjava` has been extended to support the JDK-like command such as `runjava` and its options. For example:

```
ojvmjava -u scott/tiger -runjava -d <machine-name-or-ip>:1521:<sid> -t
```

And, then followed with:

```
$java -cp .<class>
```

In addition, a new standalone Class Closure Tool (`ojvmtc`) is provided for ensuring that a given list of file system resident classes and deployable JARs to be closed as much as possible before loading (for example, `loadjava` invocation).

The benefits are increased usability and simplified deployment of Java applications in Oracle Database.

See Also:

Oracle Database Java Developer's Guide for details

1.1.5.10 OracleJVM Manageability: JMX Interface

This feature provides support for OracleJVM with the standard JMX interface.

Support for JMX interface provides:

- Ease-of-use (any standard JMX-enabled GUI or console)
- Runtime monitoring of OracleJVM (memory objects heaps, threads, GC and classes)
- Ability to trigger GC from the console

1.1.6 JDBC and SQLJ

For Java developers, this release provides the following enhancements to JDBC and SQLJ APIs:

- Support for the latest Java standards including Java SE 1.5 and JDBC 4.0.
- A faster protocol for the Streams/AQ Java interface.
- Support for Oracle advanced security in JDBC-Thin.
- Enhancements to SQLJ code generation and customizer.
- Enhancements to XA support.

- Support of event notification.

1.1.6.1 JDBC 4.0 Connection and Statement Enhancements

JDBC 4.0 introduces new methods for:

- Validating a connection.
- Setting and fetching client information on a connection.
- Returning a list of client properties supported by the driver.
- Allowing an application to mark that a statement should not be pooled.
- Allowing a statement pool manager to register a `StatementEventListener` on the pooled connection.

In addition, it provides a new `StatementEventListener` interface and a new `StatementEvent` class.

These JDBC 4.0 enhancements improve the scalability of JDBC applications and provide better control and management of statement objects.

See Also:

Oracle Database JDBC Developer's Guide and Reference for details

1.1.6.2 JDBC 4.0 Support for New Data Types

This feature provides new standard types including `java.sql.XMLType`, `java.sql.RowID`, and `java.sql.Nclob` to support the National Character Set (`NCHAR`, `NVARCHAR`, `LONGVARCHAR`). In addition, the JDBC 4.0 implementation enhances `BLOB` and `CLOB` data type manipulation.

Standard support for new SQL types significantly increases the portability of JDBC applications across databases.

See Also:

Oracle Database JDBC Developer's Guide and Reference for details

1.1.6.3 JDBC Support for ANYTYPE, ANYDATA, and ANYDATASET

The following support has been added to JDBC in this release:

- `ANYTYPE` which represents any SQL type including persistent database scalar or object types, and transient unnamed types.
- `ANYDATA` as a table column type to store heterogeneous values (built-in and user-defined) within the same column, along with the metadata.
- `ANYDATASET` which represents a set of data instances of the same type, along with the metadata.

The benefits include:

- Mapping any Java types to a general purpose SQL data type (`ANYTYPE`).
- Persisting Java objects to the same database table column (`ANYDATA`).
- Mapping any Java collections into `ANYDATASET`.

See Also:

Oracle Database JDBC Developer's Guide and Reference for details

1.1.6.4 JDBC Support for Continuous Query Notification

This feature allows Java Applications or Containers to subscribe to SQL queries watch list in the database and be notified when:

- DML or DDL changes affect the objects associated with the queries, or
- DML or DDL changes affect the result set

The notifications are published when the transaction (DML or DDL) commits.

Java Applications and Container prevents unnecessary database querying and implements efficient and consistent result set caching and cache invalidation.

See Also:

Oracle Database JDBC Developer's Guide and Reference for details

1.1.6.5 JDBC Support for Database Diagnosability

Through this feature, JDBC provides better support for logging, diagnosability and DMS monitoring:

- Logging has been reimplemented to fully utilize `java.util.logging`. This allows you to selectively log events that occur during the execution of the JDBC driver code such as SQL exceptions, SQL statement executions, or entry or exit from internal JDBC methods.
- JDBC support for performance monitoring uses DMS to capture performance-critical JDBC events including:
  - Connection cache statistics
  - Connection creation
  - Statement cache statistics
  - Query execution
  - Result fetch
  - Oracle RAC events
  - Load balancing statistics
  - New Oracle Database 11*g* features
- JDBC support for Diagnosability Management consists of an MBean, `oracle.jdbc.DiagnosabilityMBean`. This MBean provides the means to enable and disable JDBC logging and provides additional statistics about JDBC internals.

This feature simplifies JDBC logging and makes the interpretation of the results easy. Dynamic Monitoring System (DMS) metrics give Java applications the ability to track JDBC events and performance metrics, as well as a programmatic control and configuration of diagnosability features.

See Also:

Oracle Database JDBC Developer's Guide and Reference for details

1.1.6.6 JDBC Support for Database Startup and Shutdown

You can now perform database startup or shutdown using the following JDBC Oracle Connection methods:

- `DBSTARTUP_NO_RESTRICTION`

- DBSTARTUP_FORCE
- DBSTARTUP_RESTRICT
- DBSHUTDOWN_CONNECT
- DBSHUTDOWN_TRANSACTIONAL
- DBSHUTDOWN_TRANSACTIONAL_LOCAL
- DBSHUTDOWN_IMMEDIATE
- DBSHUTDOWN_ABORT
- DBSHUTDOWN_FINAL

Note that you must be connected to a dedicated server as SYSOPER or SYSDBA in PRELIM_AUTH mode.

See Also:

Oracle Database JDBC Developer's Guide and Reference for details

1.1.6.7 JDBC Support for Native Streams AQ Protocol

This feature exposes a new Java package (oracle.jdbc.aq) that leverages the new native JDBC APIs for Streams AQ. This package contains the following classes and interfaces:

- AQQueue
- AQMessage
- AQAgent
- AQDequeueOptions
- AQEnqueueOptions
- AQMessageProperties
- AQRawPayload
- AQObjectPayload
- AQNotificationEvent
- AQNotificationListener
- AQNotificationRegistration

Java and JMS applications run faster (using Native AQ operations as opposed to PL/SQL procedures) when accessing Streams AQ and OJMS queues. In addition, these APIs allow asynchronous notifications of AQ events and exchange rich message payloads including RAW data type, ANYDATA data type, and OBJECT data type.

See Also:

Oracle Database JDBC Developer's Guide and Reference for details

1.1.6.8 JDBC-Thin Support for Oracle Advanced Security

This feature provides complete JDBC-Thin support for Oracle Advanced Security in terms of encryption, password authentication, strong authentication, and data integrity or checksum.

New security features include Secure Sockets Layer (SSL), Remote Authentication Dial-In User Service (RADIUS) and Kerberos authentication, AES encryption, and

SHA1 checksum algorithm. Advanced security support in JDBC-Thin simplifies the development and deployment of secure Java applications.

See Also:

Oracle Database JDBC Developer's Guide and Reference for details

1.1.6.9 SQLJ: Enhanced Profile Print Option of the Customizer

SQLJ customizer is used during ISO standard code generation. The customizer tailors SQLJ profiles for a particular database implementation and vendor-specific features and data types. The profile print option (`-P-print`) of SQLJ has been enhanced to print all the customizer options that were passed into the SQLJ translator.

This feature provides information about customizations that SQLJ has received during the precompilation phase to help diagnose any problems that may arise and are specific to different customizations.

See Also:

Oracle Database SQLJ Developer's Guide and Reference for details

1.1.6.10 SQLJ: Outline Generation Option

An application can support SQL execution plan stability by using the outline feature of Oracle. An outline is implemented as a set of optimizer hints that are associated with the SQL statement. If the use of the outline is enabled for the statement, Oracle automatically considers the stored hints and tries to generate an execution plan in accordance with those hints.

A new translation time option is provided in SQLJ to generate and run a SQL file containing the Oracle outline creation commands for all the SQL statements present in a SQLJ application.

This feature helps in ensuring the expected behavior of a database query at development time and helps to better diagnose problems.

See Also:

Oracle Database SQLJ Developer's Guide and Reference for details

1.1.6.11 SQLJ: Support for JDK 1.5

SQLJ applications can now be translated and run under JDK 1.5.

The JDK 1.4 and JDK 1.5 specific language structures outside the #sql region are recognized by the translator.

See Also:

Oracle Database SQLJ Developer's Guide and Reference for details

1.1.6.12 SQLJ: XA Support

The `javax.sql.XADataSource` interface of JDBC outlines standard functionality of XA data sources. An XA data source is a factory for XA connections. Oracle JDBC implements the `XADataSource` interface through the `OracleXADataSource` class. The `getConnection()` method of the `OracleXADataSource` class returns an XA connection to the underlying data source.

The benefit of this feature is that these connections can be used in SQLJ applications by creating the appropriate connection contexts in SQLJ.

See Also:

Oracle Database SQLJ Developer's Guide and Reference for details

1.1.7 Oracle Developer Tools for Visual Studio .NET

These features enhance developer productivity for building .NET applications with Oracle Database. They provide native integration with ADO.NET 2.0 functionality using Visual Studio development environments with which .NET developers are familiar. In addition, these features reduce the time it takes to develop .NET applications for Oracle.

1.1.7.1 General Visual Studio 2005 Support

This feature provides users of Visual Studio 2005 the same support currently offered for Oracle Develop Tools (ODT) users using Visual Studio 2003. The changes include using different icons and integrating visually with the Visual Studio 2005 look and feel.

This feature makes it easy for customers to continue using ODT when they migrate from Visual Studio 2003 to the Visual Studio 2005 environment.

1.1.7.2 Oracle PL/SQL Debugger in Visual Studio

The Oracle PL/SQL debugger is now integrated directly into Visual Studio. Developers can debug code, set breakpoints, and watch variables exactly as they would for any other language. Three types of debugging are supported:

- Direct Database Debugging

  The developer runs the stored procedure from the Oracle Explorer inside of Visual Studio and enters in the parameters.

- Multi-Tier Application Debugging

  The developer steps from .NET code directly into PL/SQL and back again with the parameters automatically being passed.

- External Application Debugging

  The developer runs any 9.2 or later Oracle application that can be located anywhere (for example, from SQL*Plus). When it calls the stored procedure, the break point in Visual Studio fires and the developer can begin debugging with the stored procedure parameters being passed automatically.

This feature enables developers to seamlessly debug .NET applications that make calls to PL/SQL stored procedures. Developers can debug both .NET code and PL/SQL without having to leave Visual Studio. They can also debug PL/SQL stored procedures that are called from any arbitrary Oracle program, written in any language, and located on the same machine or on a remote machine.

1.1.7.3 Support for Visual Studio 2005 Data Sources Window and Designers

This feature exposes the DDEX interface to enable some Visual Studio 2005 designers as well as integrate with the Data Sources window to enable the new style drag-drop code generation in Visual Studio 2005.

1.1.8 Oracle Data Provider for .NET (ODP.NET)

These features enhance developer productivity for building .NET applications with Oracle Database. They provide native integration with ADO.NET 2.0 functionality

using Oracle Data Provider for .NET (ODP.NET) development environments with which .NET developers are familiar. In addition, these features reduce the time it takes to develop .NET applications for Oracle.

1.1.8.1 Provider-Independent API Using Base Classes and Factories

ODP.NET inherits from the `System.Data.Common` namespace to provide access through a single API to databases across multiple providers.

ADO.NET developers can build applications that work with multiple vendors' database servers with one set of data access code and with a minimum of vendor-specific code. Developers can now be more productive when building one set of code that works against multiple vendors' databases.

See Also:

Oracle Data Provider for .NET Developer's Guide for details

1.1.8.2 Connection String Builder

ODP.NET provides a strongly typed connection string builder class that inherits from `DbConnectionStringBuilder`. The connection string builder lets developers programmatically create syntactically correct connection strings based on user input, as well as parse and rebuild existing connection strings by using class methods. Connection string builders provide strongly-typed properties that correspond to known key/values pairs.

ODP.NET connection strings are now more secure and more manageable. Because connection string parameters can now be strongly typed, they are less subject to string injection attacks. Connection string parameters can now be referenced in configuration files or as ODP.NET properties, making managing connection information easier.

See Also:

Oracle Data Provider for .NET Developer's Guide for details

1.1.8.3 OracleDataAdapter Batch Processing

During updates to Oracle Database, the ODP.NET `OracleDataAdapter` can batch a set of rows back to the database per round trip. The number of rows per round trip can be set by the developer.

This feature enhances application performance by reducing the number of round trips to the database when applying updates from the DataSet.

See Also:

Oracle Data Provider for .NET Developer's Guide for details

1.1.8.4 Row-Level Change Notifications

.NET developers often used client-side cached data. When another user changes the original data on the server, the database sends a change notification message to the client, indicating that the client data is now invalid. Previously, change notifications were registered at the table level. In this release, developers can register their queries at the row level so that they are alerted only when the rows they selected have changed.

.NET clients can now receive more accurate change notification alerts.

See Also:

Oracle Data Provider for .NET Developer's Guide for details

1.1.8.5 Schema Discovery

ODP.NET schema discovery allows applications to request and return information about the database schema to the client. Different database schema elements such as tables, columns, and stored procedures are exposed through the `GetSchema` methods of the `OracleConnection` class.

ODP.NET can retrieve a set of generic and Oracle-specific schema information.

Schema discovery supplies a common API for all ADO.NET 2.0 drivers to retrieve database schema information to the client application.

See Also:

Oracle Data Provider for .NET Developer's Guide for details

1.1.8.6 Data Source Enumeration

The ODP.NET `OracleDataSourceEnumerator` class provides a list of active Oracle database servers.

Data source enumeration can be retrieved programmatically in a generic manner common to all ADO.NET 2.0 providers.

See Also:

Oracle Data Provider for .NET Developer's Guide for details

1.1.8.7 Faster LOB Retrieval

ODP.NET now makes fewer round trips to the database server to retrieve LOB information.

ODP.NET LOB retrieval performance is now faster.

See Also:

Oracle Data Provider for .NET Developer's Guide for details

1.1.8.8 Improved Parameter Context Caching in Statement Caching

For statement caching, ODP.NET now caches parameter contexts after the first statement execution.

Many applications execute the same SQL or PL/SQL statement repeatedly. In the first execution of the statement, ODP.NET can now cache parameter contexts so that they can be reused. This improves the performance of subsequent statement executions.

See Also:

Oracle Data Provider for .NET Developer's Guide for details

1.1.9 Oracle Provider for OLE DB

These features improve data access performance from Windows clients to Oracle Database. Many of these features are transparent, meaning applications experience faster performance by upgrading the database server and client without any code changes.

1.1.9.1 Faster Performance

Oracle Provider for OLE DB has implemented a number of new features in its provider to improve query performance for the end user.

See Also:

Oracle Provider for OLE DB Developer's Guide for details

1.1.9.2 Improved Metadata Caching

For metadata caching, Oracle Provider for OLE DB now caches the metadata after the first execution when statement caching is turned off. This is useful when a large amount of column metadata is retrieved with a small amount of row data. Oracle Provider for OLE DB has also improved metadata caching when statement caching is turned on.

Many applications execute the same SQL or PL/SQL statement repeatedly. In the first execution of the statement, Oracle Provider for OLE DB can cache a query's metadata so that it can be reused. This improves the performance of subsequent statement executions.

See Also:

Oracle Provider for OLE DB Developer's Guide for details

1.1.9.3 Improved Statement Caching

For statement caching, Oracle Provider for OLE DB now caches large fetch arrays and OCI error handles after the first execution.

Many applications execute the same SQL or PL/SQL statement repeatedly. In the first execution of the statement, Oracle Provider for OLE DB can now cache a statement's parse information so that it can be reused. This improves the performance of subsequent statement executions.

See Also:

Oracle Provider for OLE DB Developer's Guide for details

1.1.10 PHP

The features in the following sections describe the enhancements to Oracle's client-side stack so that data access open source PHP driver performs better than PHP data access drivers for other databases.

1.1.10.1 Database Resident Connection Pooling (DRCP)

Prior to this release, session sharing capabilities were available for multithreaded applications only. With this new feature, different application processes can share sessions within the same host machine as well as across multiple host machines. The pool is maintained on the database server by a new background process.

This feature is useful for applications that cannot be deployed as multithreaded (for example, PHP applications in an Apache Web server environment) and database requests are done in a session state independent manner. These applications no longer have to hold the connections to the database persistently, thereby increasing the scalability of the database.

The pool is managed by the DBA and the OCI and OCCI applications can reuse the sessions by specifying a connection class name within which the application leaves sessions of similar state.

See Also:

Oracle Call Interface Programmer's Guide for details

1.1.11 PL/SQL

The following sections discuss features that:

- Reduce downtime associated with application patching
- Enhance performance and scalability (TPCC, APPS)
- Provide integration between PL/SQL and SQL

1.1.11.1 XA API Available Within PL/SQL

The functionality available in the XA interface to support transactions involving multiple resource managers, such as databases and queues, is now available from PL/SQL.

Application developers are now able to switch or share transactions across SQL*Plus sessions or processes using PL/SQL.

See Also:

Oracle Database Advanced Application Developer's Guide for details

1.1.11.2 Allow Sequences in PL/SQL Expressions

In the previous release of Oracle Database, when a PL/SQL program needed to get a value from syntax, it used SQL. For example:

```
DECLARE n NUMBER;
BEGIN
SELECT Seq.Nextval INTO n FROM Dual;
END;
```

This posed a usability problem for PL/SQL programmers and causes runtime performance and scalability problems for the application particularly because it requires the use of a cursor.

In Oracle Database 11*g*, it is now possible to simply use `Seq.Nextval` in a PL/SQL expression. For example:

```
DECLARE n NUMBER := Seq.Nextval;
BEGIN
...
```

The same is possible for the `CURRVAL` pseudocolumn.

This feature brings improved usability for the PL/SQL programmer and improved runtime performance and scalability.

See Also:

Oracle Database PL/SQL Language Reference for details

1.1.11.3 Enhancements to the Regular Expression Built-Ins

There is new syntax to access the *n*th subexpression in `REGEXP_SUBSTR` and `REGEXP_INSTR`. This is useful if, for example, the matching string is always surrounded by tags like *<string>* and *</string>* but you want to extract just what lies between these

tags. REGEXP_SUBSTR returns the subexpression. REGEXP_INSTR returns the position.

The new built-in REGEXP_COUNT returns the number of times the pattern is matched in the input string.

This new functionality is available both in SQL and in PL/SQL.

The benefit is increased functionality for the regular expression built-ins. Earlier applications that needed this functionality had to write relatively elaborate application code which is no longer the case.

See Also:

Oracle Database Advanced Application Developer's Guide for details

1.1.11.4 Functional Completeness of Dynamic SQL for PL/SQL

In Oracle Database 11*g*, native dynamic SQL now supports statements bigger than 32K characters by allowing a CLOB argument. Also included with this feature:

- DBMS_SQL.PARSE() gains a CLOB overload
- A REF CURSOR can be converted to a DBMS_SQL cursor and vice versa to support interoperability
- DBMS_SQL supports the full range of data types (including collections and object types)
- DBMS_SQL allows bulk binds using user-define collection types

The features for executing dynamic SQL from PL/SQL had some restrictions in Oracle Database 10*g*. DBMS_SQL was needed for Method 4 scenarios but it could not handle the full range of data types and its cursor representation was not usable by a client to the database. Native dynamic SQL was more convenient for non-Method 4 scenarios, but it did not support statements bigger than 32K. Oracle Database 11*g* removes these and other restrictions to make the support of dynamic SQL from PL/SQL functionally complete.

See Also:

Oracle Database PL/SQL Language Reference for details

1.1.11.5 Method Invocation Scoping Operator

A subtype usually adds specific data attributes to its more generic supertype. Typically, a NOT FINAL member procedure, for example, Show_Attributes(), is implemented at each level in the type hierarchy. It is natural to implement this at one level by reusing the implementation at the parent level and by adding new code just for the attributes of the subtype.

Oracle Database 11*g* supports the ANSI SQL generalized-invocation syntax to allow the following:

```
(<type-instance> AS <type-name>).<method-name>.
```

Without this syntax, programmers were forced to use a cumbersome workaround which relied on using static subprograms rather than member subprograms.

This feature brings the implementation of object-oriented programming in Oracle Database closer to completeness.

See Also:

Oracle Database Object-Relational Developer's Guide for details

1.1.11.6 Named and Mixed Notation for User-Defined PL/SQL Subprogram in a SQL Statement

Consider a schema-level function `f()` declared as:

```
FUNCTION f(
p1 IN INTEGER := 1,
p2 IN INTEGER := 2,
...
pn IN INTEGER := 99)
RETURN INTEGER
```

Beginning in this release, it is now possible to invoke the function in a SQL statement. For example, named notation syntax is:

```
SELECT f(pn=>3, p2=>2, p1=>1) FROM dual
```

Or, mixed notation is:

```
SELECT f(1, pn=>3) FROM dual
```

In previous releases, attempting named or mixed notation resulted in an error.

The benefit is improved usability when a PL/SQL function, with many default parameters, is invoked in a SQL statement and only one or a small number of the actual parameters need to differ from their default values.

See Also:

Oracle Database PL/SQL Language Reference for details

1.1.11.7 New Compound Trigger Type

A compound trigger has a section for each of the `BEFORE STATEMENT`, `BEFORE EACH ROW`, `AFTER EACH ROW`, and `AFTER STATEMENT` timing points. All of these sections can access a common PL/SQL state. The common state is created when the firing statement starts and destroyed when it completes, even when the firing statement causes an error. A compound trigger can be used to speed performance in an auditing scenario where audit rows can be accumulated in the EACH ROW section and bulk inserted to the audit table in the `AFTER STATEMENT` section. A similar approach is sometimes required to avoid the mutating table error. The alternative to using compound triggers is to model the common state with an ancillary package. This approach is both cumbersome to program and subject to memory leak when the triggering statement causes an error and the `AFTER STATEMENT` trigger does not fire.

This feature brings improved usability for the PL/SQL programmer and improved runtime performance and scalability.

See Also:

Oracle Database PL/SQL Language Reference for details

1.1.11.8 PL/Scope

PL/Scope allows you to browse PL/SQL source code analogously to the way that Cscope (see `http://cscope.sourceforge.net/`) allows you to browse C source code. You can search for and display all types of definitions, declarations, assignments and references in the PL/SQL source code.

The PL/SQL compiler can optionally derive the metadata needed to support PL/Scope and store it in the database catalog. The metadata takes into account the nuances of the language, including scoping and overloading. You can generate reports (especially hyperlinked HTML reports) with supplied report generators. This feature is exposed through Oracle SQL Developer's interactive PL/SQL development environment.

PL/Scope supports increased developer productivity, especially for those who need to maintain someone else's code.

See Also:

Oracle Database Advanced Application Developer's Guide for details

1.1.11.9 PL/SQL CONTINUE Statement

Many programming languages allow transfer of control either out of the loop entirely or within a loop back to a new iteration. In previous releases, PL/SQL only allowed transfer of control out of the loop using the `EXIT` statement. In Oracle Database 11*g* Release 1 (11.1), the new `CONTINUE` statement implements the ability to transfer control within a loop back to a new iteration.

Many algorithms require the loop control that this new `CONTINUE` statement now provides. With this explicit feature, you are no longer required to code several lines of PL/SQL and maintain commonly-adopted style rules.

See Also:

Oracle Database PL/SQL Language Reference for details

1.1.11.10 PL/SQL Hierarchical Profiler

PL/SQL Hierarchical Profiler identifies hot spots and performance tuning opportunities in PL/SQL database applications. It reports the dynamic execution program profile organized by subprogram calls. SQL and PL/SQL execution times are accounted for separately. PL/SQL Hierarchical Profiler requires no special source or compile time preparation.

Start and stop procedures in the new package, `DBMS_HPROF`, control the recording of raw PL/SQL Hierarchical Profiler data. Then other subprograms in this package are used to transform the raw data into schema-level tables to allow the generation of reports. For example, hyperlinked XML or HTML reports.

This feature is part of Oracle SQL Developer's interactive PL/SQL development environment.

Programmer productivity and program performance are both enhanced because PL/SQL Hierarchical Profiler directs tuning effort towards those PL/SQL functions that will benefit most.

See Also:

Oracle Database Advanced Application Developer's Guide for details

1.1.11.11 PL/SQL Inlining Optimization

Modern PL/SQL software includes many procedures frequently called in various programs. The power of the PL/SQL optimizing compiler is increased by incorporating the inlining optimization (replacement of a procedure call with a copy of the procedure body). The revised code executes more quickly because call overhead

has been eliminated and because further optimizations are possible when the compiler knows the actual values with which the inlined procedure is called.

Performance gains can be substantial. The compiler does this optimization automatically. However, you may control the optimization in detail when necessary. Many PL/SQL programs execute faster, as a direct consequence of upgrade, simply by recompiling existing units.

See Also:

Oracle Database PL/SQL Language Reference for details

1.1.11.12 SIMPLE_INTEGER Data Type

The new PL/SQL `SIMPLE_INTEGER` data type is a binary integer for use with native compilation which is neither null checked nor overflow checked. The range is all integer values in the inclusive interval [-2147483648..2147483647] and does not include a null value. These rules allow a direct hardware implementation when used in combination with PL/SQL native compilation.

The `SIMPLE_INTEGER` data type yields significant performance benefits compared to `PLS_INTEGER`. It can be used when the value will never be null and when overflow check is not needed.

See Also:

Oracle Database PL/SQL Language Reference for details

1.1.12 XML Application Development

The features described in the following sections ensure that PL/SQL, Java, C and .NET programmers can leverage the full capabilities of Oracle XML DB and the Oracle XML DB repository. These new features also extend the infrastructure components to support Binary XML and provides an efficient method for handling very large XML documents. There is also support relevant to JSRs for XML application development including JSR-170 for content repository access and the emerging JSR-225 standard for XQuery. The Oracle XML libraries continue to deliver the best possible application development framework for XML developers working in J2EE and .NET environments, regardless of whether they are using the Oracle XML DB feature. These new features deliver highly optimized XSLT and XQuery engines for use with Oracle Application Server and Fusion Middleware.

1.1.12.1 Binary XML Support for Java

The new XDK binary XML processors include a binary XML encoder, a binary XML decoder, and a token manager to convert the schema or nonschema-based binary XML stream to and from XML 1.0 text and `SAX` events. The format is the same as the format used within Oracle XML DB.

This feature allows XML processing applications to leverage a compact, pre-parsed XML format which could save in memory and CPU consumption as well as network bandwidth. With a single format supported by Oracle XML DB and Oracle XDK and the Oracle XML DB support within the binary XML processors, this feature allows the performance optimization across tier XML applications.

See Also:

Oracle XML Developer's Kit Programmer's Guide for details

1.1.12.2 Binary XML Support for Oracle XML DB

No single approach to persisting XML can solve all of the possible use cases presented by the XML application development paradigm. Binary XML (which is a way of representing the structure of an XML document) adds a third storage model for persisting XML in the database. It complements, rather than replaces, the existing object storage and text storage models.

Binary XML has two significant benefits:

- It enables significant optimizations of XML operations whether or not an XML schema is available.
- It allows the internal representation of XML to be the same regardless of whether the XML is in disk, in memory, or on the wire.

As with other storage mechanisms, the application developer can continue to use `XMLType` and its associated methods and operators without regard to the details of storage.

See Also:

Oracle XML DB Developer's Guide for details

1.1.12.3 Configurable DOM Support for Java

The following options are provided for building Document Object Model (DOM):

- `static final String ACCESS_MODE = "DOM_ACCESS_MODE";`
- `static final String READ_ONLY = "READ_ONLY";`
- `static final String UPDATEABLE = "UPDATEABLE";`
- `static final String FORWARD_READ = "FORWARD_READ";`
- `static final String STREAMING = "STREAM_READ";`

This feature enables XML applications to achieve maximum efficiency in terms of memory usage and performance by configuring the DOM building with different access patterns such as read-only, forward read, or streaming. This feature prevents light-weight XML applications from having to deal with a heavy DOM object tree.

See Also:

Oracle XML Developer's Kit Programmer's Guide for details

1.1.12.4 Event-Based XML Pull Parsing for C

XML pull parsing supports the stream-based parsing model which limits memory footprints by not preserving any document states in memory. Compared with `SAX`, the new pull parsing model only parses events requested by the application and accepts multiple input XML documents.

This feature improves the C XML processing performance and ensures the scalability with the new stream-based XML pull parsing. This feature also limits memory footprints by not preserving any document states in memory and improves the performance of the XML processing by only parsing stream events requested by the application. The pull parsing also makes the XML processing easy-to-use by allowing one parsing process to handle multiple XML documents.

See Also:

Oracle XML Developer's Kit Programmer's Guide for details

1.1.12.5 High Performance XPath Query for Java

Support JAXP 1.3 APIs for query XML content using XPath with the following extensions:

- Supports variable resolver in both static and dynamic context.
- Allows users to register runtime context.

This feature improves the performance and the usability of Java XPath/XSLT implementation.

1.1.12.6 In-Place XML Schema Evolution

This feature lets users make changes to a registered XML schema without having to unload and reload all of the XML documents that are compliant with the XML schema. Existing documents are not modified during in-place evolution, so only changes that do not invalidate the existing corpus of XML documents are permitted.

This feature enables rapid responses to changes in requirements by allowing certain common changes to XML schemas be made without having to unload and reload data. Organizations can take advantage of the benefits of XML schema-based XML storage in environments where the dynamic nature of the application or the large volume of XML content might make unloading and reloading the entire XML corpus to accommodate a change in the XML schema unrealistic.

The feature greatly reduces the overhead involved in making changes to XML schema in the majority of cases. Evolving an XML schema in cases where there are large number of documents compliant with the XML schema is a very expensive process that can take days to complete. In-place XML schema evolution allows the majority of common changes to be made almost instantaneously, with zero downtime, resulting in significant cost savings for the organization.

See Also:

Oracle XML DB Developer's Guide for details

1.1.12.7 Mid-Tier XQuery Engine Performance Improvements

XQuery performance improvements have been made in this release.

XQuery performance improvements for the mid-tier speed up applications relying on the mid-tier XQuery services.

1.1.12.8 Pluggable DOM Support for Java

The pluggable DOM splits the DOM implementation into two separate layers, the DOM API layer and the data layer. The API layer consists of a lightweight, transient node fragment that links to underlying data, which can be backed by external storage for scalability. The data has two varieties, internal data and plug-in data. The plug-in data support is through the InfosetReader implementations.

This feature allows XML applications to build DOM based on different forms of XML data through a unified interface. This feature improves portability of the XML applications.

See Also:

Oracle XML Developer's Kit Programmer's Guide for details

1.1.12.9 Scalable and High Performance XML Validation for C

The stream-based XML Schema validation allows XML applications to validate XML content against XML schemas in XML pull parsing process.

High performance XML processing and its scalability are very critical for enterprise XML applications. This feature improves the C XML processing performance and ensures the scalability with the new stream-based XML Schema validation.

See Also:

Oracle XML Developer's Kit Programmer's Guide for details

1.1.12.10 Scalable DOM Support for Java

This feature ensures the scalability of Database Object Model (DOM) processing for XML applications with the following support:

- Allows XML applications to process large XML documents by loading and saving DOM nodes in memory and using a page manager for physical binary data management.
- Allows storing DOM updates with references to the source data in Binary XML.
- Allows multiple applications sharing same DOM data source and operating on a lightweight, transient node fragment that links to underlying data, which can be backed by external storage for scalability.
- Supports Binary XML as the data source or the output.
- Supports shadow copy in DOM (`cloneNode() or importNode()`) which does not require the whole content being deep copied.

Scalability is the key to the success of the enterprise XML applications. This feature ensures the scalability of DOM processing for XML applications.

See Also:

Oracle XML Developer's Kit Programmer's Guide for details

1.1.12.11 Unified Java API for XML

The new unified Java API for XML allows mid-tier Java programs to leverage all of the benefits of `XMLType` by providing a disconnected mode of operation that allows `XMLType` to be used with a session pool model of connection management.

This allows the `XMLType` object to be disconnected from the database session that was used to create it.

This feature improves Java programmer productivity by allowing a single API to be used to manipulate XML content stored in Oracle XML DB and XML content stored in other sources.

Unified Java API also provides a disconnection mode of operation that allows `XMLType` to be used in a mid-tier environment where session pooling makes it impractical to hold the database connection that was used to instantiate the `XMLType`.

1.1.12.12 Unified Java API for XML

This feature allows mid-tier Java programs to leverage all of the benefits of `XMLType` by providing a disconnected mode of operation that allows `XMLType` to be used with a session pool model of connection management. This allows the `XMLType` object to be disconnected from the database session that was used to create it.

This feature improves Java programming productivity by allowing a single API to be used to manipulate the XML content stored in Oracle XML DB and the XML content stored in other sources.

This feature provides a disconnection mode of operation that allows `XMLType` to be used in a mid-tier environment where session pooling makes it impractical to hold the database connection that was used to instantiate the `XMLType`.

See Also:

Oracle XML Developer's Kit Programmer's Guide for details

1.1.12.13 XMLDiff Support for C

The `XMLDiff` detects the differences between the two XML documents and represents the difference in XML. The `XMLPatch` takes the differences created by `XMLDiff` and applies the changes on the target XML document.

This feature allows high performance XML document comparison and user controlled updates. With the support for detecting differences between two XML documents, XML applications can easily control document versions, merge and synchronize the content. The feature also reduces the processor demand by handling only the changed data, reduces bandwidth demands by transmitting only updates, and increases security by splitting sensitive data into separate streams which can then be reassembled on delivery.

See Also:

Oracle XML Developer's Kit Programmer's Guide for details

1.1.12.14 XMLIndex Enhancements

This feature provides an asynchronous mode for `XMLIndex`, where the indexing operations take place separately from the `INSERT` or `UPDATE` operations. Applications no longer have to wait for indexing to complete (which, for large documents, can be a significant operation) before being able to complete a transaction.

This feature also provides the ability to define the set of XPath expressions that are indexed by a given `XMLIndex`. The subset can be defined using `INCLUDE` or `EXCLUDE` syntax.

Subsetting XPath expressions that are indexed allows organizations to balance index size with disk usage, and to balance performance on insert with flexibility on retrieval.

This feature provides more flexibility for the organizations using `XMLIndex`. `XMLIndex` can be used in a wide range of applications. The feature improves application performance by allowing indexing to take place separately from `INSERT` and `UPDATE` operations. It also allows organizations to balance performance, flexibility, and disk usage when using `XMLIndex`.

See Also:

Oracle XML DB Developer's Guide for details

1.2 Availability

The following sections describe Availability features for Oracle Database 11*g* Release 1 (11.1).

1.2.1 Availability Interfaces in Oracle Enterprise Manager

The following sections describe improvements to the Enterprise Manager interfaces for high availability features.

1.2.1.1 Browser-Based Enterprise Manager Integrated Interface for LogMiner

This feature now makes it possible to use the browser-based Oracle Enterprise Manager Database Control interface for LogMiner. In prior releases, administrators were required to use the standalone Java Console to use LogMiner. The Console was not integrated with the rest of Enterprise Manager and was cumbersome to install. With this new interface, administrators have a task-based, intuitive approach to using LogMiner.

This new feature improves the manageability of LogMiner. The task-based work flows enable log mining and are integrated with Flashback Transactions.

1.2.2 Data Guard Improvements

The features in the following sections describe enhancements to Oracle Data Guard.

1.2.2.1 Fast-Start Failover for Maximum Performance Mode in a Data Guard Configuration

This feature enables fast-start failover to be used in a Data Guard configuration that is set up in the maximum performance protection mode. Since there is some possibility of data loss when a Data Guard failover occurs in maximum performance mode, administrators can now choose not to do a fast-start failover if the redo loss exposure exceeds a certain amount.

This enhancement allows a larger number of disaster recovery configurations to take advantage of Data Guard's automatic failover feature.

See Also:

Oracle Data Guard Broker for details

1.2.2.2 Compression of Redo Traffic (Only for Gap Resolution) Over the Network in a Data Guard Configuration

Archived redo logs, if needed to be transmitted to the standby database in an Oracle Data Guard configuration, can now be compressed as they are sent over the network.

This feature improves the network bandwidth utilization in an Oracle Data Guard configuration.

See Also:

Oracle Data Guard Concepts and Administration for details

1.2.2.3 Real-Time Query Capability of Physical Standby Database

It is now possible to query a physical standby database while Redo Apply is active.

This new capability increases your return on investment in Data Guard technology because a physical standby database can now be used to offload queries from the primary database in addition to providing data protection.

See Also:

Oracle Data Guard Concepts and Administration for details

1.2.2.4 Fast Role Transitions in a Data Guard Configuration

This feature improves the performance of Oracle Data Guard role transitions.

The benefit is increased flexibility and manageability of your disaster recovery configurations using Oracle Data Guard.

1.2.2.5 User Configurable Conditions to Initiate Fast-Start Failover in a Data Guard Configuration

This feature enables the administrator to choose and configure a list of conditions which, if they occur, will initiate fast-start failover.

This feature increases the flexibility and manageability of customers' disaster recovery configurations.

See Also:

Oracle Data Guard Broker for details

1.2.3 Data Guard Integration, Simplification, and Performance

The new features in the following sections simplify the configuration and use of Oracle Data Guard. For example, some features provide a smaller set of integrated parameters, a unified SQL/Broker syntax, and better integration with other High Availability features like RMAN and Oracle RAC. Other features enhance the performance of key Oracle Data Guard features like redo transport, gap resolution, switchover/failover times, and Logical Standby Apply.

1.2.3.1 Dynamic Setting of Oracle Data Guard SQL Apply Parameters

It is now possible to configure specific SQL Apply parameters without requiring SQL Apply to be restarted.

This ability improves the manageability, uptime, and automation of a Logical Standby configuration.

1.2.3.2 Enhanced Data Guard Broker Based Management Framework

The enhancements for this release include:

- Data Guard Broker improved logging and tracing
- OMF support for Data Guard Broker configuration files
- Data Guard Broker integration with database startup
- Data Guard Broker support for advanced redo transport settings
- Data Guard Broker support of prepared switchovers for Logical Standby

These enhancements make it possible to use Data Guard Broker in a wider variety of disaster recovery configurations.

See Also:

Oracle Data Guard Broker for details

1.2.3.3 Enhanced Data Guard Management Interface (Using SQL*Plus)

This feature makes it much simpler to use Oracle Data Guard administration-related SQL statements and initialization parameters.

The benefit of this feature is increased flexibility and manageability of disaster recovery configurations using Oracle Data Guard.

See Also:

Oracle Data Guard Concepts and Administration for details

1.2.3.4 Histogram for Redo Transport Response Time in a Data Guard Configuration

This feature advises on a suitable value for `NET_TIMEOUT` based on the response times for `SYNC` redo transport destinations.

The benefit is an increase in the flexibility and manageability of disaster recovery configurations using Data Guard.

See Also:

Oracle Data Guard Concepts and Administration for details

1.2.3.5 Snapshot Standby

A physical standby database can be open temporarily (that is, activated) for read/write activities such as reporting and testing. A physical standby database in such a state can still receive redo data from the primary database, thereby providing data protection for the primary database while still in the reporting database role.

This feature provides the combined benefit of disaster recovery, and reporting and testing using a physical standby database.

See Also:

Oracle Data Guard Concepts and Administration for details

1.2.3.6 Strong Authentication for Data Guard Redo Transport

Redo transport network sessions can now be authenticated using SSL. This provides strong authentication and makes the use of remote login password files optional in a Data Guard configuration.

The benefit is increased security of your disaster recovery configurations using Oracle Data Guard.

See Also:

Oracle Data Guard Concepts and Administration for details

1.2.3.7 Enhanced DDL Handling in Oracle Data Guard SQL Apply

SQL Apply executes in parallel (based on availability of parallel servers) DDLs that were executed with a parallel hint at the primary database.

1.2.3.8 Enhanced Oracle RAC Switchover Support for Logical Standby Databases

When switching over to a logical standby database where either the primary database or the standby database is using Oracle RAC, the `SWITCHOVER` command can be used without having to shut down any instance either at the primary or at the logical standby database.

1.2.4 Data Guard Logical Apply Completeness

The following sections describe features that add support for other database server features that are not currently handled by Oracle Data Guard. For example:

- LogMiner support for unsupported data types
- Objects and XML

1.2.4.1 Oracle Scheduler Support in Data Guard SQL Apply

It is now possible to use a logical standby database to provide data protection for a primary database in which the `DBMS_SCHEDULER` package was executed.

This enhancement allows Data Guard SQL Apply (logical standby database) to be used for a wider variety of applications.

See Also:

Oracle Data Guard Concepts and Administration for details

1.2.4.2 Fine-Grained Auditing (FGA) Support in Data Guard SQL Apply

It is now possible to use a logical standby database to provide data protection for primary databases using Oracle Fine-Grained Auditing (FGA).

This new support allows Data Guard SQL Apply (logical standby database) to be used for a wider variety of applications. In this case, applications with security features built-in.

See Also:

Oracle Data Guard Concepts and Administration for details

1.2.4.3 Support Transparent Data Encryption (TDE) with Data Guard SQL Apply

It is now possible to use a logical standby database to provide data protection for the primary database tables with Transparent Data Encryption (TDE).

This new support allows Data Guard SQL Apply (logical standby database) to be used for a wider variety of applications. In the case, applications with security features built-in.

See Also:

Oracle Data Guard Concepts and Administration for details

1.2.4.4 Support XMLType Data Type (Only CLOB) in Data Guard SQL Apply

It is now possible to use a logical standby database to provide data protection for primary database tables with the `XMLType` data type (only the `CLOB` representation).

This new support allows Data Guard SQL Apply (logical standby database) to be used in a wider variety of applications.

See Also:

Oracle Data Guard Concepts and Administration for details

1.2.4.5 Virtual Private Database (VPD) Support in Data Guard SQL Apply

It is now possible to use a logical standby database to provide data protection for primary database tables with Oracle Virtual Private Database (VPD).

This new support allows Data Guard SQL Apply (logical standby database) to be used for a wider variety of applications. In this case, applications with security features built-in.

See Also:

Oracle Data Guard Concepts and Administration for details

1.2.5 High Availability Improvements

The following sections describe general improvements in High Availability.

1.2.5.1 Flashback Transaction

Flashback transaction is a new feature that can easily back out a transaction and its dependent transactions. This recovery operation utilizes undo data to create and execute the corresponding, compensating transactions that revert the affected data back to its original state.

This feature increases availability during logical recovery by easily and quickly backing out a specific transaction or set of transactions and their dependent transactions, with one command, while the database remains online.

See Also:

Oracle Database Advanced Application Developer's Guide for details

1.2.5.2 SMP Scalable Redo Apply

Redo Apply performance has been significantly improved as a result of some internal optimizations in media recovery. Wait event generation in media recovery has also been enhanced.

This feature enables faster performance of media recovery and also Data Guard Redo Apply (physical standby database), thereby enabling real-time disaster recovery for more voluminous and highly transactional databases.

1.2.5.3 Transportable Databases Between Linux and Windows

This feature enables simple moving of databases between Linux and Windows.

This results in transportable databases between Linux and Windows enabling cross-platform physical standby and simple moving of databases between Linux and Windows.

1.2.6 Information Life Cycle Management

The features described in the following sections provide automated and policy-based movement of data between storage tiers and they provide mechanisms for complying with retention and immutability regulations.

1.2.6.1 Flashback Data Archive

The historical data store provides the ability to store and track all transactional changes to a record for the duration of its lifetime.

The database maintains every state of the record during its lifetime. You no longer need to build this intelligence into the application. This feature is useful for compliance reporting and audit reports.

See Also:

Oracle Database Advanced Application Developer's Guide for details

1.2.7 Integration, Simplification, and Performance of Availability Features

These features describe improvements to the interoperability, usability, performance and robustness of existing functionality.

1.2.7.1 Automatic Reporting of Corrupt Blocks

During instance recovery, if corrupt blocks are encountered, the `DBA_CORRUPTION_LIST` is automatically populated.

Block validation occurs at every level of backup, media recovery, and instance recovery.

1.2.7.2 Block Media Recovery Performance Improvements

In prior releases, block media recovery needed to restore original block images from disk or tape backup before applying needed archived logs. In this release, if flashback logging is enabled and contains older, uncorrupted blocks of the corrupt blocks in question, then these blocks will be used, speeding up the recovery operation.

The benefit is a reduction in the time it takes for block media recovery by restoring block images from flashback logs instead of from disk or tape backups.

See Also:

Oracle Database Backup and Recovery User's Guide for details

1.2.7.3 Creation and Recovery of Long-Term Backups Improved

Long-term backups created with the `KEEP` option only back up the archive logs needed to make the backup consistent. No further archive log backups are retained.

This feature reduces the archive log backup storage needed for online, long-term `KEEP` backups by only backing up the archive logs needed to recover the long-term backup.

See Also:

Oracle Database Backup and Recovery User's Guide for details

1.2.7.4 End-to-End Redo Validation

This feature enhances redo validation.

End-to-end redo validation provides foreground redo checksum at generation time. This improves CPU and cache efficiency for the checksum computation.

1.2.7.5 Merge Catalogs

The new `IMPORT CATALOG` command allows one catalog schema to be merged into another, either the whole schema or just the metadata for specific database IDs.

This feature simplifies catalog management by allowing separate catalog schemas, created at different versions, to be merged into a single catalog schema.

See Also:

Oracle Database Backup and Recovery User's Guide for details

1.2.7.6 Parallel Backup and Restore for Very Large Files

Backups of large data files now use multiple parallel server processes to efficiently distribute the workload for each file. This is especially useful for very large files.

This features improves the performance backups of large data files by parallelizing the workload for each file.

See Also:

Oracle Database Backup and Recovery User's Guide for details

1.2.8 Intelligent Data Protection and Repair

The following sections describe features that automate the process of determining what kind of recovery is needed and applying that needed recovery.

1.2.8.1 Data Recovery Advisor

Data Recovery Advisor is a built-in tool that automatically diagnoses data failures and reports the appropriate repair options.

Data Recovery Advisor ensures the correct repair is performed for a failure and decreases the amount of time to recover from a failure.

See Also:

Oracle Database Backup and Recovery User's Guide for details

1.2.8.2 Enhance LogMiner to Support Automatic Data Repair

This new feature adds support for Automatic Data Repair (ADR).

Automatic Data Repair (ADR) requires the ability to lookup metadata when the database is not available. This project enhances the existing functionality in LogMiner to support ADR.

1.2.8.3 Integrated Protection From Various Data Corruptions

Through the creation of a new initialization parameter, `DB_ULTRA_SAFE={off, data_only, data_and_index}`, this capability provides an integrated mechanism to offer protection from various possible data corruptions that may impact Oracle Database. This parameter controls the setting of other related parameters such as `DB_BLOCK_CHECKING`, `DB_BLOCK_CHECKSUM`, and `DB_LOST_WRITE_PROTECT`, and also controls other data protection behavior within Oracle Database (for example, requiring ASM to do sequential mirror writes).

This feature improves the data protection capabilities of Oracle Database by offering an integrated mechanism to control various levels of protection from data corruptions. By making it possible to detect data corruptions in a timely manner, this feature also provides critical high availability benefits for Oracle Database.

1.2.8.4 Lost-Write Detection Using a Physical Standby Database

Lost-write is a serious form of data corruption that can adversely impact a database. Lost-write occurs when an I/O subsystem acknowledges the completion of a block write in the database while, in fact, the write did not occur in the persistent storage. This arises mostly because of hardware or software malfunctioning. This feature enables the use of the physical standby database to detect data corruptions that could arise as a result of lost-writes on the primary database.

This feature allows for more comprehensive data corruption detection using a physical standby database, helping you get increased value from your disaster recovery investment and maintain the uptime requirements of your business.

See Also:

Oracle Data Guard Concepts and Administration for details

1.2.9 Online Application Maintenance and Upgrade

The following features can significantly reduce the downtime required to make changes to an application's database objects.

1.2.9.1 DDL With the WAIT Option

Data definition language (DDL) commands require exclusive locks on internal structures. If DDL commands are issued, these locks may not be available causing the statement to immediately fail even though the DDL could have possibly succeeded sub-seconds later. This can be resolved by setting the `WAIT` option in the `DDL_LOCK_TIMEOUT` initialization parameter.

The `WAIT` option is the new default. The wait time is specified instance-wide (in the initialization parameter file) and can be modified on a session level.

The `WAIT` option gives you more flexibility to define grace periods for such commands to succeed instead of raising an error right away, thus requiring additional application logic to handle such errors.

See Also:

Oracle Database Administrator's Guide for details

1.2.9.2 Enhanced ADD COLUMN Functionality

Default values of columns are maintained in the data dictionary for columns specified as `NOT NULL`.

Adding new columns with `DEFAULT` values and `NOT NULL` constraint no longer requires the default value to be stored in all existing records. This not only enables a schema modification in sub-seconds and independent of the existing data volume, it also consumes no space.

See Also:

Oracle Database Administrator's Guide for details

1.2.9.3 Finer Grained Dependencies

In previous releases, metadata recorded mutual dependencies between objects with the granularity of the whole object. For example, PL/SQL unit `P` depends on PL/SQL unit `Q` or that view `V` depends on table `T`. This means that dependent objects were sometimes invalidated when there was no logical requirement to do so. For example, if view `V` depends only on columns `C1`, `C2`, and `C3` in table `T` and a new column, `C99`, is added, the validity of view `V` is not logically affected. Nevertheless, in earlier releases, `V` was invalidated by the addition of column `C99`.

Oracle Database 11*g* records dependency metatdata at a finer level of granularity so that the addition of `C99` does not invalidate view `V`. Similarly, if procedure `P` depends only on elements `E1` and `E2` in package `PKG`, then if element `E99` is added to `PKG`, procedure `P` is not invalidated. (In Oracle Database 10*g*, this change to `PKG` would invalidate procedure `P`.)

By reducing the consequential invalidation of dependent objects in response to changes in the objects they depend upon, application availability is increased. The benefit is felt both in the development environment and when a live application is parsed or upgraded. The benefit occurs when an Oracle Database patchset is applied because changes to schema objects are required to be compatible and, therefore, not cause consequential invalidations.

See Also:

Oracle Database Concepts for details

1.2.9.4 Invisible Indexes

An invisible index is an alternative to making an index unusable or even to drop it. An invisible index is maintained for any DML operation but is not used by the optimizer unless you explicitly specify the index with a hint.

Applications often have to be modified without being able to bring the complete application offline. Create invisible indexes temporarily for specialized non-standard operations, such as online application upgrades, without affecting the behavior of any existing application. Furthermore, invisible indexes can be used to test the removal of an index without dropping it right away, thus enabling a grace period for testing in production environments.

See Also:

Oracle Database Administrator's Guide for details

1.2.9.5 Materialized View Logging Control

In this release, Oracle Database has added session-level control for materialized view logs. The capture of changes for materialized views (materialized view logs) can be disabled for an individual session while logging continues for changes made by other sessions.

This feature reduces downtime for application upgrade.

1.2.9.6 Minimize Dependent PL/SQL Recompilation After Online Table Redefinition

This feature minimizes the need to recompile dependent PL/SQL packages after an online table redefinition. If the redefinition does not logically affect the PL/SQL packages, recompilation is not needed. This optimization is on by default.

This feature reduces the time and effort to manually recompile dependent PL/SQL after an online table redefinition. This also includes views, synonyms, and other table dependent objects (with the exception of triggers) that are not logically affected by the redefinition.

1.2.9.7 Online Index Creation and Rebuild Enhancements

In highly concurrent environments, the requirement of acquiring a DML-blocking lock at the beginning and end of an online index creation and rebuild could lead to spikes of waiting DML operations and, therefore, a short drop and spike of system usage. While this is not an overall problem for the database, this anomaly in system usage could trigger operating system alarm levels. This feature eliminates the need for DML-blocking locks when creating or rebuilding an online index.

Online index creation and rebuild prior to this release required a DML-blocking lock at the beginning and end of the rebuild for a short period of time. This meant that there would be two points at which DML activity came to a halt. This DML-blocking lock is no longer required, making these online index operations fully transparent.

See Also:

Oracle Database Administrator's Guide for details

1.2.9.8 Online Redefinition for Tables with Materialized View Logs

In previous releases, a table could not be redefined if it had a log or materialized views defined.

With this feature, this restriction has been lifted.

You can now clone the materialized view log onto the interim just like triggers, indexes, and other similar dependent objects. At the end of the redefinition, `ROWID` logs are invalidated. Initially, all dependent materialized views need to do a complete refresh.

See Also:

Oracle Database Administrator's Guide for details

1.2.9.9 Read-Only Tables

Oracle Database 11*g* introduces new `ALTER TABLE` syntax. For example:

```
ALTER TABLE <name> READ ONLY
and
ALTER TABLE <name> READ WRITE
```

The operating system sets the precedent to make a file read-only even for its owner. Earlier, a table could be made read-only (by granting only `SELECT` on it) to users other than the owner of the table. With this feature, the owner too can be prevented from doing unintended DML to a table.

1.2.10 RMAN Integration, Simplification, and Performance

The following sections describe RMAN integration, simplification, and performance features for Oracle Database 11*g* Release 1 (11.1).

1.2.10.1 Archive Log Management Improvements

This feature provides the following enhancements:

- Ensure that archive logs are deleted only when not needed by required components (for example, Data Guard, Streams, and Flashback).
- In a Data Guard environment, allow all standby destinations to be considered where logs are applied (instead of just mandatory destinations), before marking archive logs to be deleted. This configuration is specified using `CONFIGURE ARCHIVELOG DELETION POLICY TO APPLIED ON ALL STANDBY`.
- Allow optional archive log destination to be utilized in the event that the flash recovery area is inaccessible during backup. Archive logs in this optional destination can be deleted using `BACKUP .. DELETE INPUT` or `DELETE ARCHIVELOG`.

This feature simplifies archive log management when used by multiple components. It also increases availability when backing up archive logs, when an archive log in the flash recovery area is missing or inaccessible. In this case, the backup will failover to an optional archive log destination to continue backing up the archive logs.

See Also:

Oracle Database Backup and Recovery User's Guide for details

1.2.10.2 Fast Incremental Backups on Physical Standby Database

You can enable block change tracking on a physical standby database. RMAN uses the change tracking file, on incremental backups, to quickly identify the changed blocks since the last incremental backup and to read and write just those blocks.

This feature enables faster incremental backups on a physical standby database than in previous releases.

1.2.10.3 Improved Backup Compression Performance

In this release, when creating compressed backup sets, performance will be noticeably faster than in previous releases.

The benefit is faster backup compression performance.

See Also:

Oracle Database Backup and Recovery User's Guide for details

1.2.10.4 Improved Integration with Data Guard

Improved integration with Data Guard includes the following:

- Persistent configurations can be set for each primary and standby database. You now have the ability to modify database-specific configurations in the catalog without having to connect to the specific primary or standby database.
- Allow conversion of a backup control file to a standby control file and vice versa. This eliminates the need to explicitly create a standby control file from the primary database.
- Share backups among primary and standby databases in a storage area network (SAN) by supporting restore to or from different mount points.
- Define which backups are available to particular databases, for restore operations, based on the network topology.

This feature simplifies backup and recovery operations across all primary and physical standby databases when using a catalog.

See Also:

Oracle Data Guard Concepts and Administration for details

1.2.10.5 Network-Aware DUPLICATE Command

A clone database on a remote site can now be easily created directly over the network with the enhanced `DUPLICATE` command without existing backups. ASM-to-ASM `DUPLICATE` over the network is also supported.

This feature eliminates the need to copy or move backups to the remote site before executing the `DUPLICATE` command. This reduces DBA time and effort, and eliminates storage for the additional copy at the remote site.

See Also:

Oracle Database Backup and Recovery User's Guide for details

1.2.10.6 Optimized Undo Backup

During backup, undo that is not needed for recovery of the backup (for example, for transactions that have already been committed), is not backed up. In past releases, all undo was backed up. This new feature is enabled with `CONFIGURE BACKUP UNDO OPTIMIZATION`. If you wish to preserve active undo within the `UNDO_RETENTION` period in the backup (for example, for future flashback query or flashback table operations), an option is available to only eliminate the inactive undo beyond the `UNDO_RETENTION` period.

The benefit is reduced overall backup time and storage by not backing up undo that applies to transactions that have already been committed by the time the backup starts.

1.2.10.7 Read-Only Transported Tablespaces Backup

Transported tablespaces can be backed up, whether in read-only or read/write mode. This feature removes the restriction in previous releases, that transported tablespaces must be enabled for read/write before making a backup.

The feature allows transported tablespaces to be immediately backed up, without having to be made read/write first.

See Also:

Oracle Database Backup and Recovery User's Guide for details

1.3 Database Overall

The following sections describe overall database features for Oracle Database 11*g* Release 1 (11.1).

1.3.1 Utilities

The database utilities give you the ability to move large volumes of data and metadata into and between Oracle databases with a high degree of performance, reliability, and manageability. This area includes Upgrade/Downgrade, Data Pump Export and Import, the Metadata API, and the SQL*Loader utility.

1.3.1.1 Compressed Dump File Sets

Data Pump provides the ability to compress the metadata of a dump file set, which can shrink the dump files by 10-15%.

In this release, Data Pump adds the ability to compress the entire dump file set, including data and metadata.

This feature lets you specify compression for metadata, row data, or the entire dump file set.

See Also:

Oracle Database Utilities for details

1.3.1.2 Data Pump API Enhancements

The Data Pump API has been enhanced to allow individual worker processes to be restarted after a failure without terminating the Data Pump job. This enhancement improves availability when using the parallel capabilities of Data Pump.

The new DATA_REMAP API allows you to specify transformations to be applied to data during export or import. This can be useful when you need to obfuscate sensitive data while retaining its data type, structure, or format which may be needed when moving data from a production system to a test environment.

1.3.1.3 Database Upgrade/Downgrade Enhancements

Oracle Database 11*g* upgrade now includes enhanced information gathering and reporting facilities. The pre-upgrade information utility has improved resource estimations, and the post-upgrade status tool has improved error message generation and display.

Over the past three database releases, improvements have been made to simplify both manual upgrades and upgrades done using the Database Upgrade Assistant (DBUA). This feature further reduces the complexity of database upgrades and gives DBAs more information they can use to determine whether their database upgrade was successful.

See Also:

Oracle Database Upgrade Guide for details

1.3.1.4 Direct Path API Multiple Subtype Support

Prior to Oracle Database 11*g*, only one object type could be loaded into an object column or object table when using the Direct Path API.

This feature extends that support to allow loading of multiple subtypes into a column object or object table using Direct Path INSERT, thereby providing more robust support for objects and for XML schema data. Data Pump now supports the export and import of XML data.

See Also:

Oracle XML DB Developer's Guide for details

1.3.1.5 Encrypted Dump File Sets

In this release, Data Pump adds the ability to encrypt the dump file set, including data and metadata. You can choose to encrypt part or all of the dump file set by specifying encryption for encrypted columns, metadata, row data, all, or none.

Customers want sensitive data residing in dump files to be secure.

1.3.1.6 Enhancements to Metadata API

The Metadata API heterogeneous object types have been enhanced to support all object types currently supported by original Export (exp). In particular, XML Schemas and schema-based tables are now supported.

Data Pump also supports editable XML. You can extract metadata from the dictionary, manipulate it, and then convert the metadata to DDL so that the object can be re-created on the same database or on another database.

The TABLE_EXPORT heterogeneous object type is enhanced to properly support AQ tables. This allows AQ tables to be exported. All dependent objects for the table are also exported.

The Metadata API provides the ability to transform a local index defined on a single partition into an index on a nonpartitioned table.

1.3.1.7 Oracle Data Pump External Table Enhancement

In previous releases of Data Pump, a row error in an external table caused the entire table load to abort, requiring you to reload the table.

This feature ensures that a row error does not cause a table load to abort, thus saving load time.

1.3.1.8 Single Partition Transportable for Oracle Data Pump

Until this release, the Transportable Tablespaces mechanism for Data Pump could be used to specify only the physical tablespaces to be exported. This feature adds a partition mode, which can be used to move one or more partitions or subpartitions of a table without having to move the entire table or exchange out the partition or subpartition.

In addition, partitions can be imported to the target database either as part of an existing table or as a separate table for each partition.

See Also:

Oracle Database Utilities for details

1.3.1.9 SQL*Plus BLOB Support

Queries in SQL*Plus now support `BLOB` columns.

This feature allows you to verify the contents of `BLOB` columns.

See Also:

*SQL*Plus User's Guide and Reference* for details

1.4 Business Intelligence and Data Warehousing

The following sections describe Business Intelligence and Data Warehousing features for Oracle Database 11*g* Release 1 (11.1)

1.4.1 Complete the Information Cycle

These features tighten the integration of operational and analytical databases, to enable higher quality and more up-to-date information.

1.4.1.1 Change Data Purge Enhancements

In earlier releases, purging the change table would always remove all consumed change data from the change table. The change data purge enhancements feature introduces a date range-based model to purge change data.

This feature increases the flexibility of purging change data from the change tables. Users can specify a date range for which data should be purged.

See Also:

Oracle Database Data Warehousing Guide for details

1.4.1.2 Change Subscription Enhancements

In this release, the change data subscription window can now be moved forward and backward.

This feature makes it easier to maintain a subscription window to change data. You are given more control over the definition of the subscription window.

See Also:

Oracle Database Data Warehousing Guide for details

1.4.1.3 Enabling or Disabling Synchronous Change Data Capture

This feature adds an interface to enable users to turn synchronous Change Data Capture (CDC) off once it has been started, and turn it on when needed.

With this feature, users have more flexibility and control of the use of synchronous CDC to meet their needs.

1.4.1.4 Enhanced DML Tracking

The enhanced DML tracking feature enables change capture for all DML operations, including direct path loads and partition maintenance operations. A separate record in the change table indicates that such an operation has taken place. This feature also extends Change Data Capture support for data types including `CLOB`, `NCLOB`, and `BLOB` in the asynchronous modes.

Change Data Capture formerly did not always capture direct path load operations or implicit data changes as the result of partition maintenance operations. The enhanced DML tracking feature relieves this restriction.

See Also:

Oracle Database Data Warehousing Guide for details

1.4.1.5 Enhanced Materialized View Catalog Views

New catalog views display the partition change tracking (PCT) information for a given materialized view by showing which sections of the materialized views data are fresh or stale.

This is a critical piece of information for the user to be able to view the partition staleness information of the materialized view. It affects the usability and maintainability of the materialized view.

See Also:

Oracle Database Data Warehousing Guide for details

1.4.1.6 Query Rewrite Enhancement to Support Queries Containing Inline Views

In this release, query rewrite has been enhanced to support queries containing inline views.

With this feature, more queries are eligible for query rewrite thus improving system throughput and performance.

See Also:

Oracle Database Data Warehousing Guide for details

1.4.1.7 Query Rewrite Support of Queries on Remote Tables

In this release, query rewrite has been enhanced to support queries when the tables are remote.

With this feature, more queries are eligible for query rewrite thus improving system throughput and performance.

See Also:

Oracle Database Data Warehousing Guide for details

1.4.1.8 Refresh Performance Improvements

Refresh operations on materialized views are now faster with the following improvements:

- Refresh statement combinations (merge and delete)
- Removal of unnecessary refresh hint
- Index creation for `UNION ALL MV`
- PCT refresh possible for `UNION ALL MV`

With this feature, the time required to refresh materialized views is reduced.

See Also:

Oracle Database Data Warehousing Guide for details

1.4.2 Enable the Information Grid

The new features in the following sections:

- Provide a single source for information, handling both operational and analytic workloads.
- Support database systems with hundreds of CPU's, hundreds of TBs, thousands of users, and dozens of applications.

1.4.2.1 Composite List-Hash Partitioning

Composite list-hash partitioning enables hash subpartitioning of a list-partitioned object to, for example, enable partition-wise joins.

Composite list-hash partitioning provides an additional modeling strategy to optimally map a business need to the partitioning of an object.

1.4.2.2 Composite List-List Partitioning

Composite list-list partitioning enables logical list partitioning along two dimensions. For example, list partition by `country_id` and list subpartition by `sales_channel`.

Composite list-list partitioning provides an additional modeling strategy to optimally map a business need to the partitioning of an object.

1.4.2.3 Composite List-Range Partitioning

Composite list-range partitioning enables logical range subpartitioning within a given list partitioning strategy. For example, list partition by `country_id` and range subpartition by `order_date`.

Composite list-range partitioning provides an additional modeling strategy to optimally map a business need to the partitioning of an object.

See Also:

Oracle Database VLDB and Partitioning Guide for details

1.4.2.4 Composite Range-Range Partitioning

Composite range-range partitioning enables logical range partitioning along two dimensions. For example, partition by `order_date` and range subpartition by `shipping_date`.

Composite range-range partitioning provides an additional modeling strategy to optimally map a business need to the partitioning of an object.

1.4.2.5 Interval Partitioning

This new partitioning strategy fully automates the partition creation for range. In other words, new partitions will be created when they are needed. By defining the interval criteria, the database knows when to create new partitions for new or modified data.

Managing the creation of new partitions can be a cumbersome and highly repetitive task. This is especially true for predictable additions of partitions covering small ranges, such as adding new daily partitions. Interval partitioning automates this operation by creating partitions on-demand.

See Also:

Oracle Database VLDB and Partitioning Guide for details

1.4.2.6 Metadata for SQL Built-In Operators and Functions

In this release, metadata for all Oracle SQL built-in operations and functions has been made accessible through fixed tables in the dictionary.

This feature enables all third party tools to leverage any built-in Oracle SQL operation or function without having to maintain the function usage metadata redundantly in the application layer.

1.4.2.7 OLTP Table Compression

The Oracle compression feature allows heap tables to be stored in a compressed format, resulting in significant savings in disk storage, I/O, and redo logs. This feature extends the compression feature to OLTP environments so that compression works seamlessly for all DMLs.

Oracle's heap segment compression (HSC) has been well received in data warehousing environments. With this feature compression is extended to OLTP environments so that compression works well with all DMLs including conventional `INSERT`, `UPDATE`, and `DELETE`.

See Also:

Oracle Database Administrator's Guide for details

1.4.2.8 Reference Partitioning

The partitioning key is resolved through an existing parent-child relationship, enforced by active primary key or foreign key constraints.

The benefit of this feature is that tables with a parent-child relationship can be logically equi-partitioned by inheriting the partition key from the parent table without duplicating the key columns. The logical dependency will also automatically cascade partition maintenance operations, thus making application development easier and less error-prone.

See Also:

Oracle Database VLDB and Partitioning Guide for details

1.4.2.9 SQL PIVOT and UNPIVOT Operators

The `PIVOT` and `UNPIVOT` operators are extensions to the table expression in the `FROM` clause of a `SELECT` statement. `PIVOT` spreads values from multiple rows into multiple columns, aggregating data in the process. PIVOT is commonly used to create a result set with more columns and fewer rows than the source data. The `PIVOT` operator supports multiple pivot columns, multiple aggregates, wildcards, and aliasing. `UNPIVOT` moves data in the opposite direction from `PIVOT`. For each input row, `UNPIVOT` moves values from multiple columns into multiple output rows. The process creates a result set with fewer columns and more rows than the source data. `UNPIVOT` supports multiple unpivot columns, multiple measure columns, and aliasing.

`PIVOT` can create aggregated cross-tabular output that condenses many rows into a compact result set. For example, input data holding sales of 1 month per row can be pivoted into output holding 12 months per row, with each month in its own column. Another use of `PIVOT` is to combine multiple input rows into a single output row, enabling inter-row comparison without a table self-join. `UNPIVOT` reshapes data into a format useful for further relational operations. For example, if a source data set presents 12 months of sales values per row, `UNPIVOT` can reshape each source row into 12 output rows, each holding 1 month of sales data. The unpivoted results can, in turn, be manipulated with much simpler and more efficient SQL than required for the source data set.

See Also:

Oracle Database Data Warehousing Guide for details

1.4.2.10 System Partitioning

System partitioning enables application-controlled partitioning for arbitrary tables and indexes. The database simply provides the ability to break down an object into partitions without any data placement rules. All aspects of data placement and retrieval are controlled by the application.

System partitioning provides the well-known benefits of partitioning (scalability, availability, and manageability), but the partitioning and actual data placement are controlled by the application.

See Also:

Oracle Database Data Cartridge Developer's Guide for details

1.4.2.11 Virtual Columns

Virtual columns are defined by evaluating an expression the results of which become the metadata of the columns for tables. Virtual columns can be defined at table creation or modification time.

Virtual columns enable application developers to define computations and transformations as the column (metadata) definition of tables without space consumption. This makes application development easier and less error-prone, as well as enhances query optimization by providing additional statistics to the optimizer for these virtual columns.

1.4.2.12 Virtual Column-Based Partitioning

In Oracle Database 11*g*, you can now partition key columns defined on virtual columns of a table.

Frequently, business requirements to logically partition objects does not match existing columns in a one-to-one manner. Oracle partitioning has been enhanced to allow a partitioning strategy being defined on virtual columns, thus enabling a more comprehensive match of the business requirements.

See Also:

Oracle Database VLDB and Partitioning Guide for details

1.4.3 OLAP in Every Data Warehouse

The following sections describe the integration of OLAP and relational technology to broaden the appeal of OLAP for relational data warehouses, and also extend core OLAP capabilities.

1.4.3.1 Cost-Based Aggregation of OLAP Cubes

Cost-based aggregation of cubes provides a statistics-based approach to determining the best data within the cube to preaggregate and store for both efficient updates of the cube and efficient query.

Cost-based aggregation provides the DBA with an easy-to-use method of optimizing the cube as a general purpose summary management solution. The DBA needs to indicate only the amount of resources that should be applied to the task of optimizing summary data. Cost-based aggregation automatically determines and executes the best summary management plan to support ad-hoc query and reporting of the cube. This results in shorter refresh times and better query times.

1.4.3.2 Cube Organized Materialized Views

Cube organized materialized views represent the cube to SQL-based applications as materialized views that can be used for both refresh and query rewrite. Cube organized views are metadata objects that access data stored or calculated with the OLAP cube; they do not replicate data from the cube into new storage.

Cube organized materialized views can improve the ease and efficiency of dimension and cube updates and improve query performance.

Cube organized materialized views allow the database administrator to manage OLAP dimension and cube updates using the materialized view refresh system of Oracle Database, thus leveraging existing DBA skills and incremental data loading using materialized view log tables.

Cube organized materialized views can improve query performance by allowing SQL-based applications querying detail relational tables to automatically and transparently access summary data in the OLAP cube.

See Also:

Oracle OLAP User's Guide for details

1.4.3.3 Fine-Grained Partitioning of OLAP Cubes

Scalability improvements in cube partitioning increase the number of partitions that can be efficiently maintained in a cube.

Fine-grained partitioning of OLAP cubes provide support for larger cubes and more efficient cube update processing. Fine-grained partitioning allows the cube to be partitioned by more members of a dimension (as compared to Oracle Database 10*g* Release 2). Support for many more partitions can be used to create larger cubes (for example, retaining more time periods). Smaller partitions can be processed more quickly during a cube update.

1.4.3.4 OLAP Security Enhancements

Security of OLAP cubes and dimensions is enhanced to include support for Oracle Database object privileges (`SELECT`, `INSERT`, `UPDATE` and `DELETE`) on analytic workspaces, cubes and dimensions. For example, `GRANT SELECT ON sales_cube`.

Fine-grained access control to OLAP dimensions and cubes, which provides support for granting access privileges (SELECT, INSERT, UPDATE and DELETE) on data within a dimension and cube, is also supported in OLAP 11*g* and Analytic Workspace Manager.

For the general OLAP or database user, this collection of features enhances the integration of multidimensional data types with standard operational features of the database with the goal of making multidimensional data types more secure and easier to manage.

See Also:

Oracle OLAP User's Guide for details

1.4.3.5 SQL Optimizer Support for OLAP Option Cubes and Dimensions

The SQL Optimizer adds support for OLAP cube and dimension statistics and for pushing joins between cubes and dimensions into the OLAP cube. Optimizations also include efficient joins between relational tables and dimension and cube views.

SQL Optimizer support for OLAP cubes and dimensions provides general performance improvements and support for a wider variety of SQL (including counts and joins) to be efficiently executed against the cube. The main benefit is that a wider variety of SQL-based tools and applications will be able to easily query the cube.

See Also:

Oracle Database PL/SQL Packages and Types Reference for details

1.4.3.6 Storage and Access Improvements of Compressed Cubes

This feature provides internal scalability and performance improvements of compressed cubes.

This feature supports larger compressed cubes and broadens the applicability of compressed cubes to include a wider variety of data sets.

1.4.3.7 Unified OLAP API Metadata Change Management

Unified OLAP API metadata change management involves initial deployment of Analytic Workspace (AW) dimension objects, dependency tracking for these objects, and merging new versions of AWs and AW metadata with site-specific customizations.

Unified OLAP API metadata change management allows you to make site-specific modifications to the dimensional model. This feature is primarily in support of Oracle applications.

See Also:

Oracle OLAP Java API Developer's Guide for details

1.4.4 Predictive Analytics and Data Mining

Oracle Database 11*g* simplifies the management of data mining processing, increases the integration of data mining with SQL, and introduces new analytic capabilities.

1.4.4.1 Automated and Embedded Data Transformations for Mining (Supermodels)

This new feature in Oracle Database 11*g* can:

- Perform automatic data preparation for data mining models to reduce the burden on the user and provide an improved out-of-the-box experience (supermodels).
- Encapsulate the transformations with the model metadata to simplify model deployment (the transformations are automatically deployed with the model itself).
- Allow advanced users to specify some amount of transformations to the model build routine, which simplifies deployment for the advanced user as well.

This feature adds support to automate data preparation, the most difficult part of the data mining process. Another benefit is the simplification and automation of the overall data preparation process which simplifies model building and model scoring activities.

See Also:

Oracle Data Mining Concepts for details

1.4.4.2 Data Mining Schema Objects

In Oracle Database 10*g* Release 2, Data Mining models were not defined in the database at the same class level as other DBMS objects. In Oracle Database 11*g*, models and the Data Mining environment are enhanced as follows:

- Eliminate the `DMSYS` schema, migrating necessary information to `SYS`
- Enhance existing catalog views to be Data Mining model aware
- Add new catalog views to expose Data Mining model metadata
- Add specific privileges for building, viewing, and scoring Data Mining models

Data Mining models are now more similar to other database objects. This allows you to assign access privileges, start and stop Data Mining models from Oracle Enterprise Manager and, in general, manage Data Mining models as full database objects.

See Also:

Oracle Data Mining Administrator's Guide for details

1.4.4.3 Java API for Oracle Data Mining (JSR-73 and JSR-247)

JSR-247 provides algorithms and functions to meet new features provided in Oracle Data Mining for Oracle Database 11*g*; for example, Multiple Linear and Logistic Regression. Doing so maintains equivalent functionality between the SQL and PL/SQL APIs and Java API.

This feature provides enhancements for the Java API for Oracle Data Mining conforming to the Java Data Mining standard (JSR-73 and JSR-247).

See Also:

Oracle Data Mining Application Developer's Guide for details

1.4.4.4 Multivariate Linear Regression

Multivariate Linear Regression (MLinR) enables data analysts to build classic statistical models of the relationships between a range of input attributes and a target (numerical) output attribute. Additionally, MLinR finds the strength of the relationships between variables. Oracle Data Mining MLinR supports a large number of input attributes. This is a differentiator as most MLinR implementations cannot do this well. MLinR models can score data using the `PREDICTION` operators introduced in Oracle Database 10*g* Release 2. Several model diagnostics, both at global and attribute level, can be reported.

Oracle Database 11*g* augments the statistical capabilities available in Oracle Database by adding support for Multivariate Linear Regression in Oracle Data Mining.

1.4.4.5 Multivariate Logistic Regression

Multivariate Logistic Regression (MLogR) enables data analysts to build classic statistical models of the relationships between a range of input attributes and a target (categorical) output attribute. Additionally, MLogR finds the strength of the relationships between variables. The Oracle Data Mining Multivariate Linear Regression (MLinR) supports a large number of input attributes. This is a differentiator as most MLogR implementations cannot do this well. MLogR models can score data using the `PREDICTION` operators introduced in Oracle Database 10*g* Release 2. Several model diagnostics, both at global and attribute level, can be reported.

Oracle Database 11*g* Release 1 (11.1) augments the statistical capabilities available in the database by adding support for Multivariate Logistic Regression in Oracle Data Mining.

See Also:

Oracle Data Mining Concepts for details

1.4.4.6 Predictive Analytics: PROFILE

The Predictive Analytics (PA) package, introduced in Oracle Database 10*g* Release 2, provided a very simple interface for business users to leverage the power of Data Mining without the requisite statistical background. Also, in Oracle Database 10*g* Release 2, functions were provided to `PREDICT` (classification and regression) and to `EXPLAIN` (attribute importance). In Oracle Database 11*g*, a new Predictive Analytics (PA) method is added to develop profiles of a specified target attribute. `PROFILE` uses the decision tree algorithm.

`PROFILE` Predictive Analytics automatically sorts through the records and finds patterns and relationships of similar records (or customers) given a specified target attribute which designates a business problem. For example, `PROFILE` can be used to find the various profiles of customers who churn versus those who remain loyal. Similarly, `PROFILE` can be used to find the profiles of highly valued customers, customers who commit fraud, patients who contract a disease, and so on.

See Also:

Oracle Data Mining Concepts for details

1.4.4.7 SQL Prediction Enhancements

SQL built-in functions for Data Mining scoring (for example, `PREDICTION`) were added in Oracle Database 10*g* Release 2. Oracle Database 11*g* is enhanced to:

- Allow cost matrix identification in the `PREDICTION` family to push down cost computations and provide a simple, best prediction by cost.
- Enable on-the-fly model builds (requiring no user intervention) with associated data preparation, as well as a mechanism for model inspection for the more advanced user).
- Enhance our `*_DETAILS` routines to support more algorithms and provide more details than the current form.

SQL Prediction enhancements allow for rapid, on-the-fly Data Mining model creation and tuning. This enables Data Mining to be more easily and seamlessly integrated into applications.

See Also:

Oracle Data Mining Application Developer's Guide for details

1.5 Clustering

The following sections describe clustering features for Oracle Database 11*g* Release 1 (11.1).

1.5.1 Oracle Real Application Clusters Ease-of-Use

The following sections describe Oracle RAC enhancements that make clustering and Oracle RAC setup and installation easier to perform by anyone unfamiliar with both Oracle RAC and clustering.

1.5.1.1 Enhanced Oracle RAC Monitoring and Diagnostics in Enterprise Manager

This feature improves Oracle RAC performance monitoring and diagnostics through the GUI interface in Enterprise Manager. Specifically:

- Enhanced service-based monitoring.
- Summary and tile-based views for monitoring performance metrics.

This enhanced GUI interface eases management of Oracle RAC.

See Also:

Oracle Database 2 Day + Real Application Clusters Guide for details

1.5.1.2 Enhanced Oracle Real Application Clusters Configuration Assistants

This feature updates the graphical configuration utilities (DBCA, DBUA and NetCA) for all features of Oracle RAC. In addition, the `rconfig` utility is enhanced to convert single instance ASM to clustered ASM.

The benefit is a reduction in the management costs of cluster database environments through wizard-driven GUI management tools.

See Also:

Your platform-specific Oracle Real Application Clusters installation and configuration guide for details

1.5.1.3 OCI Runtime Connection Load Balancing

In Oracle RAC environments, session pools to services use service metrics to load balance sessions across different instances at runtime. OCI and OCCI applications have this feature enabled by default provided the necessary events are turned on.

Applications using OCI Session Pools have the best possible throughput for the resources provided. No application changes are required. Load balancing works toward the goal specified for the service. If an instance becomes hung or unresponsive, connections are directed to other instances that can provide better service.

See Also:

Oracle Call Interface Programmer's Guide for details

1.5.1.4 Parallel Execution for Oracle Real Application Clusters

Parallel execution is now aware of the service definition and automatically takes on the appropriate `PARALLEL_INSTANCE_GROUPS` setting, thus making the explicit setting of `PARALLEL_INSTANCE_GROUPS` for a services-controlled information grid environment unnecessary.

Oracle RAC services and the control of eligible nodes for parallel execution are now integrated and controlled by the services setup, making parallel execution fully transparent in an Oracle RAC services-controlled environment.

See Also:

Oracle Real Application Clusters Administration and Deployment Guide for details

1.5.1.5 Support for Distributed Transactions in an Oracle RAC Environment

An XA transaction can now span Oracle RAC instances. Individual branches have the ability to execute on different instances in the cluster.

This feature allows any application using XA to take full advantage of the Oracle RAC environment, thus enhancing high availability and scalability of the application.

See Also:

Oracle Database Advanced Application Developer's Guide for details

1.6 Content Management Services

The following sections describe enhancements to Content Management Services.

1.6.1 Oracle SecureFiles

The following sections describe enhancements to the LOB infrastructure in this release, called Oracle SecureFiles. Enhancements include built-in compression and encryption.

1.6.1.1 Fast Bulk Data Transfers

Oracle Database now uses a larger session data unit (SDU) by default. Optimizations to the Network Session SDU size result in significant performance gains for large data transfers.

This feature also takes advantage of zero copy and vectored I/O at the network level.

Fast bulk data transfers result in significant performance improvements for large data transfers.

See Also:

Oracle Database Net Services Administrator's Guide for details

1.6.1.2 LOB Prefetch

OCI now allows programs to set the amounts of prefetching of LOB data for optimized information retrieval, especially of smaller data sizes.

Business applications that often access smaller LOBs perform faster, because there are fewer round trips to the server for data.

See Also:

Oracle Call Interface Programmer's Guide for details

1.6.1.3 SecureFiles

This feature introduces a completely reengineered large object (LOB) data type to dramatically improve performance, manageability, and ease of application development. The new implementation also offers advanced, next-generation functionality such as intelligent compression and transparent encryption.

This feature significantly strengthens the native content management capabilities of Oracle Database.

See Also:

Oracle Database SecureFiles and Large Objects Developer's Guide for details

1.6.1.4 SecureFiles: Compression

This feature allows you to explicitly compress SecureFiles to gain disk, I/O, and redo logging savings.

The benefits of this feature are:

- Reduced costs due to the most efficient utilization of space.
- Improved performance of SecureFiles as compression reduces I/O and redo logging (at some CPU expense).

See Also:

Oracle Database SecureFiles and Large Objects Developer's Guide for details

1.6.1.5 SecureFiles: Deduplication

Oracle Database can now automatically detect duplicate SecureFiles LOB data and conserve space by storing only one copy.

This feature implements disk storage, I/O, and redo logging savings for SecureFiles.

See Also:

Oracle Database SecureFiles and Large Objects Developer's Guide for details

1.6.1.6 SecureFiles: Encryption

This feature introduces a new encryption facility for SecureFiles. The encrypted data is now stored in-place and is available for random reads and writes.

The benefit of this feature is enhanced data security.

See Also:

Oracle Database SecureFiles and Large Objects Developer's Guide for details

1.6.2 Text Manageability

The following sections describe enhancements to Text Manageability, which include:

- Improving ease-of-use and self-manageability of the text subsystem by providing Enterprise Manager support.
- Exposing the text subsystem as Text services through Java and Web for ease of application development.

1.6.2.1 Improved Support for Advanced Features in More Languages

Oracle Database 11*g* includes multilingual and linguistic support that improves Oracle Text handling of documents in different languages. Two new components are introduced:

- A new lexer that automatically detects the language of the document.
- A new document service that detects the character set and language of a document.

Oracle Text supports keyword searches in all Unicode languages. However, advanced features such as stemming and alternate spelling require lexing support in different languages. The new multilingual and linguistic support make advanced search available in 28 languages including Arabic, Hebrew, and Russian.

See Also:

Oracle Text Reference for details

1.6.2.2 Incremental Indexing Enhancements

This feature introduces three improvements that facilitate large text index creation:

- The new `ctx_ddl.populate_pending` interface
- New `sync_index` enhancements
- `NOPOPULATE` support

In large text warehouses, applications cannot afford to have the indexing process running continuously. This feature provides interfaces that let applications create large indexes in a manageable way.

See Also:

Oracle Text Reference for details

1.6.2.3 Oracle Enterprise Manager Support for Text

This feature provides improved Oracle Enterprise Manager support for Oracle Text.

DBAs can now administer Oracle Text from Oracle Enterprise Manager.

See Also:

Oracle Text Reference for details

1.6.2.4 Re-Create Index Online

This feature provides the ability to re-create an Oracle Text index without producing any undesirable query results until the application is ready to switch over to the modified index.

Users can now re-create an Oracle Text index with new preference values, while preserving base table DML and query capability during the re-creation. Users can re-create the index in one operation or can step through each stage of the re-creation manually.

See Also:

Oracle Text Reference for details

1.6.3 Text Performance and Scalability

The following sections describe feature integration with Oracle RAC for scalability and other subsystems such as the optimizer for performance.

1.6.3.1 Composite Index

This feature facilitates structured `ORDER BY` criteria, structured range, or combinations of both. This is accomplished by allowing the specification of `FILTER BY` and `ORDER BY` structured column(s) at index creation time.

This feature provides better performance for mixed queries involving relational and text predicates required by today's Web applications.

See Also:

Oracle Text Reference for details

1.6.3.2 More Types of Operations Allowed on Document Sections

This feature introduces a new type of document section called Structured/Sort Data (`SDATA`). The content of an `SDATA` section is typed and not tokenized. `SDATA` sections support range and equality query operations.

The benefit of this feature is faster queries on document metadata by pushing more metadata into the Text index.

See Also:

Oracle Text Reference for details

1.6.3.3 Text Support for Very Large Number of Partitions

Until now, the maximum number of partitions allowed has been 9,999. This limit has been increased to 1,223,054.

This increase is of significant benefit to Text users.

See Also:

Oracle Text Application Developer's Guide for details

1.6.3.4 User-Defined Score

User-defined scoring offers users a mechanism to define how the `CONTAINS` query will score textual content. This mechanism can use the `DEFINESCORE` or the `DEFINEMERGE` operator.

In some cases, the application would improve from more direct control of how to score documents based on structured values like date. The user-defined score feature allows applications to customize scoring of textual content.

See Also:

Oracle Text Reference for details

1.6.4 XML Content Management Platform

The following sections describe enhancements to XML-based Content Management.

1.6.4.1 Content Repository Management (JSR-170) Support for Java

This feature supports JSR-170 1.0 defined operations on both unstructured and semi-structured content and provides a comprehensive Oracle-proprietary content management platform, which includes:

- API extensions for access control, based on the DAV Access Control model
- Support for type definition and evolution, based on Oracle XML DB support for XML Schema
- Support for Oracle XML DB repository events

This feature provides a set of simple, consistent, and standard-based APIs to traverse, query, access, and operate on data in Oracle XML DB when building mid-tier content-centric applications. The feature streamlines the development of internet applications and encourages customer uptake and partner integration on Oracle XML DB.

See Also:

Oracle XML DB Developer's Guide for details

1.6.5 XML Database

The following sections describe enhancements to the Oracle XML database technology.

1.6.5.1 DAV ACL Support

The DAV ACL standard is an industry standard for defining security in a WebDAV environment. Support for DAV ACL allows the XML DB repository to interact directly with other tools that understand the DAV ACL standard.

This feature enables interoperability between the Oracle XML DB repository and programs and tools that implement the DAV ACL standard. It enables the implementation of document-level security using industry standards, leading to reduced costs and increased productivity. Finally, it allows organizations to adopt authoring and security products that best meet their business needs.

See Also:

Oracle XML DB Developer's Guide for details

1.6.5.2 Enable the XDB HTTP Server for SOA

This feature allows Oracle Database to be treated as simply another service provider in a service-oriented architecture (SOA) environment. The database is now capable of exposing PL/SQL packages, procedures, and functions as Web services. The database can also execute dynamic XQuery and SQL queries.

These enhancements reduce complexity and cost by allowing the database HTTP server to expose native Web services in an SOA environment without requiring any additional Oracle or third-party software components. The feature provides performance enhancements over previous architectures in cases where Oracle mid-tier software is already providing SOA capabilities on top of an Oracle Database.

See Also:

Oracle XML DB Developer's Guide for details

1.6.5.3 Large Text Node Handling

In earlier releases, nodes with an Oracle XML DB document were limited to 64K of text. In most cases this was not a problem, because 64K translates to approximately 32 pages of text. However, when an XML document has non-XML documents embedded inside it, this limitation can be a significant problem.

For example, a text node might contain an embedded HTML or RTF document such as an abstract or Web page. Or an XML document might contain embedded binary content, such as JPEG images, Word Documents, or PDF files.

The XML standard does not allow an XML document to contain binary content, so the content of the embedded documents has to be encoded into a character-based format using algorithms such as base64, binHex, or UUENCODE. The resulting in text node can easily exceed 64K.

This feature allows Oracle XML DB to handle this class of XML document by eliminating the current 64K limit on the size of a text node and by providing a streaming API that makes it possible to manipulate nodes that contain large amounts of text in a memory-efficient manner.

See Also:

Oracle XML DB Developer's Guide for details

1.6.5.4 Oracle XML DB Repository Triggers

This feature provides a series of events related to repository operations. Developers can attach code to these events in the same way that they can attach code to database triggers. This code can then perform application logic based on repository events in the same way that triggers allow application code to be executed based on operations on database triggers.

This feature provides a framework for developing intelligent content management systems based on the Oracle XML DB repository. It allows businesses to reduce costs and ensure compliance by allowing business logic to be associated with document-centric applications that make use of the Oracle XML DB repository. Organizations can ensure that they remain in compliance with document and content management regulations in legislation such as Sarbens/Oxley.

See Also:

Oracle XML DB Developer's Guide for details

1.6.5.5 Recursive Schema Handling

This feature allows Oracle XML DB to deliver improved performance execution of XPath-based operations on XML documents that contain highly recursive structures. It enables XPath Rewrite in cases where all of the predicates operate on nodes that can occur at an arbitrary depth within the recursive structure.

The ability to easily describe and manipulate highly recursive structures is a key benefit of the XML paradigm. This project results in significant performance improvements when XML schema-based documents of this nature are stored and manipulated using Oracle XML DB.

See Also:

Oracle XML DB Developer's Guide for details

1.6.5.6 Repository Performance Improvements

Performance is improved in the Oracle XML DB repository for complex folder-restricted queries.

This feature improves the response time for certain kinds of SQL operations on the Oracle XML DB repository.

1.6.5.7 Scalable XSL Output

Until now, the XSLT capabilities of XML DB could not be used to generate HTML or Text formatted documents. This feature allows XSLT processing to write directly to a `CLOB` or Output stream, thus removing this limitation and allowing the XML DB XSLT capabilities to be used in a much wider set of cases without requiring any additional infrastructure.

This feature helps organizations to reduce cost and complexity by generating large HTML and Text formatted documents directly from the database without requiring additional software components.

1.6.5.8 SQL/XML Standards Compliance and Performance Optimizations

This feature implements the `XMLExists` and `XMLCast` operators, which are part of the SQL:2005 standard. In addition, this feature:

- Adds support for the `*` operator with `XMLForest(*)`.
- Adds support for complex storage parameters in the `STORE AS CLOB` clause of `XMLType` column DLL.
- Adds enhancements to the `XMLAgg` operator. The `XMLAgg` operator is used very commonly when generating XML from relation tables, or reconstructing XML from the internal object representation. `XMLAgg` is required whenever an element occurs multiple times within the document being generated. The proposed optimizations result in a significant reduction in the resources required to aggregate the target nodes, leading to noticeable performance improvements for these operations.

This feature provides significant performance improvements related to common XML generation operations. Reduced coding and maintenance costs result from a reduction in the amount of code required to perform common XML-centric tasks. Better support for relevant SQL standards enable the development of more portable application code.

See Also:

Oracle XML DB Developer's Guide for details

1.6.5.9 Support for More Types of Links in XDB

This feature adds support for a variety of links to Oracle XML DB, for example:

- Symbolic links to local XDB paths
- URL links

- Weak links (these store the OID, as do hard links, but do not affect persistence)
- Fixed links (these links can never be updated or deleted once created except by DDL operations)
- Pathname cache for fixed links
- Links as attributes
- The ability to globally disallow hard links to folders
- Preferred links

This variety of links enhances the XML Content Repository.

See Also:

Oracle XML DB Developer's Guide for details

1.6.5.10 XLink Support in XML DB

Oracle Database 11*g* provides the XML equivalent of referential integrity between XML documents using well understood W3C standards. XBRL validation, XLink validation, and compound document support are all enabled.

This feature allows organizations to use the W3C XLink recommendation to define additional rules about the relationships between documents. XLink is a popular W3C standard that allows integrity rules and other relationships be specified in an XML-centric manner. The XLink specification is used by a number of important XML standards, including XBRL, the eXtensible Business Reporting Language, which is being adopted by a number of regulatory agencies worldwide, including the FDIC and SEC in the USA.

Support of XLink also offers organizations additional flexibility on how XML documents are stored in the database. For example, they can now store a Book as a set of Chapters, providing more granularity and flexibility.

An XLink-aware repository ensures that the rules specified in the XLink language are enforced while the content is stored in the Oracle XML DB repository, making the use of the XML language a more compelling story.

See Also:

Oracle XML DB Developer's Guide for details

1.6.5.11 XML DB Operational Completeness

Oracle Streams now enables the propagation and management of data, transactions, and events in a data stream either within a database or from one database to another. The stream routes published information to subscribed destinations. The result provides greater functionality and flexibility than traditional solutions for capturing and managing events and for sharing the events with other databases and applications. As users' needs change, they can implement a new capability of Oracle Streams, without sacrificing existing capabilities.

Oracle Streams and Logical Standby can now be used in conjunction with the `XMLType` data type.

Oracle Streams delivers a common infrastructure that can meet the complex information sharing needs of today's organizations. These capabilities are now being extended so that they can be used in conjunction with all of the storage paradigms supported by the Oracle `XMLType` data type.

1.6.5.12 XML Query Optimization with Schemaless or Weakly-Typed Schema-Based XML

This feature enables cost-based optimizations of XQuery and XPath operations in cases where the underlying storage of some or all of the XML content is `CLOB` or `CSX` (binary XML). This results in significant performance improvements for queries over this kind of structure.

This feature allows organizations that cannot or do not want to leverage the schema-based storage model to improve the performance of their applications.

1.6.5.13 XML Translation Support

Operations on some classes of XML documents are now language-aware. As a result, the information returned by an operation on an XML document will be based on the users language preferences. This enhancement can significantly reduce the amount of code required to develop applications that are independent of a users language preferences. Until now, this capability was available only for documents stored using the Binary XML storage model.

This feature reduces costs associated with developing applications that are independent of the target users preferred language.

See Also:

Oracle XML DB Developer's Guide for details

1.6.5.14 XML Update Performance Optimizations

This feature provides significant optimizations in the way in which `UPDATE`, `DELETE`, and `INSERT` operations on existing documents are implemented. The changes, which are totally transparent to the application, result in significant reductions on the amount of code that is executed internally and the amount of I/O and log that is generated when the structure of an existing document is updated.

The performance of update operations on XML content stored in the database is greatly improved.

See Also:

Oracle XML DB Developer's Guide for details

1.6.5.15 XMLIndex

`XMLIndex` addresses the serious limitations inherent in the design and operation of the `CTXSYS.CTXXPATH` index. Among its benefits, `XMLIndex`:

- Does not require any prior knowledge of shape or content of the XML.
- Does not require any particular storage model for the `XMLType`.
- Can resolve any Xpath expression without requiring a subsequent filter of the result set identified by the index.
- Is transactional.
- Can support date and numeric range searching.
- Can support text-based searching of textnodes and attribute values through the `ora:contains()` XPath extension function.

This feature allows an organization to efficiently index XML documents without requiring explicit prior knowledge of shape or content of the XML that will be

indexed. It provides full support for extremely efficient XPath-based searching of the indexed XML content.

See Also:

Oracle XML DB Developer's Guide for details

1.6.5.16 XQuery and SQL/XML Performance Improvements

This feature implements performance improvements in a number of areas for SQL/XML and XQuery in the database.

XQuery and SQL/XML enhancements provide customers with performance gains when they upgrade to the current release.

See Also:

Oracle XML DB Developer's Guide for details

1.6.5.17 XQuery Standards Conformance

Oracle XQuery is in compliance with the final version of the W3C XQuery Standards and the JSR 225 XQJ standard.

Oracle XQuery standards compliance allows customers to build portable applications based on industry standards.

See Also:

Oracle XML DB Developer's Guide for details

1.6.5.18 XSLT Performance Improvements

This feature optimizes some classes of XSLT transformation by rewriting the operation into an equivalent XQuery expression and then executing the XQuery. The XQuery then becomes the subject of XQuery optimizations, which can result in set-based rather than iterative execution. This in turn can improve performance significantly.

XSLT transformation is commonly used to generate one format of XML from another format of XML. This feature provides significant performance improvements for some classes of XML to XML transformation.

See Also:

Oracle XML DB Developer's Guide for details

1.7 Fault Diagnosability

The following sections describe diagnosability features for Oracle Database 11*g* Release 1 (11.1).

1.7.1 Ease Diagnosis

The following sections describe new features for automatic information generation, packaging for problems or errors, and tools to ease diagnosis.

1.7.1.1 Automatic Inter-Component Deadlock Detection

Automatic inter-component deadlock detection automatically executes deadlock detection when such conditions are created or persist in the database.

This feature also provides the necessary infrastructure for other features in Oracle Database 11*g* that minimize database and application upgrade downtime.

See Also:

Oracle Database Performance Tuning Guide for details

1.7.1.2 Hang Manager

The Hang Manager automatically detects, analyzes, and dumps diagnostic information for hangs in Oracle Database environments, including Oracle RAC and ASM databases.

This feature simplifies database management by enabling you to more quickly resolve database hangs.

See Also:

Oracle Database Concepts for details

1.7.1.3 Incident Packaging Service (IPS)

Providing appropriate information to Oracle Support or Development is a tedious and sometimes time-consuming task for users. IPS provides a facility that extracts diagnostic and test case data associated with product exceptions (incidents) from Automatic Diagnostic Repository (ADR) and packages the data for transport to Oracle.

IPS also provides mechanisms to automatically generate test cases, such as SQL test cases, so that support and development personnel can easily reproduce the problems in-house for analysis and resolution.

See Also:

Oracle Database Administrator's Guide for details

1.7.1.4 SQL Test Case Builder

SQL Test Case Builder is a PL/SQL package that gathers all of the information at the customer site that is needed to reproduce a problem on a different system.

Obtaining a reproducible test case is the single most important factor in the speed of incident resolution. The SQL Test Case Builder gathers as much information as possible related to a SQL incident and packages it in a way that allows the problem to be reproduced.

See Also:

Oracle Database Performance Tuning Guide for details

1.7.2 First-Failure Capture

The following sections describe complete and efficient capture of diagnostic information during failures.

1.7.2.1 Automatic Diagnostic Repository

Automatic Diagnostic Repository (ADR) is a new system-managed repository for storing and organizing trace files and other error diagnostic data. ADR provides a comprehensive view of all the serious errors encountered by the database and maintains all relevant data needed for problem diagnostics and their eventual resolution.

Automatic Diagnostic Repository (ADR) provides a uniform and consistent mechanism to store, format, and locate all database diagnostic information. Customers can now correlate errors across various components such as Oracle RAC, Oracle Clusterware, OCI, Net, processes, and so forth. ADR automatically generates incidents for serious errors and provides incident management functionality. As a result, ADR significantly reduces time to problem resolution time for customers.

See Also:

Oracle Database Administrator's Guide for details

1.7.2.2 Enhanced Network Diagnosability

This feature enhances network diagnosability by consolidating Oracle Net diagnostics for various clients with the database infrastructure and presenting the diagnostic information in a standardized readable format. The network diagnostics information for Oracle products such as clients, application server, and database is stored in the Automatic Diagnostic Repository (ADR).

Oracle Net can now use the same diagnostic infrastructure as Oracle Database. This feature helps users diagnose Oracle Net related problems more easily than in earlier releases.

See Also:

Oracle Database Net Services Administrator's Guide for details

1.7.2.3 OCI Diagnosability

The following enhancements have been added in Oracle Database 11*g* to ease problem diagnosis in OCI applications:

- Record appropriate application, process, network, or other state information and dump it to log files when necessary.
- Add correlation information between client and server to diagnose problems easily.

The support for enhanced diagnosability mechanisms in OCI results in faster resolution of customer problems.

See Also:

Oracle Call Interface Programmer's Guide for details

1.7.3 Intelligent Resolution

These features describe the system-guided repair and resolution of problems.

1.7.3.1 Improved Block Corruption Detection

In earlier releases, RMAN-detected block corruptions were recorded in `V$DATABASE_BLOCK_CORRUPTION`. In Oracle Database 11*g*, live block corruptions are also recorded in that view. This view is automatically updated when corruptions are repaired; for example, using block media recovery or data file recovery.

This feature shortens the time to discover block corruptions.

See Also:

Oracle Database Backup and Recovery User's Guide for details

1.7.3.2 SQL Repair Advisor

When a problematic SQL statement is identified, the SQL Repair Advisor reloads the SQL and recompiles it in a more detailed diagnostic mode. The SQL Repair Advisor gathers more information on the incident and provides a workaround or recommendations to the DBA.

The SQL Repair Advisor is a new tool that helps DBAs diagnose problematic SQL statements and automatically provides workarounds for them.

See Also:

Oracle Database Administrator's Guide for details

1.7.4 Problem Prevention

The following sections describe features that are proactive in early problem detection to prevent catastrophes and limit damage to your system.

1.7.4.1 Automatic Health Monitoring

Automatic Health Monitoring proactively checks the health of the database and identifies any issues affecting the database. Problems are detected even before users run into them and before they cause widespread damage. The results of Automatic Health Monitoring are reflected in a simple meter (the Health Meter) designed to allow the DBA to quickly gauge the health of the database. With the help of Automatic Health Monitoring, a DBA can get a comprehensive picture of the current health of the database as well as help on how to remedy the problems affecting the database.

Automatic Health Monitoring proactively detects problems early in their life-cycle and notifies the DBA of the user impact and recommended action to take. Depending on the component or situation, Automatic Health Monitoring can either quarantine the offending component/resource or provide a fix to a problem. This feature helps businesses minimize their downtime and plan ahead for outages.

See Also:

Oracle Database Administrator's Guide for details

1.7.4.2 Automatic Quarantine of Corrupt Undo Segments

This feature automatically quarantines corrupt undo segments and prevents future transactions from using the same undo segments.

This enhancement limits corruption propagation and thus limits the damage caused on the system.

1.7.4.3 Fast ANALYZE: Finding Table-Index Corruptions

`ANALYZE VALIDATE CASCADE` operations are extremely expensive for large indexes. This feature introduces a hashing scheme for validating mismatch conditions.

The hashing is significantly faster than traditional cascade validation. The hash is designed so that both full table scans and index fast full scans produce the same hash value.

Many customers use `ANALYZE TABLE ... VALIDATE STRUCTURE CASCADE` statements to detect incompatibilities or corruptions between tables and indexes. `ANALYZE` operations in earlier releases took a long time for large tables and often could not complete in the diagnostics or maintenance window. With this feature, these `ANALYZE` operations now run much faster.

See Also:

Oracle Database Administrator's Guide for details

1.7.4.4 Holistic Undo Health Monitoring

This feature performs a number of health checks to determine the holistic health of the transactional engine. It proactively examines data and index blocks for transactional consistency and validates undo segments for possible corruption. Any corruption detected results in a proactive dump of all the redo related to the block.

This feature protects user data from software and hardware malfunction and helps prevent corruption propagation.

1.7.5 Support Workbench

The following sections describe features for Enterprise Manager's GUI workbench for customers and support to ease diagnosis and resolution of database errors.

1.7.5.1 Support Workbench

The Support Workbench provides an easy-to-use interface that presents database health-related incidents on the system to the DBAs in a timely manner along with information on how to manage the incidents. It also assists DBAs in viewing diagnostic information from multiple Oracle products (such as Net, clients, ASM, Oracle RAC, etc.), running health checks, packaging incident data to Oracle Support, and managing incidents.

The Support Workbench significantly reduces problem resolution time for customers by providing a simple workflow interface to view and diagnose incident data and package it for Oracle Support.

See Also:

Oracle Database 2 Day DBA for details

1.8 Information Integration

The following sections describe information integration features for Oracle Database 11*g* Release 1 (11.1).

1.8.1 Information Grid: Computation at Scale

The following sections describe support for:

- 2007-scale computation
- 2007 data center with very many nodes and databases
- Extremely large deployments without scaling costs linearly as the number of participants grows
- Extremely large data volumes and rapid provisioning of them
- Continuous Query Notification enhancements

1.8.1.1 Event Notification Grouping and Management

Event notifications can be grouped by a time interval, enabling a single batch notification for the group rather than multiple individual event notifications. Notifications may be retained for user-specified durations if the client is unavailable.

Notification configuration and management is more flexible.

See Also:

Oracle Streams Advanced Queuing User's Guide for details

1.8.1.2 Oracle Scheduler for Streams Jobs

Jobs scheduled for propagation and notification for Streams are now managed with the `DBMS_SCHEDULER` package.

This conversion to the standard Oracle Scheduler simplifies administration of Streams.

See Also:

Oracle Streams Concepts and Administration for details

1.8.1.3 Scalability for Streams Notifications

The notification server (EMON) is now a multiprocess server, consisting of 5 processes.

This feature enables large numbers of simultaneous notifications system-wide.

See Also:

Oracle Streams Advanced Queuing User's Guide for details

1.8.2 Information Grid: Heterogeneous Information

New features in the following sections describe:

- Virtualizing and provisioning information from many sources.
- Supporting many representations of information.
- Providing synchronous, replica management and change notification for many sources.
- Extending events to include implicit events in non-Oracle Database systems.

1.8.2.1 Messaging Gateway Service-Based Failover

Messaging Gateway agents can now be started by the Oracle Scheduler. The scheduler uses the Oracle RAC service feature, through which users can specify a primary instance and associated machine to host the Messaging Gateway agent. When the primary instance fails, the service fails over to another instance. The Messaging Gateway is started on the machine hosting that instance and connects to the instance on that node.

With this feature, you can now use Oracle RAC services to control where the Messaging Gateway will fail over, allowing for high availability without the need to make every database instance ready to run the Messaging Gateway agent.

See Also:

Oracle Streams Advanced Queuing User's Guide for details

1.8.2.2 Multiple Messaging Gateway Agents in Oracle RAC Environments

This feature provides support for running multiple Messaging Gateway agents on a single database. Users can now scale the messaging workload with the number of instances, by balancing the workload across multiple Oracle RAC instances.

With this feature, you can now build messaging solutions with greater throughput that scales across multiple Oracle RAC instances.

See Also:

Oracle Streams Advanced Queuing User's Guide for details

1.8.2.3 Simplified Messaging Gateway Propagation Job Configuration

A new enhanced PL/SQL API consolidates the propagation subscriber and the propagation schedule into a new propagation job.

It is now easier and less confusing to create and schedule a propagation job for a Messaging Gateway. A new API provides a more intuitive means for configuring these jobs.

See Also:

Oracle Streams Advanced Queuing User's Guide for details

1.8.2.4 Database Gateways: Performance Improvements

With this feature, you experience significant improvement when loading large volumes of data from the non-Oracle systems. Performance of the gateways also improves in general.

This feature improves the performance of Database Gateways.

See Also:

Oracle Database Heterogeneous Connectivity Administrator's Guide for details

1.8.2.5 Database Gateway for Adabas

Oracle Database 11*g* now provides a Database Gateway for Adabas.

Database Gateway for Adabas allows you to integrate data from Adabas using standard Oracle SQL.

1.8.2.6 Database Gateway for IMS

Oracle Database 11*g* provides a Database Gateway for IMS.

Database Gateway for IMS enables a customer to integrate data from IMS using standard Oracle SQL.

1.8.2.7 Database Gateway for VSAM

Oracle Database 11*g* provides a Database Gateway for VSAM.

Database Gateway for VSAM allows a customer to integrate data from VSAM using standard Oracle SQL.

1.8.3 More Granular Change Notifications

The following section describes Continuous Query Notification enhancements.

1.8.3.1 Continuous Query Notification Enhancements

Continuous Query Notification has been enhanced to provide notifications when the result set for a query changes on the server. These notifications can be registered at the

row-level, rather than just at the table-level. Thus, alerts will only be sent when changes to the specific rows selected have been modified.

Applications, such as Radio Frequency Identification (RFID), can now receive notifications directly from the database rather than build this functionality into the application.

See Also:

Oracle Database Advanced Application Developer's Guide for details

1.8.4 Streams Support for XMLType and TDE

The following sections describe support for:

- Oracle Streams data types
- Logical Standby
- Transparent Data Encryption (TDE)

1.8.4.1 Additional Data Type Support for Streams

Streams provides support the `XMLType` data type and for Transparent Data Encryption (TDE).

Data can be shared more easily between applications and databases.

See Also:

Oracle Streams Concepts and Administration for details

1.9 Location Services and Specialty Data

The new features in the following sections describe:

- Extending Oracle leadership in managing spatial and multimedia content.
- Addressing the unique performance, scalability and application development needs of rapidly growing machine-generated data created by sensors, RFID, semantic web, medical, life sciences, satellite imaging, and remote sensing technologies.

Note: The name Oracle *inter*Media has been changed to Oracle Multimedia in Oracle Database 11*g* Release 1 (11.1). The feature remains the same, only the name has changed. References to Oracle *inter*Media will be replaced with Oracle Multimedia, however some references to Oracle *inter*Media or *inter*Media may still appear in graphical user interfaces, code examples, and related documents in Oracle Database documentation library for 11*g* Release 1 (11.1).

1.9.1 New Capabilities for Management of Geospatial and Multimedia Data

This release further enhances industry-leading spatial and multimedia features used in internet, geospatial, and media-oriented applications. New Web services, XML services, and usability features support mainstream business applications.

Enhancements for three-dimensional data, very large network data models, and GeoRaster data models support business-critical applications in government,

homeland security and defense, life sciences, utilities, energy exploration, transportation, urban planning, simulation, and geoengineering.

1.9.1.1 Spatial: 3-D Geometry, Surface, and Point Cloud Storage and Indexing Functions

Oracle Spatial provides native storage, querying, and retrieval for three-dimensional (3-D) data including points, lines, surfaces, triangulated irregular networks (TINs - an alternative to rasters), and point clouds. Spatial R-tree indexing now supports 3-D data; SQL operators and analysis functions for 3-D data are also provided.

Oracle Spatial supports very large 3-D data sets such as urban models, point clouds, and terrain models with open, secure, scalable, and improved data management for urban planning and design, government, homeland security, military, oil and gas exploration, transportation engineering, gaming and simulation, geoengineering, and LIDAR-based map production.

See Also:

Oracle Spatial Developer's Guide for details

1.9.1.2 Spatial Web Services

Oracle Web Services now support the following XML-based geospatial Web Services standards on a variety of client technologies and platforms:

- OGC OpenLS 1.1
- Web Feature Service 1.0
- Web Feature Service - Transactional 1.0
- Catalogue Service 2.0

Oracle Database and Oracle Application Server provides security for these Web services standards including authorization, authentication, and transport confidentiality and integrity. Java and PL/SQL client APIs are provided.

Oracle Spatial Web Services are location-ready and tightly integrated with Oracle Database and Oracle Application Server to provide a secure, transactional, service-oriented architecture platform to access and incorporate any published services for routing, geocoding, directory, catalog, feature, and mapping.

See Also:

Oracle Spatial Developer's Guide for details

1.9.1.3 Spatial Routing Engine Enhancements

The Oracle Spatial routing engine now generates driving directions in German, French, Spanish, and Italian, and in turn-specific geometries.

The Oracle Spatial routing engine provides driving directions in various Western European languages and turning diagrams for linear asset management and travel and transport applications.

See Also:

Oracle Spatial Developer's Guide for details

1.9.1.4 Spatial Network Data Model Enhancements

The Oracle Spatial network data model supports multiple concurrent Application Server requests with its thread-safe Java API and supports an XML interface for Web services queries.

New analysis and modeling capabilities include database-level handling of user or application-specific attributes. These features enable the network data model to manage nonconnectivity-related information. You can also selectively extract a subset of the network using SQL-like filters. Other enhancements include path arithmetic support such as path addition, subtraction, intersection, comparison, and duration modeling.

The Oracle Spatial network data model provides Java and XML APIs for Service Oriented Architectures and high query volume applications such as field service scheduling and logistics.

New analysis and modeling capabilities are provided for applications that analyze very large networks with complex attributes and relationships such as utilities, logistics, transportation, life sciences, and web information management.

See Also:

Oracle Spatial Topology and Network Data Models Developer's Guide for details

1.9.1.5 Load-On-Demand for Very Large Networks in Oracle Spatial Network Data Model

Oracle Spatial now supports load-on-demand for partitioned networks, which eliminates physical memory as a limitation for in-memory analysis. Large networks can be partitioned into manageable subnetworks and incrementally loaded into memory as needed for performance and scalable analysis. Partitioning utilities are available for partitioning large spatial networks.

Oracle Spatial network data model can analyze very large networks without loading the entire network into memory. This ability benefits applications such as utility and street networks.

See Also:

Oracle Spatial Topology and Network Data Models Developer's Guide for details

1.9.1.6 Spatial GeoRaster: Manageability, Reliability, Usability Enhancements

GeoRaster DML triggers are created and monitored by the system automatically. New tools automate GeoRaster upgrade, migration, and use. GeoRaster has support for Workspace Manager long transactions and fine-grained Label Security.

More metadata and data types are now supported. GeoRaster supports a generic polynomial georeferencing model, which includes direct linear transportation and rapid positioning capability georeferencing for nonrectified airborne photos and satellite images. It also supports bitmap masks, multiple `NODATA` values and value ranges, and has limited data type support.

Advanced operations have been added for advanced mosaic, GeoRaster object or layer union, GeoRaster templates, and related functions. GeoRaster also fully supports load and export of GeoTIFF geometadata and DigitalGlobe RPC file format.

Additional raster processing capabilities include GeoTIFF geometadata, loading, and exporting.

Oracle Spatial GeoRaster supports more data types and file formats, editing of image and grid data with long transaction versioning, fine-grained security, and new management tools and functionality.

See Also:

Oracle Spatial GeoRaster Developer's Guide for details

1.9.1.7 Enhanced Workspace Manager Performance and Data Storage Options

User-specified optimizer hints on Workspace Manager operations are now supported. Partition operations (such as `ADD`, `DELETE`, `MERGE`, and `EXCHANGE`) are supported on a versioned table. New online operations on versioned tables include `ALTER INDEX` and `ALTER TABLE` operations, including moving a table to another tablespace and changing the percent free space. Row-level locking is implemented for merge and refresh operations. Compression is optional for the RemoveWorkspace operation.

This release of Workspace Manager supports greater tuning of queries for workspace operations and more physical storage options. Merge and refresh operations can be performed faster by executing them in parallel and making garbage collection optional if the workspace is removed.

See Also:

Oracle Database Workspace Manager Developer's Guide for details

1.9.1.8 Enhanced Workspace Manager Security and Administration

Oracle Label Security policies can now be set and altered on a versioned table. Changes to tables involved in a referential integrity constraint can be merged as a group. The name of the workspace that creates and retires a history row is recorded.

Workspace Manager allows Oracle Label Security policies to be applied and altered for long transactions. Changes to groups of tables can be merged as a set. Changes to data can be tracked by project.

See Also:

Oracle Database Workspace Manager Developer's Guide for details

1.9.1.9 Improved Workspace Manager Valid Time Usability

In this release, the unique `ROWID` for history rows with a Valid Time is available to query. Initial Valid Time timestamp values can be specified when a table is versioned. Valid Time timestamps can be presented as `TIMESTAMP WITH TIME ZONE` scalar type as an alternative to using the `WM_PERIOD` object type. Changes to Valid Time timestamps are now allowed for an Oracle Spatial feature stored in a parent table of a referential integrity constraint.

Valid Time, also known as effective dating, is a capability of Workspace Manager that allows versioned data to be stored with an associated time range that indicates when the data is valid. This feature includes several usability enhancements. A specific historical change with Valid Time can be presented to a user. Valid Time timestamp values can be initialized with user-defined values when a table is versioned and timestamps can be used as simple scalar values in a Microsoft .Net environment. Valid Time timestamps can be updated anytime an Oracle Spatial feature participates in a referential integrity constraint.

See Also:

Oracle Database Workspace Manager Developer's Guide for details

1.9.1.10 Oracle Multimedia Performance and Scalability

Earlier versions of Oracle *inter*Media (now called Oracle Multimedia) object types (`ORDImage`, `ORDAudio`, `ORDVideo`, and `ORDDoc`) allowed storage of electronic media objects of up to 4 Gbytes. In Oracle Database 11*g*, this media size limit is extended to the `BLOB` size limit, between 8 and 128 terabytes.

In addition, Oracle Multimedia uses less memory for scale-down and thumbnail generation functions, raising the practical limit for images to be processed in this manner.

Applications in the security, media and entertainment, medical, and life sciences markets are generating more detailed and higher fidelity digital media. Advances in capture technology and reductions in the cost of storage present the need to organize more and larger digital images, audio, and video. In the past, individual digital pictures, audio, and video rarely exceeded a gigabyte. With Oracle Database 11*g*, Oracle Multimedia supports emerging applications that may require tens of gigabytes for a single multimedia file.

See Also:

Oracle Multimedia User's Guide for details

1.9.2 New Capabilities for Medical Data

This following sections describe new features for the management of DICOM format medical data for health care and life sciences imagery.

1.9.2.1 Multimedia DICOM Medical Imaging Format Support

Oracle Multimedia now fully supports the format universally recognized as the standard for medical imaging, Digital Imaging and Communications in Medicine (DICOM) version 3. Applications can now use Multimedia JAVA and PL/SQL APIs to store, manage, and manipulate DICOM content.

It is now quite reasonable to build large archives of medical content such as single frame and multiframe images, waveforms, slices of 3-D volumes, video segments, and structured reports that are managed and secured using powerful Oracle Database tools. Complete DICOM metadata support makes it very easy to index and search DICOM content for research. Central storage of DICOM content makes telemedicine practical. Incorporating DICOM content in a database makes it feasible to build electronic health care records applications using powerful application development tools from Oracle or other third-party vendors.

See Also:

Oracle Multimedia DICOM Developer's Guide for details

1.9.2.2 ORDDicom Object Type

A new Multimedia object type, `ORDDicom`, has been defined to hold the data associated with DICOM content such as single or multiframe images, waveforms, or structured reports, and to implement the methods to manipulate the DICOM content. A Java proxy class, `OrdDicom`, is also defined. It provides access to the `ORDDicom` database object through JDBC in a Java application. For applications that already store DICOM content directly in `BLOB` or `BFILE`, a relational interface is also provided.

By presenting DICOM content stored in a database as objects, Oracle enables both rapid application development and easy, secure management of large medical archives.

See Also:

Oracle Multimedia DICOM Developer's Guide for details

1.9.2.3 DICOM Metadata Extraction

In the initial release of Oracle Multimedia DICOM support, the most important metadata tags associated with DICOM content could be extracted into an XML document that could then be indexed and searched to find DICOM content that matched certain conditions. With Oracle Database 11*g*, that support is extended to offer complete and extensible metadata extraction. Customers can use either the supplied DICOM mapping document to extract metadata into an XML document, or they can create their own mapping documents to include support for private tags or subsets of the standard DICOM metadata tags. The extracted metadata can then be stored in a table to facilitate DICOM content searching based on standard or private DICOM attributes.

With this significant enhancement to the DICOM metadata extraction ability, customers can build large archives of DICOM content that will prove invaluable for research. By customizing DICOM mapping documents, customers can easily create highly specialized indexes to DICOM content such as single and multiframe images, waveforms, and structured reports based on both standard and private metadata tags.

See Also:

Oracle Multimedia DICOM Developer's Guide for details

1.9.2.4 DICOM Conformance Validation

Given DICOM content and a set of user-specified conformance rules, Oracle Multimedia can verify that the DICOM content adheres to the conformance rules.

DICOM content is produced by many modalities. Most content conforms to the DICOM standard, but inevitably some does not. It is worthwhile to be able to identify DICOM content that does not conform to the standard. Validating DICOM content for conformance can ensure the consistency of a DICOM archive. It enables a database to accept DICOM content from multiple sources and to verify the integrity of the DICOM metadata.

See Also:

Oracle Multimedia DICOM Developer's Guide for details

1.9.2.5 DICOM Image Processing

Oracle Multimedia includes the `ORDDICOM` data type to natively support imagery produced by medical imaging devices. This release adds methods and functions to copy and convert images from DICOM to JPEG, GIF, PNG, TIFF, and other formats and to generate scaled versions and thumbnails.

In order to view medical images stored in the DICOM format in Web applications, the image must be reformatted and converted into a browser-ready format. Oracle Database 11*g* provides methods to reformat and deliver DICOM images to applications that require popular industry-standard image formats.

See Also:

Oracle Multimedia DICOM Developer's Guide for details

1.9.2.6 Making DICOM Content Anonymous

Oracle Database 11*g* offers a `makeAnonymous()` method which can be used to ensure that users of a DICOM medical archive see only the metadata that they are authorized to see.

Rules for making DICOM content anonymous can be customized in an anonymity document that specifies the set of attributes that should be made anonymous and how each should be obfuscated.

Clinicians need to be able to view all the metadata included in DICOM content for a patient being treated. Researchers need to be able to see the same DICOM content but, due to privacy regulations, must not be able to see any personally identifying information from that same DICOM content used.

By providing anonymity services in the database, Oracle Multimedia allows appropriate access for different classes of users of DICOM archives regardless of the application used to access the data.

See Also:

Oracle Multimedia DICOM Developer's Guide for details

1.9.2.7 Creation of DICOM Content

This release of Oracle Multimedia includes the ability to generate new DICOM content by combining digital images with an XML representation of the associated DICOM metadata. The result of this operation, well-formed and validated DICOM content, can be stored in a table in the database or delivered to a DICOM viewer.

Storing and retrieving paper- or film-based medical images is expensive and prone to error. However, many old images must be retained for a very long time and might be useful for both research and clinical purposes. By storing scanned images with the known metadata about those images in DICOM format, non-DICOM images become useful again and both storage and retrieval costs are reduced. New DICOM format content can also be generated to correct metadata errors in the original DICOM content using this technique.

See Also:

Oracle Multimedia DICOM Developer's Guide for details

1.9.2.8 Runtime Updatable DICOM Data Model

The runtime behavior of Oracle Multimedia DICOM support is determined by a set of user-configurable documents. This set of documents is collectively managed by a DICOM data model repository. An administrator can modify this data model repository to configure DICOM features for a particular database instance. With this design, a customer can upgrade DICOM features at any time, without interfering with an active DICOM archive.

Hospitals need their DICOM archives to be operational 24x7x365. They cannot afford to take down the system to update to a new version of the DICOM standard, to incorporate private tags for a new piece of equipment, to change their DICOM conformance rules, or to modify the set of tags they extract from DICOM content or the XML document they produce from the extracted tags. The data model repository implemented by this feature means that users can update their systems to accommodate new equipment and standards without interrupting the operation of the archive.

See Also:

Oracle Multimedia DICOM Developer's Guide for details

1.9.3 New Capabilities for RFID

This following section describes management of Sensor and RFID data.

1.9.3.1 RFID Identity Code Data Type for EPC, DoD, and Custom Sensor Tags

A new Identity Code object type is implemented in Oracle Database to support custom, Electronic Product Code (EPC), and Department of Defense standard tag formats. Metadata is used to define rules for encoding and decoding any tag information into Identity Code objects. All EPC standard tag types and metadata standards are supported.

With this new feature, application developers can define and process identity code tags faster and with less programming in Oracle Database 11*g*.

See Also:

Oracle Database Advanced Application Developer's Guide for details

1.9.4 New Capabilities for Semantic Data

This following sections describe new features and capabilities for semantic data.

1.9.4.1 Improved Performance for Bulk Loading

Oracle provides native storage, inference, and querying of semantic data sets often containing hundreds of millions of triples (modeled in canonical Resource Description Framework (RDF) <Subject Predicate Object> format). A new bulk loading utility is now introduced that significantly improves the ability to handle large volumes of triples. Query performance on semantic data has been improved, especially for queries returning large result sets, using a query rewrite technique and Oracle Database optimizer. Support for typed literals in semantic data has been enhanced to include `xsd:date` and `xsd:time`. Also, `xsd:dateTime` with time zone is now supported.

Oracle Database provides scalable, secure, integrated, and efficient support to store, inference, and query large semantic data sets described using W3C standards. Performance for loading and querying of semantic data has been improved to enable scaling for large data sets used in the defense and intelligence, life sciences, and geospatial domains. Date and time information, optionally with time zone, can be stored and queried.

See Also:

Oracle Database Semantic Technologies Developer's Guide for details

1.9.4.2 Support for Storage and Query of Semantic Content

Oracle Database 11*g* extends its semantic capabilities with native, lightweight Web Ontology Language (OWL) inferencing that is a practical subset of the OWL-DL standard. Ontologies (sets of terms, associated properties, and the relationships among them) can be stored in the database to enable searching based on relationships described in the ontology using new operators, `SEM_RELATED` and `SEM_DISTANCE`.

Inferencing support includes efficient and scalable reasoning for a subset of OWL-DL constructs and APIs to generate proofs for inferred triples and to detect inconsistency in semantic data sets. Advanced users can develop custom inferencing rules.

Semantic operators can be used for filtering based on semantic relatedness (SEM_RELATED), and the results can be further restricted or ordered using proximity measures (SEM_DISTANCE). A new index type (SEM_INDEXTYPE) allows efficient execution of such queries, enabling scalable performance over large data sets.

These new semantic operators enable Oracle Database to query relational data not only through keyword matching but also using concepts and terms related to the keyword. These ontology-assisted queries are based on semantic relationships between the column value data and ontology terms. This enables more complete search results without requiring as much prior knowledge of the data set. The OWL inferencing capability enables discovery of new relationships in RDF and OWL data. This is useful in applications in life sciences, health care, and business enterprise information integration. The W3C has adopted RDF, RDFS, and OWL as standards to represent semantic data.

See Also:

Oracle Database Semantic Technologies Developer's Guide for details

1.10 Manageability

The following sections describe server manageability features for Oracle Database 11*g* Release 1 (11.1).

1.10.1 Automatic Storage Management

The new features in Automatic Storage Management (ASM) extend the storage management automation, improve scalability, and further simplify management for Oracle Database files.

1.10.1.1 ASM Fast Mirror Resync

A new SQL statement, ALTER DISKGROUP ... DISK ONLINE, can be executed after a failed disk has been repaired. The command first brings the disk online for writes so that no new writes are missed. Subsequently, it initiates a copy of all extents marked as stale on a disk from their redundant copies.

This feature significantly reduces the time it takes to repair a failed diskgroup, potentially from hours to minutes. The repair time is proportional to the number of extents that have been written to or modified since the failure.

See Also:

Oracle Database Storage Administrator's Guide for details

1.10.1.2 ASM Manageability Enhancements

The new storage administration features for ASM manageability include the following:

- New attributes for disk group compatibility

  To enable some of the new ASM features, you can use two new disk group compatibility attributes, compatible.rdbms and compatible.asm. These attributes specify the minimum software version that is required to use disk groups for the database and for ASM, respectively. This feature enables heterogeneous environments with disk groups from both Oracle Database 10*g* and Oracle Database 11*g*. By default, both attributes are set to 10.1. You must advance these attributes to take advantage of the new features.

- New ASM command-line utility (ASMCMD) commands and options

ASMCMD allows ASM disk identification, disk bad block repair, and backup and restore operations in your ASM environment for faster recovery.

- ASM fast rebalance

  Rebalance operations that occur while a disk group is in `RESTRICTED` mode eliminate the lock and unlock extent map messaging between ASM instances in Oracle RAC environments, thus improving overall rebalance throughput.

This collection of ASM management features simplifies and automates storage management for Oracle databases.

See Also:

Oracle Database Storage Administrator's Guide for details

1.10.1.3 ASM Preferred Mirror Read

When ASM failure groups are defined, ASM can now read from the extent that is closest to it, rather than always reading the primary copy. A new initialization parameter, `ASM_PREFERRED_READ_FAILURE_GROUPS`, lets the ASM administrator specify a list of failure group names that contain the preferred read disks for each node in a cluster.

In an extended cluster configuration, reading from a local copy provides a great performance advantage. Every node can read from its local diskgroup (failure group), resulting in higher efficiency and performance and reduced network traffic.

See Also:

Oracle Database Storage Administrator's Guide for details

1.10.1.4 ASM Rolling Upgrade

Rolling upgrade is the ability of clustered software to function when one or more of the nodes in the cluster are at different software versions. The various versions of the software can still communicate with each other and provide a single system image. The rolling upgrade capability will be available when upgrading from Oracle Database 11*g* Release 1 (11.1).

This feature allows independent nodes of an ASM cluster to be migrated or patched without affecting the availability of the database. Rolling upgrade provides higher uptime and graceful migration to new releases.

See Also:

Oracle Real Application Clusters Installation Guide for Linux and UNIX for details

1.10.1.5 ASM Scalability and Performance Enhancements

This feature increases the maximum data file size that Oracle can support to 128 TB. ASM supports file sizes greater than 128 TB in any redundancy mode. This provides near unlimited capacity for future growth. The ASM file size limits are:

- External redundancy - 140 PB
- Normal redundancy - 42 PB
- High redundancy - 15 PB

Customers can also increase the allocation unit size for a disk group in powers of 2 up to 64 MB.

These improvements reduce database startup time and memory requirements, and allow support for larger ASM files, making it feasible to implement several hundred TB or even PB Oracle databases on ASM. Larger allocation units provide better sequential read performance.

See Also:

Oracle Database Storage Administrator's Guide for details

1.10.1.6 Convert Single-Instance ASM to Clustered ASM

This feature provides support within Enterprise Manager to convert a non-clustered ASM database to a clustered ASM database by implicitly configuring ASM on all nodes. It also extends the single-instance to Oracle RAC conversion utility to support standby databases.

Simplifying the conversion makes it easier for customers to migrate their databases and achieve the benefits of scalability and high availability provided by Oracle RAC.

See Also:

Your platform-specific Oracle Real Application Clusters installation and configuration guide for details

1.10.1.7 New SYSASM Privilege for ASM Administration

This feature introduces the new `SYSASM` privilege to allow for separation of database management and storage management responsibilities.

The `SYSASM` privilege allows an administrator to manage the disk groups that can be shared by multiple databases. The `SYSASM` privilege provides a clear separation of duties from the `SYSDBA` privilege.

See Also:

Oracle Database Storage Administrator's Guide for details

1.10.2 Change Assurance

The following sections describe new features for automatic capture and replay of workloads before and after changes to analyze impact.

1.10.2.1 Database Replay

Before making changes, such as hardware or software upgrades, companies typically conduct extensive testing to validate the changes. However, when the change is made on the production system, problems are often encountered because the testing was not performed on a realistic workload. Tools are available in the market to construct "synthetic workloads" involving multiple users. However, they are not able to adequately simulate the complexities of a real-world workload, such as interactions between concurrent activities and unpredictability of the workload profile.

The Database Replay feature addresses this need by enabling users to perform real-world testing by capturing the actual database workload on the production system and replaying it on the test system. It also provides analysis and reporting to highlight potential problems (for example, errors encountered and divergence in performance) and recommend ways to remedy the problems.

See Also:

Oracle Database Performance Tuning Guide for details

1.10.2.2 SQL Performance Analyzer

SQL performance regression is always a concern during system changes such as database upgrades, initialization parameter changes, and addition or dropping of indexes. The SQL Performance Analyzer feature alleviates this concern by providing an easy way to assess the impact of a change on the performance of SQL statements by comparing and contrasting their response times before and after the change. SQL Performance Analyzer allows you to capture the SQL workload from the source system, such as the production database, and to replay it on the test system where the change has been applied.

Any differences in response time of SQL statements, execution plan regressions, and so forth, are reported and precise recommendations on how to tune the under-performing SQL statements are provided.

See Also:

Oracle Database Performance Tuning Guide for details

1.10.3 Database Control

The following sections describe Enterprise Manager solutions that help ease database management.

1.10.3.1 Advanced Replication Cutover

In previous releases, Advanced Replication was a trigger-based method of replication. With Oracle Database 11*g*, this feature is a cutover of existing functionality to Web-based Enterprise Manager Database Control.

This feature provides support for existing Advanced Replication customers by providing the ability to manage Advanced Replication from 11*g* Enterprise Manager Database Control.

1.10.3.2 ASM Enhancements

This feature provides several enhanced ASM features for the Enterprise Manager user interface.

Enterprise Manager has been enhanced for ASM File Access Control, OCR and Voting Disk in ASM, Disk Resync, ASM Rolling Migrations, Disk Zones, ASM Manageability and Infrastructure, ACL Enhancements, and Security Classes for Fusion Security.

1.10.3.3 Change Manager - Sync and Propagate

This feature helps users deploy an application to multiple sites and to upgrade an earlier version of an application to the current version at multiple sites.

Users can now propagate metadata definitions, and some data, from a database or baseline to multiple database targets.

1.10.3.4 Database Cloning Enhancements

This feature provides the following improvements in Enterprise Manager database cloning pages:

- Removes the requirement for source and destination stage area.
- Supports cloning from a generic RMAN backup set.
- Enhances the job summary user interface.

These improvements facilitate database cloning from Enterprise Manager using new RMAN features.

1.10.3.5 Database Configuration (ECM) Collection Enhancements

This new feature provides enhancements to Database Configuration (ECM collections) maintenance.

The benefit is an improved configuration collection framework.

1.10.3.6 Database Home Page and Performance Page Improvements

This feature provides new charts, changes the layout, includes more details, and makes other improvements to the Database Home page and Performance page.

Enhancements to the Database Home page and Performance page lead to a better utility and ease-of-use.

See Also:

Oracle Database 2 Day + Performance Tuning Guide for details

1.10.3.7 DBCA Enhancements

The following enhancements have been added to DBCA:

- Added support to configure new database options

  The following options in Oracle Database 11*g* can be configured using Database Configuration Assistant:

  - Oracle Application Express
  - Oracle Database Vault
  - Oracle Warehouse Builder

- Automatic memory management

  This is a new initialization parameter in Oracle Database 11*g* to automate the memory allocation. By default, Database Configuration Assistant now uses `MEMORY_TARGET` instead of specifying individual values for `SGA_TARGET` and `PGA_AGGREGATE_TARGET`. The memory management page of Database Configuration Assistant has new option to select automatic memory management.

- Oracle Data Mining

  In Oracle Database 11*g*, the data mining schema are created when you run the `catproc.sql` script as the `SYS` user. You no longer configure this option through the Database Features screen of Database Configuration Assistant.

- `ORACLE_BASE` and Diagnostic Destination configuration

  The directory that you specify when you are prompted for `ORACLE_BASE` by Oracle Universal Installer is stored in the Oracle home inventory. Database Configuration Assistant uses this value to derive the default database locations and the `DIAGNOSTIC_DEST` parameter. The diagnostic destination location contains all ADR directories (diagnostic files such as Alert logs and so on). Starting with Oracle Database 11*g* Release 1 (11.1), the initialization parameter settings for background dump, user dump, and core dump destinations are replaced by the Diagnostic Destination.

- Secure database configuration

Oracle Database 11*g* has new defaults for audit and password profiles. Database Configuration Assistant has a new screen to enable the new security settings during the database creation and existing database configuration.

- Switching a database from Database Control to Grid Control configuration

  In previous releases, Database Configuration Assistant contained functionality to configure a database either with Database Control or with Grid Control. You could configure a database either while creating it or later. However, reconfiguring a database from Database Control to Grid Control required significant manual effort. With Oracle Database 11*g*, Database Configuration Assistant provides the Enterprise Manager Configuration plug-in, which automates the process to switch configuration of a database from Database Control to Grid Control.

1.10.3.8 DBUA Enhancements

The following enhancements have been added to DBUA:

- Express Edition upgrade

  For single-instance databases, Oracle Database Upgrade Assistant configuration utility enables you to upgrade from Oracle Database Express Edition (Oracle Database XE) to Oracle Database 11*g*. The XE database files reside under the path `ORACLE_BASE/oradata/XE`. These files must be copied to a new location as the user may remove the XE Home after upgrade.

- Moving datafiles into ASM, SAN, and other file systems

  You can move datafiles to ASM, OFS, or other storage devices, such as Storage Area Networks (SAN) and Network Area Storage (NAS), as part of the upgrade. If you move the database files during the upgrade, you can benefit from the typical downtime for this tablespace by rebalancing disks and moving files to a better storage device, such as SAN, NAS, or ASM.

- `ORACLE_BASE` and Diagnostic Destination configuration

  The directory that you specify when you are prompted for `ORACLE_BASE` by Oracle Universal Installer is stored in the Oracle home inventory. Database Upgrade Assistant uses this value to derive the default database locations and the `DIAGNOSTIC_DEST` parameter. The diagnostic destination location contains all ADR directories (diagnostic files such as Alert logs and so on). This diagnostic destination directory is required while upgrading an earlier Oracle Database release to Oracle Database 11*g* release of the database. If the `ORACLE_BASE` directory already exists, Oracle Database Upgrade Assistant automatically retrieves this information and populates its path. Starting with Oracle Database 11*g* Release 1 (11.1), the initialization parameter settings for background dump, user dump, and core dump destinations are replaced by the Diagnostic Destination.

- Command-line option to `AUTOEXTEND` system files

  The command line option `AUTOEXTEND` facilitates auto extending of the data files as a part of the upgrade. This option autoextends the data files during the upgrade and turns the autoextend back to its original settings after the upgrade. This option is useful if there is enough room on the disk, and if you do not need to add new datafiles or manually increase the size of the files.

These features allow XE customers to upgrade to Standard Edition or Express Edition database for support and scalability.

These features also provide the ability to move to better storage devices as part of upgrade. You can move datafiles to be in compliance with OFA standards.

1.10.3.9 Enhanced Adaptive Metric Thresholds

The Adaptive Metric Thresholds feature simplifies and improves the selection of alert thresholds for database performance metrics. Key enhancements include:

- Full integration with AWR baselines as the source for metric statistics.
- Quick configuration option offers one-click starter set of thresholds based on OLTP or Data Warehouse system profiles.
- Automated time group selection for `SYSTEM` moving window baseline.
- Guided discovery of candidate alert metrics based on response levels to known problem.
- Improved metric charting and analysis.

This feature helps DBAs configure high quality, baseline-driven performance alert thresholds with minimal effort.

1.10.3.10 Fusion Requirements

This feature fulfills the requirements for Fusion in Enterprise Manager, including:

- Group copy of `tnsnames.ora`
- Monitor and validate `init.ora`
- Monitor database session trace options
- View database trace files
- Archive and purge trace and alert files

This feature allows Oracle Application Server to use Enterprise Manager for many monitoring and configuration-related tasks.

1.10.3.11 Management of Oracle Text Indexes

This feature provides a user interface to support management of Oracle Text indexes.

Oracle Enterprise Manager provides an interface for configuration, maintenance, and administration of Oracle Text, including:

- Viewing details of Oracle Text indexes
- Synchronizing Oracle Text indexes
- Optimizing Oracle Text indexes
- Rebuilding indexes
- Editing Oracle Text indexes
- Querying log analysis

See Also:

Oracle Text Application Developer's Guide for details

1.10.3.12 Migrate Database to ASM - Enhancements in Enterprise Manager

The Enterprise Manager pages for Migrate Database to ASM have been enhanced to include notification when a job is suspended and support for tablespace level migration.

This feature makes it easier to use Enterprise Manager to migrate non-ASM databases to ASM.

1.10.3.13 Storage / Audit Report and Metric Enhancement

Enterprise Manager has been enhanced to provide better storage and audit reports and metrics.

This feature facilitates monitoring in the storage area and audit area.

1.10.3.14 Storage: Schema, Security, and Configuration Enhancements

Enterprise Manager has been enhanced in the following areas: export, import, Data Pump, storage administration, schema administration, OLAP, summary management, and partition maintenance.

This feature improves Enterprise Manager support in several areas.

1.10.3.15 Wait Activity Detail Enhancement

Until now, Oracle Enterprise Manager displayed the Wait Activity Drilldown detail in a chart format.

This feature provides more detailed Wait Activity information for the client, service, module, and action pages in Oracle Enterprise Manager.

See Also:

Oracle Database 2 Day + Performance Tuning Guide for details

1.10.3.16 Workspace Manager

The Oracle Enterprise Manager interface for the Workspace Manager features has been enhanced to include support for compress workspace, manage system parameters, privilege management, `FindRICSet`, and others.

This feature facilitates the administration of Workspace Manager from Oracle Enterprise Manager.

1.10.4 Holistic Database Management

The following sections describe database-wide, holistic, consolidation management and control enhancements.

1.10.4.1 ADDM for Oracle Real Application Clusters

ADDM has been enhanced to provide comprehensive cluster-wide performance diagnostic and tuning advice. A special mode of ADDM analyzes an Oracle RAC database and reports on issues that are affecting the entire cluster as well as those that are affecting individual instances.

This feature is particularly helpful in tuning global resources such as I/O and interconnect traffic and makes the tuning of Oracle RAC databases easier and more precise.

See Also:

Oracle Database 2 Day + Performance Tuning Guide for details

1.10.5 Intelligent Infrastructure

This section describes Statistics, Alerts, Advisor and Automated Maintenance Tasks infrastructure that enable self-management.

1.10.5.1 Auto-Task

In Oracle Database 11*g*, Auto-Task can run hundreds or thousands of jobs in the manageability windows. This feature adds the necessary scheduler infrastructure to automatically manage this very high manageability job load.

This feature provides Auto-Task the necessary infrastructure to handle large numbers of jobs.

See Also:

Oracle Database Administrator's Guide for details

1.10.5.2 Automatic Maintenance Tasks Management

This feature provides out-of-the-box management of resource distribution (CPU and I/O) among the various database maintenance tasks such as Automatic Optimizer Statistics Collection, Automatic Segment Advisor, and others. The CPU is automatically managed. I/O is managed only if the I/O Resource Manager is enabled.

This feature ensures that work during maintenance operations is not affected and that user activity gets the necessary resources to complete.

See Also:

Oracle Database Administrator's Guide for details

1.10.5.3 Automatic Workload Repository (AWR) Baselines

Automatic Workload Repository (AWR) baselines enable accurate performance comparison by providing the ability to tag a certain period of time as a period of interest against which comparisons can be made at some time in the future. All the related performance statistics for this period are maintained and are available for use for any performance analysis with another time period.

AWR baselines provide powerful capabilities for defining dynamic and future baselines and considerably simplify the process of creating and managing performance data for comparison purposes.

See Also:

Oracle Database Performance Tuning Guide for details

1.10.5.4 Database Statistics Model Consolidation

In this release, the `V$SYSSTAT` and `V$SYS_TIME_MODEL` infrastructures have been consolidated into one advanced component.

This feature also implements a model to break out wait and general statistics into divisions that are usable for automated and manual tuning of the system.

See Also:

Oracle Database Performance Tuning Guide for details

1.10.5.5 Enhanced Active Session History

The Active Session History infrastructure has been enhanced to allow for improved database performance diagnosis and monitoring.

This feature includes enhancements to row source information for query progress monitoring and time model statistics.

See Also:

Oracle Database Performance Tuning Guide for details

1.10.5.6 Enhanced Advisor Framework

In this release, the Advisor Framework has been enhanced to support Health Monitor features.

The Advisor Framework also supports AWR import and export features.

1.10.5.7 Lightweight Jobs

Lightweight jobs have lower creation overhead and much less redo. You can now create large numbers of short, frequently executing jobs and still maintain performance.

See Also:

Oracle Database Administrator's Guide for details

1.10.5.8 Scheduling External Jobs on Multiple Nodes

With this new feature, you can create external jobs (for example, shell scripts) and provide a list of target machines where the jobs should be executed. For example, you can schedule a shell script `x.sh` to execute on machines A, B, and C. The target machines need not necessarily have an Oracle database installed. The jobs are managed from a central location.

This feature enables you to schedule an external job on multiple nodes while managing it from a central location.

See Also:

Oracle Database Administrator's Guide for details

1.10.5.9 Simplified Initialization Parameter Management

Several enhancements in this release make the management of server parameter file and initialization parameter values easier. Some of these enhancements are:

- More fault tolerant `SPFILE`
- Easier recovery from loss of `SPFILE`
- Preventing users from setting invalid parameter values in the `SPFILE`
- Redesigned, more intuitive Enterprise Manager initialization parameter management interface

This feature improves database manageability by making it easier to manage the server parameter file and changes to initialization parameter values.

See Also:

Oracle Database Administrator's Guide for details

1.10.6 Resource Manageability

The following sections describe enhancements to Resource Manageability.

1.10.6.1 I/O Calibration

The DBA can assess the I/O capability of the database's storage system by using the PL/SQL function `DBMS_RESOURCE_MANAGER.CALIBRATE_IO()`. This routine issues a very I/O intensive read-only workload to the database's files to assess the maximum IOPS (I/O requests per second) and MBPS (megabytes of I/O per second) that can be sustained. This data can be reexamined at any time using the DBA table, DBA_RSRC_IO_CALIBRATE, that stores I/O calibration results.

The calibration should be performed when the database is idle and during off-peak hours to minimize the heavy I/O workload from interfering with the regular workload and vice versa.

I/O calibration can be used to understand the performance of the storage subsystem and figure out whether I/O performance problems stem from the database or the storage subsystem. Unlike various external I/O calibration tools, this tool uses the Oracle code stack and issues I/O randomly rather than sequentially. The results, therefore, much more closely match the actual database performance.

See Also:

Oracle Database PL/SQL Packages and Types Reference for details

1.10.6.2 I/O Statistics

I/O Statistics provide consistent and complete statistics from all database clients by:

- I/O type (single and multi block, read and write)
- Component
- Data file
- Consumer group

The I/O statistics collected are used in AWR for monitoring, diagnostics, and tuning. The information is also used by the I/O Resource Manager.

See Also:

Oracle Database Performance Tuning Guide for details

1.10.6.3 Per Session I/O Limits

In previous versions of Oracle, the DBA could specify the maximum amount of time a session could run before some action was taken; for example, the call was aborted, the session was killed, or the session was migrated to a new consumer group. This feature is configured on a per consumer group basis in a resource plan.

In Oracle Database 11*g*, DBAs can also specify the maximum number of I/O requests or the maximum megabytes of I/O that a session can issue before the same set of actions are taken.

This feature is used for two purposes. The first is to automatically identify runaway queries. The second is to move sessions executing long-running calls to lower-priority consumer groups.

See Also:

Oracle Database Administrator's Guide for details

1.10.6.4 Resource Manager Statistics in AWR

AWR contains new historical views for Resource Manager statistics. These new views, DBA_HIST_RSRC_PLAN and DBA_HIST_RSRC_CONSUMER_GROUP, contain the historical version of the statistics in the views V$RESOURCE_PLAN and V$RESOURCE_CONSUMER_GROUP. AWR also contains per-minute metrics for resource consumption and Resource Manager-induced waits in the view V$RSRCMGRMETRIC.

This feature makes it easier to view historical Resource Manager statistics.

See Also:

Oracle Database Administrator's Guide for details

1.10.6.5 Resource Plans

This feature provides sophisticated new built-in resource plans:

- Mixed-workload plan

  Provides resource management for a mixed environment consisting of OLTP and DSS/batch jobs.

- Data Warehouse plan

  Provides resource management for a data warehousing environment.

- Maintenance Plan

  Provides resource management for the maintenance window.

The resource plans are shipped with the database and provide resource management directives that should bring immediate benefits in most customers situations.

See Also:

Oracle Database Administrator's Guide for details

1.10.7 Simplify Configuration

The following sections describe how to streamline management of Oracle Database.

1.10.7.1 Enhanced Optimal Flexible Architecture (OFA) Support

The database installation process has been redesigned to be based on the ORACLE_BASE environment variable. Until now, setting this variable has been optional and the only required variable has been ORACLE_HOME. With this feature, ORACLE_BASE is the only required input, and the ORACLE_HOME setting will be derived from ORACLE_BASE. This change streamlines how Oracle software installation is organized, thereby making ongoing management easier.

This feature improves manageability by making default Oracle Database installations more compliant with Optimal Flexible Architecture (OFA) specifications.

1.10.8 Space, Object and Transaction Manageability

Self-managing database space, object and transaction.

1.10.8.1 AUM Default Setting and Migration Support

Automatic Undo Management (AUM) is now enabled by default. A PL/SQL procedure is provided to help properly size the Undo tablespace for each individual environment.

This feature facilitates seamless migration to AUM from databases being upgraded to the new release so that they can start taking advantage of all the benefits of AUM.

See Also:

Oracle Database Administrator's Guide for details

1.10.8.2 Enhanced Statistics Collection for Partitioned Objects

An improved statistics collection process for partitioned objects avoids having to regather statistics on partitions that have not been touched by using a summary instead.

Partitioned objects tend to become larger and larger, and statistics collection, and particularly global statistics collection, can become increasingly time and resource intensive. This feature significantly improves the speed and accuracy of statistics collection for partitioned objects.

See Also:

Oracle Database VLDB and Partitioning Guide for details

1.10.8.3 Simplified Temp Space Management

This feature demystifies the management of temp space. It allows a DBA to easily determine how much temp space is being used, who is using it, and whether more is needed.

This feature helps reduce errors due to inadequate temp space configuration and identifies application design issues such as runaway queries.

See Also:

Oracle Database Administrator's Guide for details

1.10.9 SQL Manageability

The following sections describe self-managing query performance and access methods.

1.10.9.1 Automatic SQL Tuning with Self-Learning Capabilities

This feature takes automatic SQL tuning to the next level by adding self-learning capabilities to it.

Oracle Database now automatically detects high-load SQL statements and then tunes them automatically as needed in the maintenance window by creating appropriate SQL Profiles. It also issues proactive advice to create new access structures such as indexes that will significantly improve the performance of the high-load SQL statements.

See Also:

Oracle Database Performance Tuning Guide for details

1.10.9.2 Enhanced Optimizer Statistics Maintenance

The publication of statistics is now decoupled from the collection of statistics.

Changing optimizer statistics can have an adverse affect on execution plans. By separating the statistics collection from making them fully available to the optimizer,

application developers and DBAs have the opportunity to assess the effects of the new statistics on execution plans.

See Also:

Oracle Database Performance Tuning Guide for details

1.10.9.3 Multi-Column Statistics

Multi-column statistics can be now collected for any arbitrary set of columns in a single table. This enhancement improves the costing accuracy of many SQL constructs, such as multi-column filter conditions and joins involving more than one column.

Extending the type of optimizer statistics gathered helps the optimizer choose the best plan with improved accuracy.

See Also:

Oracle Database Performance Tuning Guide for details

1.10.9.4 Partition Advisor

The SQL Access Advisor has been enhanced to include advice on how to partition tables, materialized views, and indexes in order to improve performance of SQL statements.

Oracle Database offers a wide variety of partitioning options whose proper use requires expertise and time. This feature makes it easy for all users to use partitioning in a way that is most suitable for their environments by giving intelligent and accurate advice on exactly how to partition a particular object for optimal performance.

See Also:

Oracle Database Performance Tuning Guide for details

1.10.9.5 SQL Plan Management

The SQL plan management feature enables the optimizer to maintain a history of execution plans for a SQL statement. Using the execution plan history, the optimizer is able to detect a new plan representing a plan change for a SQL statement. When the optimizer detects a new plan, it stores the new plan and marks it for performance evaluation and uses the old (currently known good) plan. The optimizer uses the new plan only after its performance is verified to be better than that of the old plan.

Execution plan change for a SQL statement can cause severe performance degradation on a system. Execution plan changes occur due to system changes such as a new optimizer version, refresh of optimizer statistics, and optimizer parameter changes. This new SQL plan management feature prevents performance regressions caused by execution plan changes.

See Also:

Oracle Database Performance Tuning Guide and *Oracle Database PL/SQL Packages and Types Reference* for details

1.10.10 Streams Manageability

The following sections describe enhancements to self-managing Streams (replication and AQ) operations.

1.10.10.1 Automated Alerts for Streams

Alerts are generated automatically if a Streams process becomes disabled. Alerts can also be generated based on user-defined thresholds for key Streams metrics.

Alerts provide the Streams administrator with immediate information on critical Streams components.

See Also:

Oracle Database 2 Day + Data Replication and Integration Guide for details

1.10.10.2 Divergence Detection and Resolution

The `DBMS_COMPARISON` package enables comparison of table data between databases and provides the ability to resolve any data discrepancies.

This feature can be used to validate the consistency of data within a single database or across databases.

See Also:

Oracle Streams Replication Administrator's Guide for details

1.10.10.3 Split and Merge of Streams

Administrators can split a stream in cases where a replica is unavailable and can allow that stream to replicate at a future point in time. After the replica using the split stream catches up, the stream can be merged into a single stream again.

Use of this feature minimizes the impact of unavailable replicas in a Streams replication.

See Also:

Oracle Streams Replication Administrator's Guide for details

1.10.10.4 Streams Message Tracking

This feature provides a simple way for an administrator to track the path an LCR traverses between processes and across databases within Streams. Streams message tracking is initiated by way of a procedure for a particular session. Subsequent LCR activity generated in that session is marked. As the LCRs move through the stream, the actions on those LCRs are traced, and the information is visible in runtime views.

Streams message tracking allows administrators to easily diagnose problems in a Streams configuration.

See Also:

Oracle Streams Replication Administrator's Guide for details

1.10.10.5 Streams Performance Advisor

This feature provides an advisor to diagnose configuration and performance issues while using Streams. The Streams performance advisor reports on performance of all Streams components and provide suggestions for performance tuning.

The Streams performance advisor provides valuable information on setting up and managing a Streams configuration.

See Also:

Oracle Streams Concepts and Administration for details

1.10.10.6 Streams Topology

This feature provides a PL/SQL package and set of data dictionary views for discovery of the Streams topology. Using a single schema with database links and privileges to participating databases, the topology discoverer traverses the various components of Streams (capture, propagation sender/receiver, and apply) in each database to identify correlated components and assign a Streams identifier. Database views provide the administrator with details of the Streams topology.

Administrators can check the Streams set up quickly using the Streams topology views. In addition, the Streams advisors are used in combination with this feature to manage and monitor Streams.

See Also:

Oracle Streams Concepts and Administration for details

1.10.10.7 Synchronous Streams Capture

This feature makes it possible to capture changes to database tables as part of a user transaction.

Synchronous Streams Capture minimizes the resource requirements to replicate a small subset of database tables.

See Also:

Oracle Streams Concepts and Administration for details

1.11 Performance

The following sections describe performance features for Oracle Database 11*g* Release 1 (11.1).

1.11.1 General Server Performance

The following sections describe general server performance enhancements.

1.11.1.1 Direct NFS Client

This features implements NFS client as part of Oracle Database kernel.

The primary benefits of this feature are improved performance and manageability. The Direct NFS Client improves I/O performance by incorporating Oracle-specific optimizations and eliminating the additional overhead traditionally found in operating system kernel mode implementation of NFS protocol. In addition, it simplifies configuration by eliminating the need to manually tune most of the NFS client parameters.

1.11.1.2 Enhanced Partition Pruning Capabilities

Partition pruning now uses bloom filtering instead of subquery pruning. While subquery pruning was activated on a cost-based decision and consumed internal (recursive) resources, pruning based on bloom filtering is activated all the time without consuming additional resources.

The performance of partition pruning has been enhanced. Furthermore, partition pruning will be automatically activated for every join with a partitioned object.

See Also:

Oracle Database VLDB and Partitioning Guide for details

1.11.1.3 Intelligent Cursor Sharing

This feature helps determine whether a cursor will be shared. For example, instead of blindly reusing an existing cursor with its plan, the optimizer takes ranges of safe values into account when evaluating cursor sharability.

Cursor sharing has been enhanced to a more sophisticated sharing model. This enables you to leverage cursor sharing more commonly while preserving bind variable specific plan optimizations for shared statements.

See Also:

Oracle Database Performance Tuning Guide for details

1.11.1.4 NLS Performance Improvements

Oracle Database 11*g* includes incremental performance enhancements for NLS-related features.

This feature provides better performance for customers using NLS features.

See Also:

Oracle Database Globalization Support Guide for details

1.11.2 Improve Basic SQL and PL/SQL

The following sections describe improved performance of basic SQL and PL/SQL statements.

1.11.2.1 Binding and Defining Multiple Buffers in OCI

This feature allows OCI applications to specify noncontiguous buffers for reading or writing in the define or bind calls.

Applications with disperse data sources with frequent read/writes often have data in noncontiguous memory. With this feature, these applications perform faster and use less memory because the overhead of memory copying to a contiguous chunk is eliminated.

See Also:

Oracle Call Interface Programmer's Guide for details

1.11.2.2 Bitmap Join Index Support for Index-Organized Tables

A bitmap join index on index-organized tables (IOTs) provides access to rows of one IOT or heap table based on columns of another IOT or heap table. Bitmap join index support for IOTs can avoid actual joins of tables and reduce the volume of data to be joined by performing restrictions in advance. Bitmap join indexes are space efficient and can speed up queries through bitwise operations.

This feature provides bitmap join index support for index-organized tables (IOTs) to improve performance.

See Also:

Oracle Database Data Warehousing Guide for details

1.11.2.3 Cost-Based Optimizer Enhancements

The cost bases of various SQL execution steps and internal transformations are now determined in a more accurate manner.

Enhancements to the optimizer's internal costing mechanisms improve the accuracy of execution plan generation, thus generating more optimal plans.

1.11.2.4 Network/OCI Fusion

This feature increases the throughput and decreases the overhead of data transfer through OCI/Network layers.

This enhancement improves the performance of SQL statements as well as statements involving large amounts of data transfer.

1.11.2.5 OCI Descriptor Array Allocation

OCI descriptors are opaque data structures meant only for interpretation by OCI but used by applications to signify data such as `OCIDateTime` and `OCILobLocator`.

Until now, descriptors could only be allocated one at a time. This feature allows allocating and freeing an array of descriptors simultaneously.

Applications using many OCI resources can allocate these resources more easily with the new Descriptor Array Allocation interface, providing less code maintenance and less CPU overhead for allocating and freeing memory.

See Also:

Oracle Call Interface Programmer's Guide for details

1.11.2.6 OCI Implicit ROWID Fetching

This feature provides the ability to implicitly fetch the `ROWID` of the rows fetched as the result of a `SELECT...FOR UPDATE` query into memory defined at position 0 while doing array fetches. Users need not alter the query to fetch the `ROWID` along with other columns explicitly.

Applications taking advantage of array fetches can now update individual rows as needed, giving more flexibility to high performance applications.

See Also:

Oracle Call Interface Programmer's Guide for details

1.11.2.7 PL/SQL Function Result Cache

New in Oracle Database 11*g* is the ability to mark a PL/SQL function to indicate that its result should be cached to allow lookup, rather than recalculation, on the next access when the same parameter values are called. This function result cache saves significant space and time. Oracle does this transparently using the input value as the lookup key. The cache is system-wide so that all distinct sessions invoking the function benefit. If the result for a given set of values changes, you can use constructs to invalidate the cache entry so that it is properly recalculated on the next access. This feature is especially useful when the function returns a value that is calculated from data selected from schema-level tables. For such uses, the invalidation constructs are simple and declarative.

Concurrent, multi-user applications that use this feature experience better response times. Applications that implement a session-private scheme consume significantly less memory by using this feature and, therefore, experience improved scalability.

See Also:

Oracle Database PL/SQL Language Reference for details

1.11.2.8 PL/SQL Native Compilation Without Needing a Third-Party C Compiler

On many platforms, the PL/SQL compiler may now generate processor-specific native code directly from the PL/SQL source code without needing to use a third-party C compiler. As in Oracle Database 10*g*, the generated code will be stored canonically in the database catalog. When a unit is needed, the Oracle executable loads the code directly into memory from the catalog, without first staging the unit through a `.DLL` or `.so` file.

The execution speed of natively compiled PL/SQL programs will never be slower than in Oracle Database 10*g* and may be improved in some cases by as much as an order of magnitude.

On platforms with this feature, Oracle Database 10*g* parameters, `plsql_native_library_dir` and `plsql_native_library_subdir_count`, will no longer be needed, and need not be set. If upgrading from Oracle Database 10*g*, any existing files in those directories may safely be deleted.

The benefit of PL/SQL native compilation is automatically available with Oracle Database 11*g*. No third-party software (neither a C compiler nor a DLL loader) is needed.

Note that this feature may not be available on all platforms in the first release of Oracle 11*g*. If this feature is not available on a particular platform, Oracle Database 10*g* implementation of native compilation will be unchanged on that platform. Check your platform-specific documentation to verify that this feature is available on your platform.

See Also:

Oracle Database PL/SQL Language Reference for details

1.11.2.9 Query Result Cache

A separate shared memory pool is now used for storing and retrieving cached results. Query retrieval from the query result cache is faster than rerunning the query. Frequently executed queries will see performance improvements when using the query result cache.

The new query result cache enables explicit caching of results in database memory. Subsequent queries using the cached results will experience significant performance improvements.

See Also:

Oracle Database Performance Tuning Guide for details

1.11.3 Optimize Key Platforms

The following sections describe platform-specific optimizations on platforms such as Linux, Windows, and Intel.

1.11.3.1 ODBC Performance Improvement

The Oracle ODBC Driver has been enhanced with the following options that can be set using Setup DDL on Windows or through `odbc.ini` on Linux and Solaris:

- Statement caching
- Removal of optimizer hints for catalog functions
- Support for column binding for `NUMBER` columns as `FLOAT`

ODBC applications will perform faster, with little or no code change.

1.12 Security

Features described in the following sections include security and compliance.

1.12.1 Oracle Advanced Security

The following sections describe Oracle advanced security features.

1.12.1.1 Kerberos Cross Realm Support

This feature updates the Oracle Kerberos implementation to support cross realm authentication. Now a Kerberos principal in one realm can provide authentication to a principal in another realm.

The benefit of this feature is that it makes Kerberos a more viable alternative for customers seeking strong authentication without the overhead of managing certificates and public key infrastructure (PKI).

1.12.1.2 SYSDBA Strong Authentication

All connections to the database can now use strong authentication, including those made as `SYSDBA` and `SYSOPER`.

Oracle has supported strong authentication through PKI, Kerberos, and Radius since Oracle8*i* for all connections to the database except those connections made as `SYSDBA` or `SYSOPER`. This feature completes Oracle support for strong authentication by extending the strong authentication feature to connections as `SYSDBA` and `SYSOPER`.

See Also:

Oracle Database Security Guide for details

1.12.1.3 Tablespace Encryption

Tablespace encryption is an enhancement to the Oracle Advanced Security Transparent Data Encryption solution. Using tablespace encryption, customers can encrypt an entire tablespace, encrypting all data within the tablespace. When the database accesses the tablespace, the relevant data blocks are transparently decrypted for the application.

Transparent Data Encryption tablespace encryption provides an alternative to Transparent Data Encryption column encryption by enabling encryption of an entire tablespace. This eliminates the need for granular analysis of applications to determine which columns to encrypt, especially for applications with a large number of columns containing personally identifiable information (PII). Using tablespace encryption, customers can encrypt entire tables, eliminating the need to identify which columns contain personally identifiable information (PII). Customers who have small amounts

of data to encrypt can continue to use the Transparent Data Encryption column encryption solution.

See Also:

Oracle Database Advanced Security Administrator's Guide for details

1.12.1.4 Hardware-Based Master Key Protection

This feature allows the Transparent Data Encryption (TDE) master key to be stored in an external Hardware Security Module (HSM) for even stronger security.

This feature provides even stronger security for the Transparent Data Encryption (TDE) master key for customers who are concerned about storing the master key on the operating system.

See Also:

Oracle Database Advanced Security Administrator's Guide for details

1.12.2 Secure by Default

The following sections describe out-of-the-box secure configuration with increased password protections, secure file permissions, optional default audit settings, and new controls on network callouts from the database.

1.12.2.1 Audit By Default

This feature automatically configures the database for auditing and turns on auditing for specific events such as database connections.

Auditing is an important part of the security layer. Oracle has turned on some auditing settings by default to help customers better track connections to the database.

See Also:

Oracle Database Security Guide for details

1.12.2.2 Built-in Password Complexity Checker

This feature builds into the database the password complexity routine that is documented in the *Oracle Database Security Guide*.

This built-in functionality can be easily turned on to guarantee that complex passwords are used when setting or resetting a password.

See Also:

Oracle Database Security Guide for details

1.12.2.3 Built-in User Profile

This feature makes it easier to associate a database user with a profile by providing a build-in profile.

Customers can use a built-in profile to enforce password expiration and reuse policies.

See Also:

Oracle Database Security Guide for details

1.12.2.4 Fine-Grained Access Control on Network Call-outs from the Database

The packages `UTL_TCP`, `UTL_INADDR`, `UTL_HTTP`, `UTL_SMTP`, and `UTL_MAIL` allow Oracle users to make network callouts from the database using raw TCP or using higher level protocols built on raw TCP. Until now, when the granularity of a privilege is simply `execute` on each package, there has been no way to allow a user access to just a list of specified internet hosts. The new package `DBMS_NETWORK_ACL_ADMIN` allows fine-grained control using ACLs implemented by XML DB.

This feature enables the DBA to carefully control which internet hosts Oracle users can access using the supplied PL/SQL packages.

See Also:

Oracle Database Security Guide for details

1.12.2.5 Improve Security for Network Administration, Registration, and Operation

This feature provides secure service registration with the listener. Listener administration is secure by default. The new model provides an easy, optional, and manageable process administration and registration.

This security improvement prevents unauthorized use of remote and local operations.

See Also:

Oracle Database Net Services Administrator's Guide for details

1.12.2.6 Parameters for Enhanced Security of Database Communication

This feature provides the following enhancements:

- Reporting bad packets received on the database from protocol errors.
- Terminating or resuming server execution on receiving bad packets.
- Configuring the maximum number of authentication attempts.
- Controlling the display of the database version banner.
- Configuring banners for unauthorized access and auditing user actions.

This feature helps to keep data safe and secure.

See Also:

Oracle Call Interface Programmer's Guide, *Oracle Database Security Guide*, and *Oracle Database Net Services Reference* for details

1.12.2.7 Support Nonanonymous LDAP Access for Net Naming

This feature provides support for nonanonymous LDAP lookup for Network naming. For more secure LDAP access, users can specify that clients must identify themselves before a name lookup.

This feature results in improved security between LDAP and Net naming.

See Also:

Oracle Database Net Services Administrator's Guide for details

1.12.3 Security Manageability

The following sections describe replacement and integration of Enterprise Security Manager, Oracle Policy Manager, Selective Audit, and Audit Vault UI into a Java/Enterprise Manager SDK.

1.12.3.1 Integrated Database Security Manageability

Functionality previously provided in Enterprise Security Manager and Oracle Policy Manager has been integrated into Enterprise Manager. In addition, manageability for new features such as Transparent Data Encryption has been added to Enterprise Manager.

Over the years, Oracle has had numerous tools for managing database security. Now all of the functionality in those tools has been integrated into Enterprise Manager, dramatically simplifying security management in the database.

1.12.3.2 Virtual Private Catalog

This enhancement allows the catalog administrator to grant visibility of certain registered databases in the catalog to certain RMAN users.

This feature improves catalog security by allowing an RMAN user to view only his or her authorized, registered databases within a catalog. An RMAN user is no longer able to see all registered databases.

See Also:

Oracle Database Backup and Recovery User's Guide for details

1.12.4 Stronger Password Protection

The following sections describe authentication features for Oracle Database 11*g* Release 1 (11.1).

1.12.4.1 Standards-Based Password Algorithm

This feature modifies the verifier or hash used to store Oracle passwords.

This feature provides stronger protection for stored database passwords based on industry standard algorithms and provides enhanced security for password-based authentication by enabling usage of mixed case in passwords.

See Also:

Oracle Database Security Guide for details

1.13 Windows

The following sections describe Windows features for Oracle Database 11*g* Release 1 (11.1).

1.13.1 Integration with Windows Services

These features provide new and enhanced support for Oracle databases running Windows. They integrate natively with Windows services, including improved integration with Active Directory and Volume Shadow Copy Service support.

1.13.1.1 Active Directory Security Enhancements

This feature enhances security through better integration with the default Active Directory configuration, adding support for OS authentication with Active Directory.

This feature improves Net Naming support for Microsoft Active Directory.

1.13.1.2 Volume Shadow Copy Service (VSS) Writer

The Oracle VSS writer allows Oracle Database to participate in VSS-initiated backup and recovery on Windows 2003 and above. VSS is a Windows infrastructure composed of three participants:

- Requestors - management applications that initiate backup and recovery
- Writers - applications that own the data
- Providers - storage/hardware applications that coordinate the reading and writing of the data to appropriate storage

Several leading backup and storage vendors currently provide requestor and provider applications. With the Oracle VSS writer, customers can reliably back up and recover Oracle data in a VSS environment, which interoperates with these vendors' components.

This feature provides full interoperability with Volume Shadow Copy Service (VSS) infrastructure on Windows 2003 and above.

See Also:

Oracle Database Platform Guide for Microsoft Windows for details

Exhibit 16

THIS EXHIBIT HAS BEEN REDACTED IN ITS ENTIRETY

Exhibit 17

THIS EXHIBIT HAS BEEN REDACTED IN ITS ENTIRETY

Exhibit 18

THIS EXHIBIT HAS BEEN REDACTED IN ITS ENTIRETY

Exhibit 19

Oracle® Application Server 10*g*

Concepts

10*g* (9.0.4)

Part No. B10375-02

March 2004

ORACLE

Oracle Application Server 10*g* Concepts, 10*g* (9.0.4)

Part No. B10375-02

Copyright © 2002, 2004, Oracle. All rights reserved.

Primary Author: Theresa Robertson

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

2

J2EE, Web Services, and Internet Applications

This chapter describes the Java 2 Platform, Enterprise Edition (J2EE), Web Services, and Internet application development and deployment environment in Oracle Application Server, explaining its features and concepts. The topics include:

- Application Development and Deployment in Oracle Application Server
- Oracle HTTP Server
- Oracle Application Server Containers for J2EE
- Oracle Application Server TopLink
- Oracle Business Components for Java
- Oracle Application Server Web Services
- Oracle Application Server Forms Services
- Oracle XML Developer Kit
- Oracle Application Server PL/SQL Platform
- Oracle Content Management Software Developer Kit
- Oracle Application Server MapViewer

Application Development and Deployment in Oracle Application Server

Oracle Application Server provides an integrated, standards-based environment that provides for both developing applications and then deploying them, reliably delivering the applications to users across an enterprise. The following sections describe how this environment aids application development and deployment.

What Types of Applications Can Be Developed for Deployment in Oracle Application Server?

Oracle Application Server allows Web application developers to develop their sites in a variety of languages and technologies:

- Java and J2EE
- Web services
- XML
- PL/SQL

Table 2–1 lists the different technologies and programming languages that you can use to build applications for deployment with Oracle Application Server.

Table 2–1 Supported Technologies and Programming Languages

Java and J2EE	XML	PL/SQL	Web Services
▪ JavaServer Pages (JSP) v. 1.2 ▪ Java Servlets v. 2.3 ▪ Enterprise JavaBeans (EJB) v. 2.0 ▪ Java Database Connectivity (JDBC) v. 2.0 Extensions ▪ Java Transaction API (JTA) v. 1.0 ▪ Java Naming and Directory Interface (JNDI) v. 1.2 ▪ Java Message Service (JMS) v. 1.0.2b ▪ Java Authentication and Authorization Service (JAAS) v. 1.0 ▪ J2EE Connector Architecture v. 1.0 ▪ Java API for XML Parsing (JAXP) v. 1.1 ▪ Java Mail v. 1.0	▪ XML v. 1.0 ▪ XML Namespaces v. 1.0 ▪ Document Object Model (DOM) v. 1.0/2.0 ▪ Extensible Stylesheet Language Transformations (XSLT) v. 1.0 ▪ XML Schemas v. 1.0 ▪ Simple API for XML (SAX) v. 1.0/2.0 + Extensions ▪ XML Path Language (XPath) v. 1.0 ▪ XSQL ▪ Internet Data Access Presentation (IDAP)	▪ PL/SQL Server Pages v. 9.0.4 ▪ PL/SQL Web Toolkit v. 9.0.4.0.1 ▪ Oracle Application Server Forms Services v. 9.0.4	▪ Web Services Description Language (WSDL) v. 1.1 ▪ Universal Description, Discovery, and Integration (UDDI) v. 2.0 ▪ Simple Object Access Protocol (SOAP) v. 1.1

The following sections explain how Oracle Application Server supports these technologies and programming languages.

Oracle HTTP Server

Oracle HTTP Server is the underlying deployment platform for all programming languages and technologies that Oracle Application Server supports. It provides a Web **listener** for Oracle Application Server Containers for J2EE (**OC4J**) and the framework for hosting static and dynamic pages and applications over the Web. Based on the proven technology of the **Apache** HTTP Server, Oracle HTTP Server includes significant enhancements that facilitate load balancing, administration, and configuration. It also includes a number of enhanced **modules**, or **mods**, which are extensions to the **HTTP server** that extend its functionality for other enterprise applications and services.

Oracle HTTP Server allows developers to program their sites in a variety of languages and technologies, such as **Java**, Perl, C, C++, PHP, and PL/SQL. Additionally, it can serve as either a forward or reverse **proxy server**. The following sections describe how Oracle HTTP Server provides a robust deployment platform for dynamic Web sites and applications.

ORCL00370718

Oracle HTTP Server Components

Oracle HTTP Server consists of several components that run within the same process. These components provide the extensive list of features that Oracle HTTP Server offers when handling client requests. Major components include the following:

- **HTTP listener:** Oracle HTTP Server is based on an Apache HTTP listener to serve client requests.
- **Modules (mods):** Many of the standard Apache mods are included with Oracle HTTP Server. Oracle also includes several internal modules that are specific to Oracle HTTP Server components. Table 2–2 lists some of the Oracle HTTP Server modules.

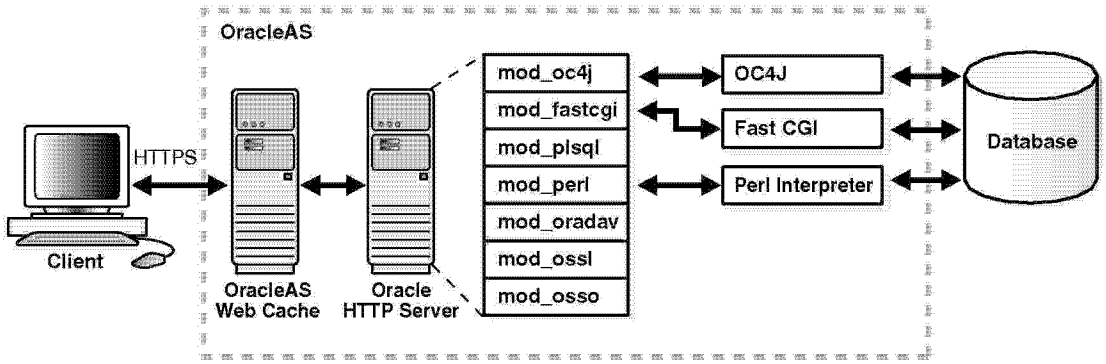
Table 2–2 Prominent Oracle HTTP Server Modules

Module	Description
mod_fastcgi	Supports FastCGI, which allows C, C++, and Java CGI programs to run in a performant environment
mod_perl	Routes requests to the Perl Interpreter
mod_plsql	Routes requests for stored procedures to the database server
mod_oc4j	Supports communication with Oracle Application Server Containers for J2EE and also performs some load balancing tasks
mod_oradav	Supports file as well as database distributed authoring and versioning
mod_ossl	Supports Secure Sockets Layer (SSL) and certificate sharing
mod_osso	Routes requests to Oracle Application Server Single Sign-On server

See Also: *Oracle HTTP Server Administrator's Guide* for a complete list of modules

Figure 2–1 shows the path of various requests through Oracle HTTP Server components.

Figure 2–1 Oracle HTTP Server Request Flow



Oracle HTTP Server Architecture

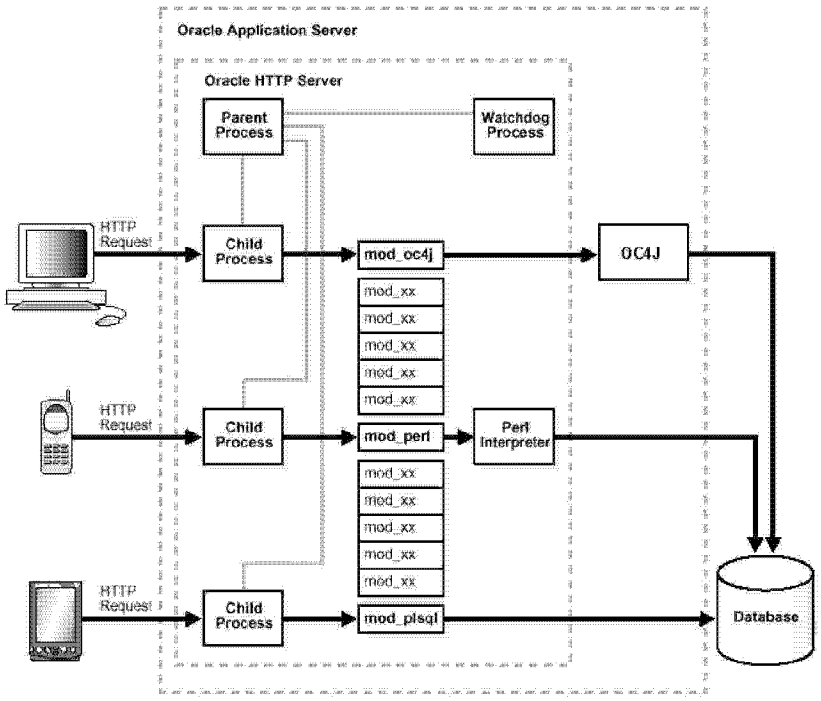
At startup, the Web server parent process loads the entire configuration and the associated mods, and spawns a preconfigured number of child processes.

ORCL00370719

Note: On Windows systems, the Web server main process is a child process that spawns multiple threads.

Figure 2–2 shows the process architecture of Oracle HTTP Server in a UNIX environment.

Figure 2–2 Oracle HTTP Server Process Architecture



The parent process does not listen to HTTP requests. Its sole job is to ensure that the child processes are running or that new ones are started when the load requires it.

On UNIX platforms, each child process handles a single HTTP request. The child processes determine who should take the next request based on a mutex mechanism that you can configure.

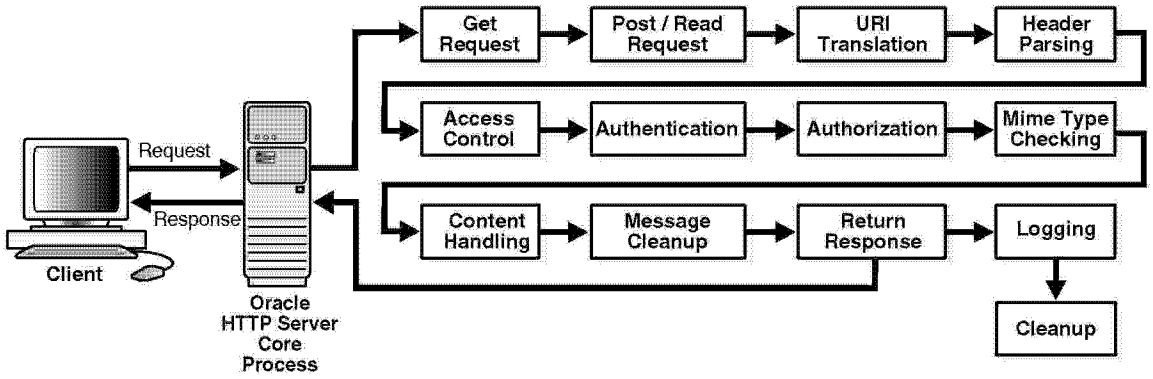
Note: On Windows platforms, threads handle HTTP requests instead of child processes.

Modular Architecture

The architecture of Oracle HTTP Server is modular. The core HTTP listener is very small, and all capabilities are implemented as modules that plug in and are invoked at the appropriate place during the HTTP request lifecycle. Figure 2–3 shows the lifecycle of an HTTP request in Oracle HTTP Server.

ORCL00370720

Figure 2–3 Oracle HTTP Server HTTP Request–Response Cycle



A child process guides the request through this entire lifecycle. The modules register their application programming interfaces (APIs), which are then either invoked automatically when the request reaches a certain stage in its lifecycle, or can be configured to be invoked only in certain situations.

Oracle HTTP Server Features

In addition to the standard Web server functionality of serving client requests to other Oracle Application Server components, Oracle HTTP Server provides the necessary features for both creating dynamic applications and implementing enterprise support. It also contains security enhancements that help protect important business resources. Key features of Oracle HTTP Server include the following:

- **Dynamic monitoring services (DMS):** These services automatically measure runtime performance statistics for both Oracle HTTP Server and Oracle Application Server Containers for J2EE (OC4J) processes. As applications run, DMS collects detailed performance statistics. This data allows you to monitor the duration of important request processing phases and status information. With this information, you can locate performance bottlenecks and tune the application server to maximize throughput and minimize response time.
- **Request ID:** To enhance request tracking through various components, a request ID is now attached to each request. This provides more detailed tracking information, allowing you to see how much time a particular request spends in any component or layer.
- **External API for performance monitoring:** This API allows you to use external, third-party performance monitoring tools to monitor Oracle Application Server-based J2EE components, such as servlets and JSPs, as well as J2EE containers.
- **Proxy Plug-In for Using Non-Oracle HTTP Listeners:** To accommodate requirements for non-Oracle HTTP listeners, Oracle HTTP Server provides a proxy plug-in that can be plugged directly into Sun ONE or Microsoft Internet Information Server (IIS). This proxy plug-in is used to forward requests for Oracle Application Server component services to Oracle HTTP Server, which is placed behind the non-Oracle listener.
- **OC4J Plug-In for IIS and Sun ONE:** The OC4J Plug-In provides a way for you to use the IIS and Sun ONE third-party listeners to access servlets running in OC4J without having to use OHS as a proxy. The OC4J Plug-In routes requests directly from the third-party HTTP listener to OC4J.

ORCL00370721

- **Oracle Application Server Single Sign-On Plug-in:** This plug-in is the Oracle single sign-on solution for third-party listeners such as Sun ONE and IIS. The plug-in is designed to protect native third-party listener applications using the single sign-on infrastructure. Using this plug-in, you can be authenticated to different third-party listener applications using only one password. You can integrate these protected third-party listener applications with other single sign-on enabled applications as long as they are all protected by the same single sign-on server.

Common Gateway Interface (CGI) Support

Requests that are sent to a common gateway interface (CGI) program may invoke two new processes—the child process that handles the HTTP request and the CGI program itself. It is possible to avoid this overhead by configuring Oracle HTTP Server to pre-start child processes and keep them running, leveraging the Perl module to run the CGI programs in memory, or using the FastCGI mechanism.

The following Oracle HTTP Server features support CGI:

- **FastCGI:** FastCGI supports applications written with Perl, C, C++, and Java. Each CGI application runs in a single child process of the Web server. This improves server performance because it eliminates the need to start a new process for every application request.
- **Perl Interpreter:** The Perl Interpreter supports Perl applications and runs them inside of the Web server process. This allows for greater performance because running applications does not start new processes. Since the interpreter runs inside of the Web server process, it has access to Web server services such as log files.

Oracle Application Server Containers for J2EE

Oracle Application Server Containers for J2EE (OC4J) is a fast, lightweight, and scalable J2EE 1.3 certified server implementation that is written in Java and runs on a standard Java Virtual Machine (JVM). It has been designed to support the standard APIs in Table 2–3.

Table 2–3 Oracle Application Server Containers for J2EE Supported APIs

API	Version
JavaServer Pages (JSP)	1.2
Java Servlet	2.3
Enterprise JavaBeans (EJB)	2.0
Java Database Connectivity (JDBC)	2.0 Extensions
Java Transaction API (JTA)	1.0
Java Message Service (JMS)	1.0.2b
JavaMail	1.2
JavaBeans Activation Framework	1.0
Java API for XML (JAXP)	1.1
J2EE Connector Architecture	1.0
Java Authentication and Authorization Service (JAAS)	1.0

ORCL00370722

Introduction to J2EE Application Development

The following sections provide introductory definitions and summaries of application programming technologies that Oracle Application Server supports. For detailed information on the technologies introduced in this section, refer to the following information sources.

See Also:

- The official J2EE Web site at `http://java.sun.com/j2ee`
- *The J2EE Tutorial* at `http://java.sun.com/j2ee/tutorial`
- The Oracle Application Server Documentation Library

What Is a J2EE Application?

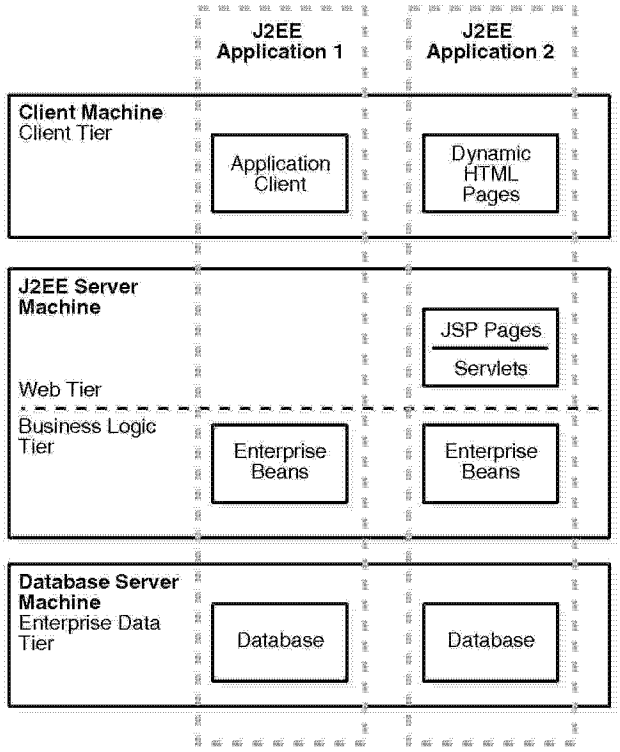
A J2EE application is an application that is written in Java using the J2EE APIs. It can be deployed, managed, and executed on a J2EE-compatible server. The J2EE application itself is composed of a set of components, such as Web presentation modules, business logic modules, and data access modules. Each component is assembled into the overall application with all of its related classes and XML deployment descriptors.

J2EE Distributed Multi-tiered Application Model

The J2EE platform provided in Oracle Application Server uses a multi-tiered distributed application model. A multi-tiered distributed application model divides application logic into components according to function, and the various application components that make up J2EE applications can be installed on different machines, depending on which tier in the multi-tiered J2EE environment the application component belongs. Figure 2–4 shows two multi-tiered J2EE applications divided into the client, Web, business logic, and enterprise data tiers.

ORCL00370723

Figure 2–4 J2EE Distributed Multi-tiered Application Architecture



You can distribute J2EE applications across the four tiers shown in this figure, but generally they are considered to be three-tier applications because they are usually distributed over the following machine locations:

- client machines
- J2EE Server machines hosting presentation services, like JSPs and servlets, and business logic components, like EJBs
- database servers or legacy machines at the back end

Three-tiered applications that run in this way extend the standard two-tiered client and server model by placing an application server between the client and the back-end storage.

Types of J2EE Clients

J2EE applications support the following clients:

- **Application Clients:** Applications running on a client machine that directly access enterprise beans that are running in the business logic tier. Application clients can also open an HTTP connection to establish communication with a servlet running on the Web tier if a J2EE application requires it.
- **Dynamic HTML and XML Pages:** In the context of J2EE applications, dynamic HTML and XML pages are either generated by servlets or created with JavaServer Pages technology running in the Web tier. These pages can be extensions to traditional static HTML pages, allowing application developers to offer customized and personalized pages to the client.

ORCL00370724

Types of J2EE Application Components

You can use the following components in J2EE applications:

- **Servlets:** A servlet is a Java class that executes behind a Web server and can extend the capability of the Web server to provide services for dynamic page creation or application logic. The servlet works in the standard HTTP request-response model.
- **JavaServer Pages:** JavaServer Pages (JSPs) are text files that contain two types of information: static template data, which can be expressed in any text-based format, such as HTML, XML, or WML (Wireless Markup Language); and JSP elements, which construct dynamic content.
- **Enterprise Beans:** Enterprise beans are server-side components that encapsulate the business logic of an application.

Types of J2EE Containers

A **container** provides the runtime support for J2EE application components. Containers provide a federated view of the underlying J2EE APIs to the application components. J2EE application components never interact directly with other J2EE application components. They use the protocols and methods of the container for interacting with each other and with platform services. Interposing a container between the application components and the J2EE services allows the container to transparently inject the services defined by the components' deployment descriptors, such as declarative transaction management, security checks, resource pooling, and state management.

Before a Web or enterprise bean component can run, it must be assembled into a J2EE application and deployed into a J2EE container. The assembly process involves specifying container settings for each component, which customize the underlying support provided by the J2EE server. These settings can be standard J2EE settings or container-specific settings, depending on your application requirements. Some of the container settings that you can specify include security services, **transaction** model, naming and directory lookup, and remote connectivity model.

There are two primary types of J2EE container:

- **Enterprise JavaBeans Container:** The Enterprise JavaBeans container (**EJB container**) manages the execution of all enterprise beans for J2EE applications. Enterprise beans and their container run on the J2EE server.
- **Web Container:** Web components such as JSP pages and servlets are managed and executed in the **servlet container**. The container provides services such as request dispatching, security, concurrency, and life cycle management. The Web container also gives **Web components**, typically JSPs and servlets, access to the J2EE APIs such as naming, transactions, and JDBC.

J2EE Application Packaging Concepts

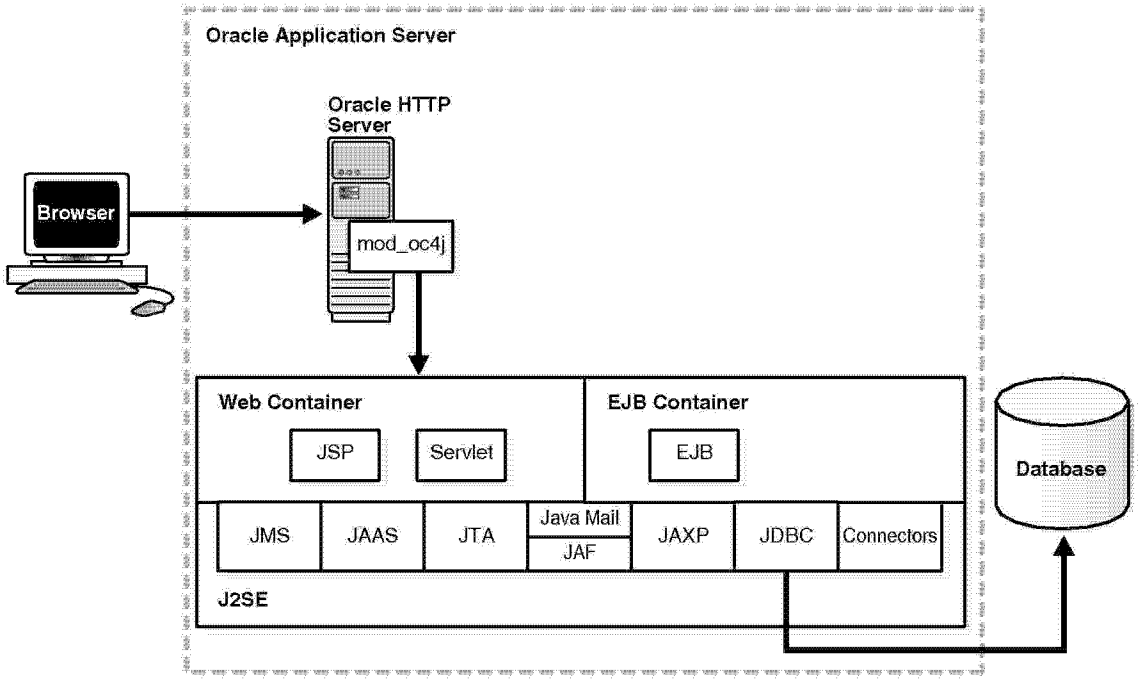
J2EE components are packaged separately and bundled into a J2EE application. Each component, with its related files such as GIF and HTML files or server-side utility classes, are packaged together with a deployment descriptor (DD), and are assembled into a **module** that is added to the J2EE application. Typically, a J2EE application is composed of one or more enterprise beans and Web or application client component modules.

ORCL00370725

Oracle Application Server Containers for J2EE Architecture

Figure 2–5 shows the architecture of OC4J within Oracle Application Server.

Figure 2–5 Oracle Application Server Containers for J2EE Architecture



OC4J is supported by the Java 2 Platform, Standard Edition (**J2SE**) infrastructure as shown in Figure 2–5. This means that the OC4J Web container and OC4J **EJB container** use the J2SE **virtual machine.** J2EE applications are modularized for reuse of application components, such as:

- User interfaces, which can be composed of JavaServer Pages (**JSPs**), dynamic **HTML**, and so on
- Business logic, which is usually contained in an Enterprise JavaBean (**EJB**) or normal Java classes

The J2EE containers also perform services for applications, such as providing access to the APIs and lifecycle management.

Oracle Application Server Containers for J2EE Features

Oracle Application Server Containers for J2EE (OC4J) has the following features.

Oracle Application Server Containers for J2EE Containers

OC4J supplies the following J2EE containers:

- A servlet container that complies with the servlet 2.3 specification
- A JSP container that complies with the Sun JSP 1.2 specification
- An EJB container that complies with the EJB 2.0 specification

ORCL00370726

Oracle Application Server Containers for J2EE Servlet Container A servlet is a Java program that runs on a J2EE server, such as OC4J. A servlet is one of the application component types of a J2EE application. It must execute under the control of a servlet container, which is part of the OC4J Web container. The servlet container calls the servlet's methods and provides services that the servlet needs when running.

See Also: "Types of J2EE Application Components" on page 2-9

About the Servlet Container

The servlet container executes and manages servlets. It provides the servlet with access to properties of the HTTP request, such as headers and parameters. Also, the container provides the servlet with access to other Java APIs, such as **JDBC** to access a database, remote method invocation (**RMI**) to call remote **objects**, or JMS to perform asynchronous messaging.

How the Servlet Container Works

When a request is mapped to a servlet, the servlet container performs the following steps:

1. If an **instance** of the servlet does not exist, the container does the following:
   a. Loads the servlet class
   b. Instantiates an instance of the servlet class
   c. Initializes the servlet instance
2. The container then invokes the servlet, passing request and response objects. The request object contains information about the client, request parameters, and **HTTP headers**. The response object returns the servlet's output to the client.

The servlet extracts information from the client request, accesses external resources, and then populates the response based on that information.

Oracle Application Server Containers for J2EE JavaServer Pages Container JavaServer Pages provide a convenient way to generate dynamic content in Web pages. JSP technology, which is closely coupled with servlet technology, allows you to include Java code fragments and make calls to external Java components (in the form of tags and directives) from within your Web pages. Typically, the markup code used to compose your Web pages is HTML or XML. JSPs work well as a front-end for business logic and dynamic functionality in **JavaBeans** or Enterprise JavaBeans (EJBs).

A JSP is translated into a Java servlet before being run, and it processes HTTP requests and generates responses like any servlet. However, JSP technology provides a more convenient way to code a servlet. Translation occurs the first time the application is run. A **JSP translator** is triggered by the `.jsp` file name extension in a **URL**.

JSPs are fully interoperable with servlets. You can include output from a servlet or forward the output to a servlet, and a servlet can include output from a JSP or forward output to a JSP.

About the JSP Translator

The JSP translator has a translator and a compiler. The JSP translator translates a JSP into a Java source file. The container compiles the source file into a Java bytecode (`.class`) file, which executes as a servlet in the servlet container using the JSP runtime library. The servlet container provides access to Java APIs and other services.

ORCL00370727

How the JSP Translator Works

When a user requests a URL that maps to a JSP file, such as `http://host/Hello.jsp`, the following steps occur:

1. The Web server invokes the JSP translator, which translates `Hello.jsp` and produces the file `Hello.java`.
2. The Java compiler is invoked, creating a `Hello.class` servlet.
3. `Hello.class` runs, using the JSP runtime library, which contains the supporting files to interpret the tags and directives from the JSP.
4. If the `Hello` class requires information from a database, then the servlet container provides JDBC access to the class so it can retrieve the information and return its output to the client browser.

Oracle Application Server Containers for J2EE Enterprise JavaBeans Container The OC4J **EJB container** manages the execution of enterprise beans for J2EE applications. Like the OC4J Web container, the EJB container uses the J2SE **virtual machine**. The following sections describe what services the EJB container provides to J2EE applications and how it works.

Enterprise beans are the J2EE components that implement Enterprise JavaBeans technology. Enterprise beans run in the EJB container. An enterprise bean is a portable server-side component that encapsulates the business logic of an application. There are three types of EJBs: **session beans**, **entity beans**, and **message-driven beans**.

About the Oracle Application Server Containers for J2EE EJB Container

The OC4J EJB container provides system-level services to EJBs similar to the services that the Web container provides to servlets and JSPs. The container has configurable settings that customize the underlying support provided by OC4J, the J2EE server. The configurable settings include security, transaction management, Java Naming and Directory Interface (**JNDI**) lookups, and remote connectivity. In addition to the configurable settings, the container also manages EJB life cycles, database connection resource pooling, data persistence, and access to the J2EE APIs.

How the EJB Container Works

How the EJB container works depends on what type of enterprise bean you are using. The container manages the execution of the enterprise bean for one J2EE application.

For **session beans**, the EJB container provides all of the services that the Web container provides to Web components, such as access to APIs and the **virtual machine**, **transaction services** like Container Managed Transactions (CMTs), and secure and authorized EJB method invocation.

For entity beans, which represent business objects in a persistent storage mechanism, there are models for how the persistence is performed. You can have either **bean-managed persistence (BMP) beans** or **container-managed persistence (CMP) beans**. With bean-managed persistence, the entity bean code contains the calls that access the database and the EJB container triggers callback methods on your code. Entity beans with bean-managed persistence execute in the EJB container with the typical container support and services. However, if you are using container-managed persistence, then the EJB container automatically generates the necessary database access calls. The EJB methods do not require any JDBC code to manage EJB data persistence.

ORCL00370728

J2EE Services

Java 2 Platform Enterprise Edition (J2EE) provides core services for writing J2EE components. The J2EE containers manage access to these services for the application components. The services are as follows:

- **Java Database Connectivity (JDBC):** This service lets you invoke SQL commands from Java programming methods. You use the JDBC API in an enterprise bean when you override the default container-managed persistence or have a session bean access the database. You can also use the JDBC API from a servlet or a JSP to access a database directly without going through an enterprise bean.

  Oracle Application Server includes the following drivers to provide highly scalable and reliable connectivity to both Oracle and non-Oracle data sources:

  - **Oracle JDBC drivers:** The Oracle JDBC drivers, in addition to providing standard JDBC API support, have extensions to support Oracle-specific datatypes and to enhance their performance. They are meant to be used with the Oracle database.
  - **J2EE Connectors:** The J2EE Connector architecture, part of the J2EE platform, provides a Java-based solution for connecting various application servers and enterprise information systems that are already in place.
  - **DataDirect Connect Type 4 JDBC drivers:** DataDirect JDBC drivers are meant specifically for connecting to non-Oracle databases, such as Microsoft SQL Server and Sybase.

- **Java Message Service (JMS):** This service is a messaging standard that allows J2EE application components to create, send, receive, and read messages. It enables distributed communication that is loosely coupled, reliable, and asynchronous.

  For this release, JMS support has been enhanced by the addition of a lightweight JMS provider in addition to the Oracle JMS offered in previous versions. The new lightweight JMS is fully JMS 1.0 compatible and can support durable messaging through a file-based persistence mechanism, which provides improved stability and performance. Support for Message Driven Beans is also now available for both Oracle JMS (AQ) and the lightweight OC4J JMS.

- **Java Transaction API (JTA):** This service provides a standard demarcation interface for demarcating transactions. Typically, it is used in J2EE applications that use two or more separate database access operations that depend on each other to demarcate where the entire transaction begins, rolls back, and commits.

  OC4J provides additional support for two-phase commits for applications that require commit coordination across machines and containers. The two-phase commit engine is responsible for ensuring that when a distributed transaction ends, changes to all participating databases are all either committed or rolled back.

- **Java Naming and Directory Interface (JNDI):** This service provides a standard interface to naming and directory services. J2EE applications use JNDI to find other distributed objects. The JNDI Interface has two parts: an application-level interface used by application programs to access naming and directory services, and a service provider interface to attach a provider of naming and directory services.

- **JavaMail Technology:** This service provides J2EE applications with a JavaMail API and a JavaMail service provider with which to send e-mail notifications.

- **Java API for XML (JAXP):** This service provides support for the industry standard SAX and DOM APIs for parsing XML documents, as well as support for XSLT

ORCL00370729

transform engines. It enables applications running in a J2EE container to make use of XML.

- **J2EE Connector Architecture:** This service provides an interface for resource adapters that allow J2EE applications to access and interact with databases and other enterprise information systems (**EIS**). Oracle Application Server provides several out-of-the box adapters, and also allows you to build your own.
- **Java Authentication and Authorization Service (JAAS):** This service provides a way for a J2EE application to authenticate users against different security provider systems and authorize a specific user or group of users to run J2EE applications under **role** permission and control enforcement.

  Oracle Application Server provides an implementation of Java Authentication and Authorization Service (JAAS) that integrates with the Oracle Application Server J2EE security infrastructure to enforce security constraints for Web components (servlets and JSPs) and EJB components. The Oracle Oracle Application Server Java Authentication and Authorization Service implementation does the following:

  - Integrates Java-based applications with Oracle Application Server Single Sign-On, including **authentication**, thereby giving you extensible security for Java-based applications
  - Manages access control policies centrally in Oracle Internet Directory, controls access by role, and partitions security **policy** by subscriber
  - Supports impersonation of a specific user, allowing an enterprise bean, servlet, or JSP to run with the permissions associated with the current client or a specified user

    See Also: The Sun Microsystems, Inc. published specifications for these services, and *The J2EE Tutorial* for instructions on how to use them in J2EE applications, at:

    `http://java.sun.com`

Oracle J2EE Services

In addition to the standard J2EE Services, Oracle also provides the following services to Java developers:

- **Java Object Cache** stores frequently accessed or resource-intensive objects in memory or on disk. It is a low-level object caching API, supporting generic object types such as memory objects, disk objects, pooled objects, and StreamAccess objects. Java Object Cache uses distributed object management to coordinate updates and invalidations of Java objects.

  Objects are loaded using a user-defined CacheLoader object and accessed through the easy-to-use API. This eliminates the need to repeatedly create and load information within a Java application. The Java Object Cache retrieves content quickly and greatly reduces the load on Oracle Application Server.

  Java Object Cache provides caching for expensive or frequently used Java objects when the application servers use a Java program to supply their content. Cached Java objects may contain generated pages, or they may provide support objects within the program to assist in creating new content. Java Object Cache automatically loads and updates objects as specified by the Java applications and includes APIs to manage the cached Java objects. The generic nature of Java Object Cache makes it an ideal cache repository for higher level caches such as Web Object Cache, easing the development effort and reducing complexity.

ORCL00370730

- **Oracle Business Components for Java (BC4J)** is a Java and XML framework that enables productive development, portable deployment, and flexible customization of multi-tier database applications from reusable business components. BC4J is an application component framework that provides developers with a set of intelligent software building-blocks that cooperate to manage all of the common facilities required to do the following:
  - Productively author and test business logic in components that automatically integrate with relational databases
  - Flexibly reuse business logic through multiple SQL-based views of data, supporting different application tasks
  - Efficiently access and update the views from servlets, JavaServer Pages (JSPs), XML clients and thin-Java Swing clients
  - Easily customize application functionality in layers without requiring modification of the delivered application

  See Also: "Oracle Business Components for Java" in this chapter

Oracle Application Server TopLink

Oracle Application Server TopLink is an advanced object-to-relational persistence framework, suitable for a wide range of Java 2 Enterprise Edition (J2EE) and Java application architectures. OracleAS TopLink development tools and runtime capabilities reduce development and maintenance efforts, and increase enterprise application functionality. Use OracleAS TopLink to build high-performance applications that store persistent data in a relational database.

The following sections introduce OracleAS TopLink and include discussions on these topics:

- Advantages of OracleAS TopLink
- OracleAS TopLink Components
- Application Development with OracleAS TopLink
- OracleAS TopLink Architectures Overview

Advantages of OracleAS TopLink

Enterprise applications rely on Java-to-database integration to implement objects and logic. OracleAS TopLink enables developers to efficiently develop and refine enterprise applications. To fully understand OracleAS TopLink, you must understand the problems that enterprise application developers face and how OracleAS TopLink resolves them.

The OracleAS TopLink Problem Space

Java-to-database integration is a widely underestimated problem in enterprise Java applications. This complex problem involves more than reading from and writing to a database. The database world includes elements such as tables, rows, columns, and primary and foreign keys; the Java and J2EE world contains entity classes (regular Java classes or Enterprise JavaBeans (EJB) entity beans), business rules, complex relationships, and inheritance. Bridging these two fundamentally different technologies is a challenging and resource-intensive problem.

ORCL00370731

The process of translating object-oriented data into relational data is referred to as *object-relational* (O-R) *mapping*. To enable an O-R solution, developers must resolve the following O-R bridging issues:

- Fundamentally different technologies
- Different skill sets
- Different staff and ownership for each of the technologies
- Different modeling and design principles

Application developers need a product that enables them to integrate Java applications and relational databases, without compromising ideal application design or database integrity. In addition, Java developers need the ability to store (or *persist*) and retrieve business domain objects using a relational database as a repository.

The OracleAS TopLink solution is a persistence framework that manages O-R mapping in a seamless manner and enables developers to rapidly build applications that combine the best aspects of object technology and relational databases.

The OracleAS TopLink Solution

OracleAS TopLink provides a mature and powerful solution that addresses the disparity between Java objects and relational databases. OracleAS TopLink enables developers to:

- Persist Java objects in virtually any relational database supported by a JDBC 2.0 compliant driver
- Map any object model to any relational schema, using the Oracle Application Server TopLink Mapping Workbench graphical mapping tool
- Use OracleAS TopLink successfully, even if they are unfamiliar with SQL or JDBC, because OracleAS TopLink provides a clean, object-oriented view of relational databases

Other OracleAS TopLink Advantages In addition to providing industry-leading O-R mapping capabilities, OracleAS TopLink provides flexibility, increases performance and maximizes the productivity of your applications. OracleAS TopLink provides the following features:

- Advanced object caching that improves performance by minimizing database access.
- Rich query support that provides easy access to sophisticated, dynamic query languages and tools such as *query by example*, Java expression-based queries, EJB QL, and SQL.
- A transactional framework that enables developers to easily create and modify mapped objects. This framework integrates the complexities of a shared memory space and caches, and provides scalability that supports multiple server instances (clustering). Although the mechanisms involved are complex, OracleAS TopLink makes it easy to leverage this functionality by simplifying the task of writing transactional code that complies with database referential integrity and optimal access patterns.

OracleAS TopLink Components

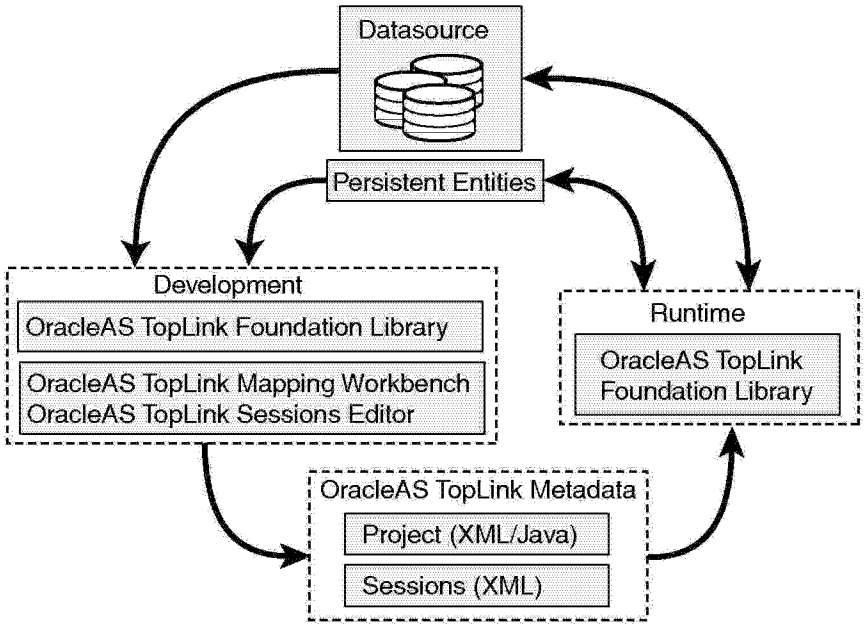
At its core, OracleAS TopLink is a runtime engine that provides Java or J2EE applications with access to persistent entities stored in a relational database. In addition to runtime capabilities, the Oracle Application Server TopLink Foundation

ORCL00370732

Library includes the OracleAS TopLink Application Programming Interface (API). This API enables applications to access OracleAS TopLink runtime features, as well as development tools that simplify application development. The tools capture mapping and runtime configuration information in metadata files that TopLink passes to the runtime.

Figure 2–6 shows how the OracleAS TopLink components relate to one another throughout the development cycle.

Figure 2–6 TopLink Components in the Development Cycle



OracleAS TopLink Development Components

OracleAS TopLink application development comprises three elements: the development environment, the OracleAS TopLink runtime, and the metadata that ties them together.

Development To create an OracleAS TopLink application, map the object and relational models using the OracleAS TopLink Mapping Workbench, and capture the resulting mappings and additional runtime configurations in the OracleAS TopLink project file (the `project.xml` file). Then build a session configuration file (the `sessions.xml` file) in the OracleAS TopLink Sessions Editor. These files together represent your entire OracleAS TopLink project.

During development, developers leverage the OracleAS TopLink API to define query and transaction logic. When developers use EJB entity beans, there is generally little or no direct use of the OracleAS TopLink API.

Runtime The OracleAS TopLink Foundation Library provides the OracleAS TopLink runtime component. Access the runtime component either directly through the OracleAS TopLink API, or indirectly through a J2EE container when using EJB entity beans. The runtime engine is not a separate or external process; instead, it is embedded within the application. Application calls invoke OracleAS TopLink to provide persistence behavior. This function allows for transactional and thread-safe access to shared database connections and cached objects.

ORCL00370733

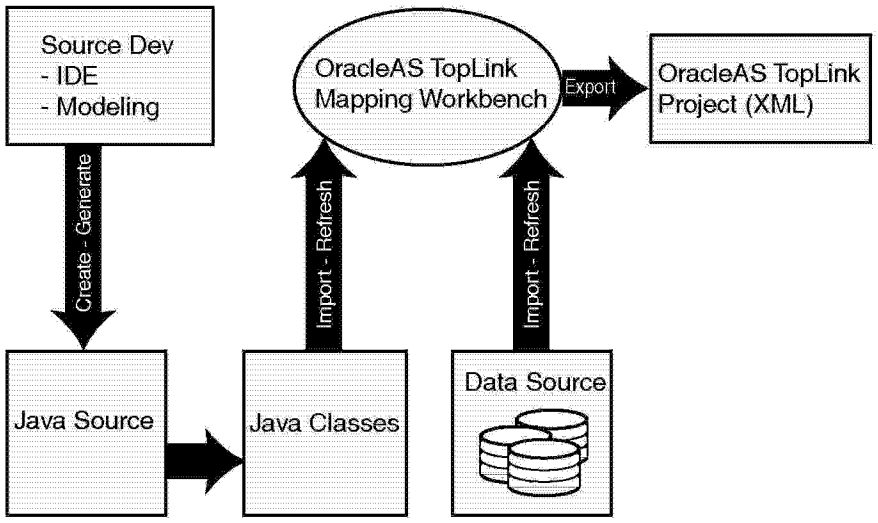
Metadata OracleAS TopLink metadata is the bridge between the development of an application and its deployed runtime. Capture the metadata using the OracleAS TopLink Mapping Workbench and OracleAS TopLink Sessions Editor, and pass the metadata to the runtime using deployment `project.xml` and `sessions.xml` files. It is also possible to hand-code these files using Java and the OracleAS TopLink API, but this approach is more labor-intensive.

The metadata, encapsulated in the `project.xml` file and the `sessions.xml` file, allows developers to pass configuration information into the runtime environment. The runtime uses the information in conjunction with the persistent entities (Java objects or EJB entity beans), and the code written with the OracleAS TopLink API, to complete the application.

Oracle Application Server TopLink Mapping Workbench

The OracleAS TopLink Mapping Workbench is a graphical development tool that enables developers to map between the object and relational models, and configure many of the OracleAS TopLink Foundation Library features. The OracleAS TopLink Mapping Workbench creates an OracleAS TopLink project, the primary object in the OracleAS TopLink metamodel. Export the project as a single deployment XML file (the `project.xml` file), which OracleAS TopLink uses in conjunction with the OracleAS TopLink runtime to provide the application-specific persistence capabilities. Figure 2–7 shows how the OracleAS TopLink Mapping Workbench fits into the OracleAS TopLink environment.

Figure 2–7 The OracleAS TopLink Mapping Workbench in a TopLink Environment



The OracleAS TopLink Mapping Workbench can import compiled entity classes (Java objects or EJB entity beans), as well as relational schema through a JDBC driver that the developer configures. Because OracleAS TopLink imports the object and relational models for mapping, developers can develop the two models relatively independently from the O-R mapping phase of project development.

Oracle Application Server TopLink Sessions Editor Most OracleAS TopLink applications include a session configuration file, the sessions.xml file, to simplify the application deployment process. The OracleAS TopLink Sessions Editor provides a graphical environment in which to configure the `sessions.xml` file.

ORCL00370734

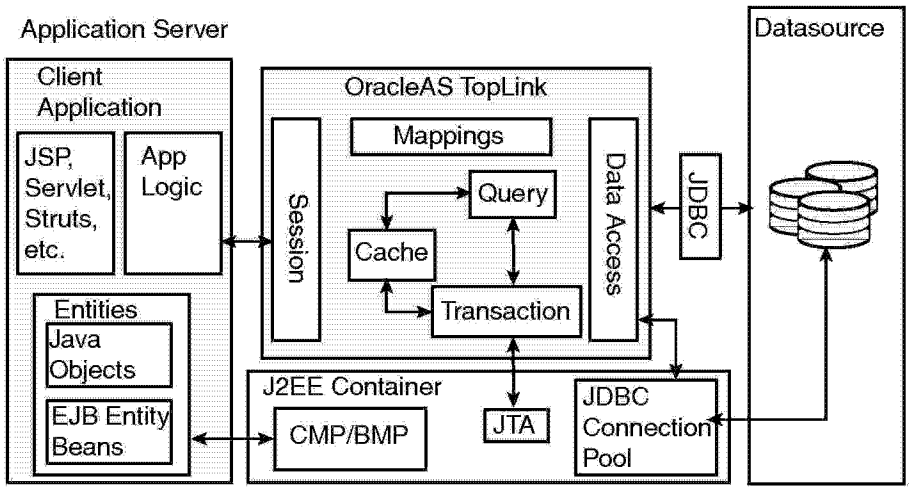
Use the `sessions.xml` file to configure one or more sessions for the OracleAS TopLink project, and associate the sessions with the project. This approach allows developers to specify individual configurations for each session and to add or modify the following:

- Database (JDBC) login information different from the login information used during development (for example, external data sources for the host application server's connection pools)
- JTA/JTS transaction usage
- Cache synchronization
- Session broker (enables client applications to view multiple databases and projects as a single OracleAS TopLink session)

Oracle Application Server TopLink Foundation Library

The Oracle Application Server TopLink Foundation Library includes a Java library that forms the runtime component of the product. It provides support and the API for the components that make up an OracleAS TopLink application. The API enables developers to interact with OracleAS TopLink to retrieve and modify their application persistent entities. Figure 2–8 shows the interaction of the parts of an OracleAS TopLink application.

Figure 2–8 OracleAS TopLink Application Components



Note: Although this section describes how these components fit into J2EE architectures, note that OracleAS TopLink also supports non-J2EE solutions. The *Oracle Application Server TopLink Application Developer's Guide* describes these solutions in more detail.

Sessions A session is the primary interface between the client application and OracleAS TopLink, and represents the connection to the underlying relational database. OracleAS TopLink offers several different session types, each optimized for different design requirements and architectures. The session manager configures and manages the session as a singleton within the application.

ORCL00370735

The most commonly-used session is the server session, a singleton session that clients access on the server through a client session. The server session provides a shared cache and shared JDBC connection resources. OracleAS TopLink also supports sessions for two-tier architectures, distributed applications, and multiple databases.

Data Access The OracleAS TopLink data access component provides access to JDBC connections through connection pooling, provided either by OracleAS TopLink or a host application server. This component manages the SQL generation required by the various query operations and reconciles any differences between JDBC drivers and SQL dialects. OracleAS TopLink offers many performance tuning options that optimize its data access capabilities.

Caching OracleAS TopLink supplies an object level cache that guarantees object identity and provides performance enhancement. Developers can configure the OracleAS TopLink cache and maximize the application efficiency by reducing the number of times the application accesses the database. In a clustered environment, developers can configure OracleAS TopLink to synchronize changes with other instances of the deployed application.

Queries The OracleAS TopLink query framework provides developers with the flexibility necessary to manage the complex persistence requirements of enterprise applications. The key features of this query framework include:

- A rich set of query types to allow object retrieval, summary results, and raw data retrieval
- The ability to specify the search criteria using OracleAS TopLink Expressions (for object model based queries), EJB QL, SQL, stored procedures, or query by example
- Configuration options that enable developers to specify how the query is executed, and customize many of its performance optimizing features

Developers can define OracleAS TopLink queries using the OracleAS TopLink Mapping Workbench, in Java code using the OracleAS TopLink API, or, in the case of EJB entity beans, through EJB Finders.

Transactions OracleAS TopLink provides the ability to write transactional code isolated from the underlying database and schema. OracleAS TopLink achieves this functionality through the Unit of Work.

The Unit of Work isolates changes in a transaction from other threads until it successfully commits the changes to the database. Unlike other transaction mechanisms, the Unit of Work automatically manages changes to the objects in the transaction, the order of the changes, and changes that might invalidate other OracleAS TopLink caches. The Unit of Work manages these issues by calculating a minimal change set, ordering the database calls to comply with referential integrity rules and deadlock avoidance, and merging changed objects into the shared cache. In a clustered environment, the Unit of Work also synchronizes changes with the other servers in the cluster.

If an application uses EJB entity beans, developers do not access the Unit of Work API directly, but they still benefit from its features: the integration between the OracleAS TopLink runtime and the J2EE container leverages the Unit of Work automatically.

JTA/JTS Integration: By default, OracleAS TopLink allows the application to create transaction boundaries for all object-level changes. OracleAS TopLink explicitly manages the database transaction, and if it encounters problems, safely rolls back both the database changes and the object-level changes.

ORCL00370736

In the case of a J2EE application, developers can configure OracleAS TopLink to synchronize with the JTA/JTS subsystem of the host application server. This feature allows an application to use container-managed transactions, rather than the default user-managed transactions.

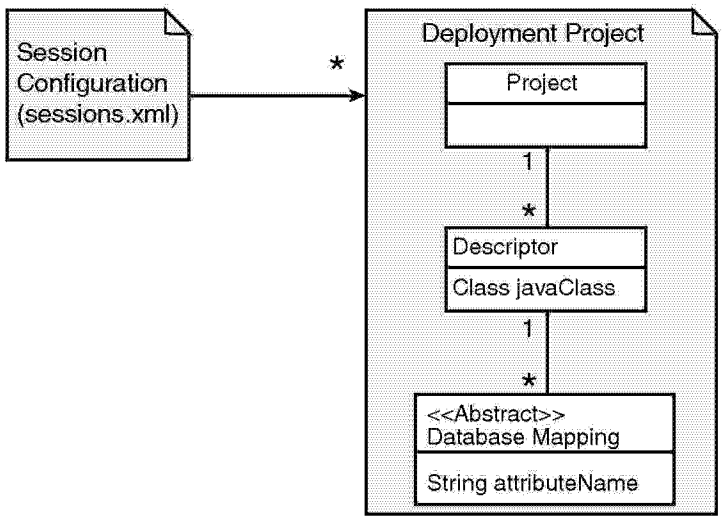
Note that this functionality is not limited to EJB architectures. Developers can configure any OracleAS TopLink architecture to use container-managed transactions.

OracleAS TopLink Metadata

The OracleAS TopLink approach to persistence is based on metadata that defines the class structure (objects) and relational schema, along with other configuration information used by OracleAS TopLink at runtime. Developers can use the OracleAS TopLink Mapping Workbench to define this metadata, and the OracleAS TopLink runtime component uses the metadata to provide the necessary persistence capabilities, using Java's reflective and introspective capabilities.

The TopLink application metadata model is based around the OracleAS TopLink project. The project includes descriptors, mappings, and various policies that customize the runtime capabilities. Figure 2–9 shows this metadata model.

Figure 2–9 OracleAS TopLink Metadata



Sessions XML Use the `sessions.xml` file to configure sessions for the project. Developers can build and edit these files with the OracleAS TopLink Sessions Editor. The session manager uses the `sessions.xml` configuration file during application initialization.

Project The OracleAS TopLink deployment project is the primary container for the metadata. A project generally represents an application and contains the mapping information for all persistent classes and their relationships. Each session (excluding the session broker) in the deployed application references a single project. Although developers can build a project by coding it using the OracleAS TopLink API, we recommend that developers create and manage the project in the OracleAS TopLink Mapping Workbench, and use the OracleAS TopLink Mapping Workbench to generate either an XML or Java source version of the project for use at runtime.

Descriptor A descriptor represents the association between a persistent Java class and a relational table or tables. The descriptor contains configuration information for the

ORCL00370737

class level within a project, as well as a set of mappings for each of its persistent attributes. Many of the more advanced configuration options are set at the descriptor level. The OracleAS TopLink Mapping Workbench supports most of these options, but there are a few that developers must set using the OracleAS TopLink API.

Mappings Mappings describe how the attributes of a mapped class are associated with columns in the database. OracleAS TopLink provides a sophisticated set of flexible and customizable mappings that allow for complex mapping scenarios between the object and relational models.

There are two types of mappings: direct mappings and relationship mappings.

- **Direct Mappings:** Direct mappings relate an attribute or attributes to a column or columns in the relational schema. OracleAS TopLink provides several direct mappings that allow for conversions between the types from the database and the object model's attribute types. Here are the direct mappings and their functions:
  - **Direct-to-field mappings** map a Java attribute directly to a value database column.
  - **Type conversion mappings** explicitly map a database type to a Java type.
  - **Object type mappings** match a fixed number of database values to Java objects.
  - **Serialized object mappings** store large data objects, such as multimedia files and BLOBs, in the database.
  - **Transformation mappings** offer specialized translations between how a value is represented in Java and in the database, such as when developers map multiple fields into a single attribute.
- **Relationship Mappings:** OracleAS TopLink offers sophisticated relationship mapping, which enables developers to represent object relationships based on the database table columns and foreign keys. Here are the relationship mappings and their functions:
  - **One-to-one mappings** represent simple pointer references between two Java objects. The references use any of foreign keys, target foreign keys, or variable classes to define the pointer.
  - **Aggregate object mappings** represent the relationship between a given object and a target object. The objects have a strict one-to-one relationship, and all the attributes of the second object are retrievable from the same table as the owning object.
  - **Aggregate collection mappings** represent the relationship between a single-source object and a collection of target objects. Unlike one-to-many mappings, in which there must be a one-to-one back reference mapping from the target objects to the source object, there is no back reference required for the aggregate collection mappings, because the foreign key relationship is resolved by the aggregation (object and collection).
  - **One-to-many mappings** represent the relationship between a single source object and a collection of target objects.
  - **Many-to-many mappings** represent the relationships between a collection of source objects and a collection of target objects. They require an intermediate table for managing the associations between the source and target records.
  - **Object-relational mappings** are mappings that leverage databases that support object-relational entity storage within tables.

ORCL00370738

Application Development with OracleAS TopLink

Using OracleAS TopLink to build an application does not affect the choice of development tools or the creative process. However, OracleAS TopLink does influence how developers approach development. This section highlights some of the key areas in which using OracleAS TopLink affects application development. These areas exist, regardless of whether developers are building an application to support Java objects, EJB entity beans, or both.

Mapping

OracleAS TopLink maps the application's persistent entities to the database, using the descriptors and mappings developers build with the OracleAS TopLink Mapping Workbench. The OracleAS TopLink Mapping Workbench supports several approaches to project development, including:

- Importing classes and tables for mapping
- Importing classes and generating tables and mappings
- Importing tables and generating classes and mappings
- Creating both class and table definitions with mapping creation and model generation

The OracleAS TopLink Mapping Workbench supports all these options; however, the most common solution is to develop the persistent entities using a development tool, such as an integrated development environment (IDE) or modeling tool, and to develop the relational model through appropriate relational design tools. Developers then use the OracleAS TopLink Mapping Workbench to construct mappings that relate these two models.

The OracleAS TopLink Mapping Workbench does offer some facilities for generating persistent entities or the relational model components for an application; however, these utilities are intended only to assist in rapid initial development strategies, rather than complete round-trip application development.

> **See Also:** *Oracle Application Server TopLink Mapping Workbench User's Guide*

Session Management

Sessions are the primary interface between the application and OracleAS TopLink persistence capabilities. When developing an OracleAS TopLink application, developers must ensure that they properly initialize and manage the sessions.

When using EJB entity beans with container-managed persistence (CMP) or bean-managed persistence (BMP), the client code that modifies the entity beans does not access the OracleAS TopLink session directly. Instead, changes occur transparently, through integration with the container or through EJB callbacks.

Well-designed applications that employ Java objects as persistent entities use the session manager provided in the OracleAS TopLink API. This class initializes and manages the singleton session. Developers configure the session manager in the `sessions.xml` file, which allows for easy configuration and customization of the deployed application.

Querying

OracleAS TopLink furnishes several object and data query types, and offers flexible options for query selection criteria, including:

ORCL00370739

- OracleAS TopLink expressions
- EJB QL
- SQL
- Stored procedures
- Query by example

With these options, developers can build any type of query. We recommend that developers use predefined queries to define application queries. Predefined queries are held in the project metadata and referenced by name. This simplifies application development and encapsulates the queries to reduce maintenance costs.

The OracleAS TopLink Mapping Workbench provides the simplest way to define queries. Developers can also build queries in code, using the OracleAS TopLink API.

If the application includes EJB entity beans, developers can code finders completely using EJB QL, which enables the application to comply with the J2EE specification. Alternatively, developers can use any of the other OracleAS TopLink query options. All querying options are available, regardless of the architecture or persistent entity type.

Transactions

In an OracleAS TopLink application, the Unit of Work ensures that OracleAS TopLink transactions comply with the transactional requirements of the application.

The Unit of Work is one of the most sophisticated and powerful components of the OracleAS TopLink Foundation Library. Although developers that use CMP or BMP entity beans do not use the OracleAS TopLink API to apply transactional changes to their persistent entities, the Unit of Work is used behind the scenes. Understanding how the Unit of Work behaves, and developing simple coding patterns to use it, are the keys to building efficient, maintainable applications.

Packaging and Deployment

Application packaging (for deployment in the host Java or J2EE environment) influences OracleAS TopLink use and configuration. For example, developers package a J2EE enterprise application in an Enterprise Archive (EAR) file. Within the EAR file, there are several ways to package persistent entities within Web Application (WAR) and Java libraries (JAR). How developers configure OracleAS TopLink depends, in part, on how they package the application and how they use the host application server class loader.

Monitoring and Performance Tuning

OracleAS TopLink enables developers to monitor functionality and performance throughout application development, testing, and quality assurance cycles. OracleAS TopLink offers many textual logging features, as well as the API required to implement custom logging strategies. Developers can use these features to ensure that the application behaves and performs as they expect.

OracleAS TopLink includes a performance profiler feature, available through the OracleAS TopLink Foundation Library API. This runtime feature tracks query execution time, which developers can use for performance analysis. This tool provides the information necessary to identify bottlenecks that hinder application performance.

OracleAS TopLink also offers a rich set of performance enhancement features. Understanding how to configure these features can have a strong influence on application performance, especially in the later phases of application development.

ORCL00370740

OracleAS TopLink Architectures Overview

OracleAS TopLink is designed to work in both Java and J2EE applications. Since it was first introduced, the flexibility OracleAS TopLink provides has led to its use in many architectural styles. This section introduces the five most common architectures associated with OracleAS TopLink. Although this section describes the architectures in relation to J2EE, OracleAS TopLink continues to fully support non-J2EE and Java applications as well.

Three-Tier

The three-tier (or J2EE Web) application is one of the most common OracleAS TopLink architectures. This architecture is characterized by a server-hosted environment in which the business logic, persistent entities, and the OracleAS TopLink Foundation Library all exist in a single Java virtual machine (JVM).

The most common example of this architecture is a simple three-tier application in which the client browser accesses the application through servlets, Java Server Pages (JSPs), and HTML. The presentation layer communicates with OracleAS TopLink through other Java classes in the same JVM, to provide the necessary persistence logic. This architecture supports multiple servers in a clustered environment, but there is no separation across JVMs from the presentation layer and the code that invokes the persistence logic against the persistent entities using OracleAS TopLink.

EJB Session Bean Facade

A popular variation on the three-tier application involves wrapping the business logic, including the OracleAS TopLink access, in EJB session beans. This architecture provides a scalable deployment and includes integration with transaction services from the host application server. Communication from the presentation layer occurs through calls to the EJB session beans. This architecture separates the application into different tiers for the deployment.

The session bean architecture can persist either Java objects or EJB entity beans.

EJB Entity Beans with CMP

OracleAS TopLink provides CMP support for applications that require the use of EJB entity beans. This support is available on the leading application servers. OracleAS TopLink CMP support provides the developer with an EJB 1.1 and 2.1 CMP solution transparent to the application code, but still offers all the OracleAS TopLink runtime benefits.

Applications can access OracleAS TopLink-enabled EJB entity beans using CMP directly from the client, or from within a session bean layer. OracleAS TopLink also offers the ability to use regular Java objects in relationships with EJB entity beans.

EJB Entity Beans with BMP

Another option for using EJB entity beans is to leverage OracleAS TopLink BMP in the application. This architecture enables developers to access the persistent data through the EJB API, but is platform-independent.

The BMP approach is portable—that is, after a developer creates an application, you can move it from one application server platform to another.

Two-Tier

A two-tier (or client-server) application is one in which the OracleAS TopLink application accesses the database directly. Although less common than the other

ORCL00370741

architectures discussed here, OracleAS TopLink supports this architecture for smaller or embedded data processing applications.

Oracle JDeveloper

Oracle JDeveloper is a J2EE and XML development environment with end-to-end support for developing, debugging, and deploying business applications and Web services. To maximize developer productivity, JDeveloper provides a comprehensive set of integrated tools to support the complete development life cycle, including the industry's fastest Java debugger and innovative profiler, and CodeCoach tools for code performance analysis and improvement. JDeveloper simplifies J2EE development by providing wizards, editors, visual design tools, and deployment tools to create high-quality, standard J2EE components, including applets, JavaBeans, JavaServer pages, servlets, and Enterprise JavaBeans.

To simplify the development of scalable, high-performance J2EE applications, JDeveloper offers an open and extensive J2EE framework called Oracle Business Components for Java (BC4J). BC4J is an object-relational mapping tool that implements the Sun J2EE design patterns, allowing developers to quickly build sophisticated J2EE applications.

Oracle JDeveloper Features

Oracle JDeveloper is written entirely in Java and is certified for use with all major platforms, including Windows, Solaris, and Unix. The following some key features of Oracle JDeveloper.

- **CodeCoach Utility and Profiling Tools:** To assist developers with optimizing Java code, JDeveloper provides a code coaching utility and three integrated profiling tools. With CodeCoach, you can increase the robustness of your code by optimizing system resources. The three profiling modes enable you to create a statistical analysis of the performance of your application with respect to it functionality, both at compile time and runtime. You can also analyze your application's memory use in the Java heap, and the occurrence and duration of various events.
- **Extensible and Pluggable Integrated Development Environment:** JDeveloper provides a public Extension SDK, enabling its development environment to be extended and customized. Using this SDK, customers and partners can develop extensions and integrate them directly into the JDeveloper IDE. Developers can also share the technologies written with the Extension SDK at the JDeveloper Extensions Exchange at:

  `http://otn.oracle.com/products/jdev/htdocs/partners/addins/exchange/content.html`
- **Support for SQL, PL/SQL, and XML:** In addition to Java, JDeveloper provides native support for SQL, PL/SQL, and XML. This support includes syntax highlighting and code insight, as well as PL/SQL development, PL/SQL debugging, and SQL tuning. Additionally, JDeveloper provides direct access to the database, allowing you to view, create, modify, and delete tables, views, triggers, indexes, sequences, and more.
- **Debugging Support:** JDeveloper offers robust debugging support for both Java and PL/SQL. Debugging in these two environments is seamlessly integrated when using an Oracle database, providing the ability to step from Java code directly into PL/SQL code within the same debugging session. Other features include remote

ORCL00370742

debugging, debugging multiple processes on multiple servers, tracking relevant data members in a smart data window, and a count of instantiations per class.

- **Business Intelligence Beans:** To allow the development of highly analytical business intelligence applications, JDeveloper has integrated Oracle Business Intelligence Beans (BI Beans) into its environment. BI Beans enable developers to build applications that take advantage of the new analytic capabilities available in the Oracle database. Using modular components and intuitive wizards, developers can build sophisticated BI applications quickly and easily.

Oracle Business Components for Java

Oracle Business Components for Java (BC4J) is a standards-based, server-side framework for creating scalable, high-performance J2EE applications. The framework provides design-time facilities and runtime services to simplify the task of building, debugging, customizing, and reusing business components. The applications developed with BC4J can be deployed on any J2EE platform and can be accessed from a wide variety of clients, including Web browsers, professional desktop clients, XML clients, and wireless devices.

The J2EE programming model provides developers with a set of core APIs and technologies for developing multi-tier applications that are scalable and portable. In addition to these APIs, developers often find themselves implementing design patterns to address some of the design issues that are common to most multi-tier applications. Understanding these design issues and coming up with solutions for them can be both challenging and time consuming. BC4J steps into this space by providing an application framework for J2EE developers that implements many common design patterns and provides out-of-the-box solutions for tasks like object-relational mapping, validation, data access, network traffic minimization, and user interface binding.

By eliminating the substantial coding and testing work related to these common development issues, BC4J lets application developers focus full-time on implementing business solutions. The benefits of this framework include reduced development cost, lower project risk, and shorter time-to-market.

Additionally, the BC4J framework provides flexibility by allowing application developers to override any of the automatic behaviors the framework provides. Using BC4J, developers can take advantage of built-in behavior while maintaining control over application functionality.

Oracle Business Components for Java Features

The following sections discuss key BC4J features.

Object-Relational Mapping

The Object-Relational (O/R) mapping facilities that BC4J provides make it easy to use Java objects to transparently store and retrieve data in a relational database. Wizards and UML modeling tools make it easy to reverse engineer database tables and views into Java objects. For developers who prefer to work with entity beans, you can also create business components mapped to EJB 2.0 local entity beans.

Validation Model

Enforcing business rules is a critical part of any application, and BC4J provides a framework for consistently developing and enforcing validation rules across your applications. With BC4J, rather than coding business logic into each client, the

ORCL00370743

developer writes it once in a reusable, domain-specific business component. Also, BC4J allows you to perform validation at different levels, depending on the sophistication of your business rules.

Scalability and Performance

The BC4J framework provides features such as resource pooling, network optimization, and disk spillover that are designed to enhance an application's performance and scalability. Over the standard EJB remoting architecture, BC4J implements additional optimizations to keep the number of network roundtrips between the client tier and the application tier to a minimum. Further, for JSP and servlet-based applications deployed as J2EE Web modules, BC4J provides unique performance and scalability features. These include declarative JDBC connection pooling, application module pooling, session state persistence, and disk spillover for large result sets.

Flexible Deployment

BC4J presents a productive programming model for application development that is tier-independent and completely based on J2EE standards. The framework enforces a strict logical separation of the client tier and the business logic tier, making it possible to deploy the same application in multiple configurations without modifying the application code. BC4J supports the standard J2EE deployment modes: J2EE Web module, EJB session bean, and local mode. With BC4J you do not have to choose your deployment platform beforehand, and you can switch platforms easily without making changes to the application code.

Rich User Interface Support

BC4J makes it easy to bind JSP, Java Swing, wireless, and other XML clients to business components. The tier-independent programming model ensures that the same application can be bound to different user interfaces without changing any code in the business logic tier.

For Java client development, JDeveloper provides JClient, a common client framework for building Java applications and applets for business components. JClient uses the model-view-controller architecture to allow you to bind standard Swing controls or any third party model-based controls to BC4J data sources. JClient also provides a number of composite controls, making it easy to design Java clients with richer functionality than the standard Swing controls provide. Using the JClient design-time wizards, you can quickly generate data-aware forms, which you can then customize with the JDeveloper user interface design tools.

For JSP and servlet applications, JDeveloper provides the BC4J Data Tag Library, and set of JSP 1.1 compliant tags for binding JSP pages to BC4J data sources. The BC4J data tags make it easy to design and build JSP database applications with complete control of both the HTML design and the database access operations. When using the data tags, the developer does not have to write copious amounts of Java code to interact with BC4J data or customize the user interface. The client JSP application can mostly be comprised of HTML and JSP tags, giving JSP developers the flexibility to use their favorite visual HTML editors in conjunction with JDeveloper to design the exact appearance of their JSP applications.

Layered Customization Support

Once you have developed and delivered an enterprise application, the person or company installing the software typically needs to tailor it to fit their needs. BC4J offers a simple, innovative way to do this. All framework components have their Java

ORCL00370744

implementation class cleanly separated from their XML component definition. Developers can extend both the Java class and the XML component definition to customize their applications. Further, the extended components can globally replace the original components across the application.

Tight Integration with JDeveloper

Oracle JDeveloper provides integrated design-time support for the BC4J framework. The wizards, editors, and other integrated design tools in JDeveloper make it easy to model, develop, test, deploy, and debug BC4J applications.

Integration with Oracle *inter*Media

In addition to standard object datatypes, BC4J now has the ability to bind to the four Oracle *inter*Media object types to provide multimedia retrieval and upload support. With this integration complete, you can use BC4J to access image, audio, video, and generic media content stored in Oracle *inter*Media databases, and deliver these digital media through UIX applications on Web browsers.

Oracle Application Server Web Services

To enable **e-businesses** to work effectively, the Internet needs to support a standards-based infrastructure that enables companies and their enterprise applications to communicate with other companies and their applications more efficiently. These standards should allow discrete business processes to expose and describe themselves on the Internet, allow other services to locate and invoke them, and provide a predictable response.

Web services drive this transformation by promising a fundamental change in the way businesses function and enterprise applications are developed and deployed. A Web service is a discrete business process that does the following:

- **Exposes and describes itself:** A Web service defines its functionality and attributes so that other applications can understand it. A Web service makes this functionality available to other applications.
- **Allows other services to locate it on the Web:** A Web service can be registered in an electronic Yellow Pages, so that applications can easily locate it.
- **Can be invoked:** Once a Web service has been located and examined, the remote application can invoke the service using an Internet standard protocol.
- **Returns a response:** When a Web service is invoked, the results are passed back to the requesting application over the same Internet standard protocol used to invoke the service.

Web services provide a standards-based infrastructure through which any business can do the following:

- Offer appropriate internal business processes as value-added services that can be used by other organizations
- Integrate its internal business processes and dynamically link them with those of its business partners

Oracle Application Server Web Services Architecture

Oracle Application Server Web Services run as **servlets** in the **OC4J servlet container**. This gives Web services the same **scalability**, **availability**, and load balancing facilities that all **J2EE** applications have in Oracle Application Server.

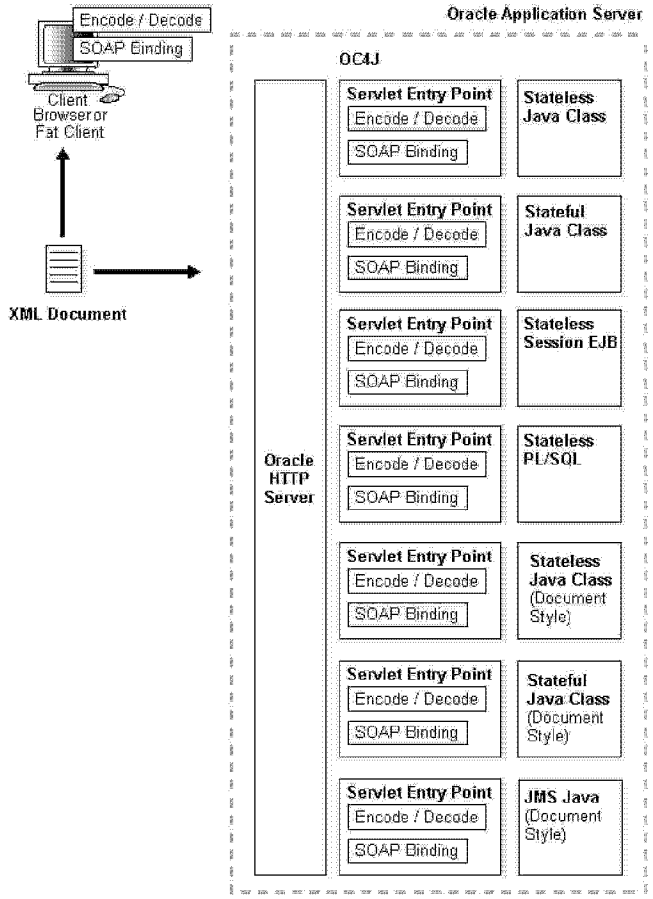
ORCL00370745

See Also: "Oracle Application Server Containers for J2EE Containers" on page 2-10

Oracle Application Server Web Services support both Remote Procedure Call (RPC) style exchange and message oriented, or Document Style, exchange. Supported RPC Web services include Java classes, stateless session EJBs, and stateless PL/SQL stored procedures. Supported Document Style Web services include Java Class Document Style Web Services and JMS Document Style Web Services.

Oracle Application Server Web Services use a different J2EE standards-compliant servlet for each implementation type to provide an entry point into the Web services implementation. Figure 2–10 illustrates the Oracle Application Server Web Services runtime architecture, including the servlet entry points.

Figure 2–10 Oracle Application Server Web Services Architecture



For a detailed discussion of the Oracle Application Server Web Services architecture, see the *Oracle Application Server Web Services Developer's Guide*.

ORCL00370746

Oracle Application Server Web Services Features

Oracle Application Server Web Services provides advanced runtime features and comprehensive support for developing and deploying Web services.

Oracle Application Server Web Services Development Features

Key Oracle Application Server Web Services development features include:

- **Development Environment:** Oracle Application Server Web Services allows application developers to implement Web services using J2EE components. In addition, you can use Java classes or PL/SQL stored procedures to implement Web services. Web services inherit all the runtime and lifecycle management elements of J2EE applications.
- **Development Tools and Wizards:** Oracle Application Server Web Services developers can use the same set of command line utilities to create, package, and deploy Web services that they use for other Oracle Application Server Containers for J2EE (OC4J) applications. In addition, Oracle Application Server Web Services provides the Web Service HTML/XML Streams Processing Wizard that assists developers in creating an EJB whose methods access and process XML or HTML streams.
- **Automatically Generates WSDL:** Oracle Application Server Web Services can generate Web Services Description Language (WSDL) and client-side proxy stubs. This generation occurs when the Web service is assembled using the Web Services Assembly tool or, for a deployed Web service, the first time the WSDL or the client-side stubs are requested. After the first request, the previously generated WSDL or client-side proxy stubs are sent when requested.
- **Registration, Publishing, and Discovery:** Oracle Application Server Web Services provides a standards-compliant UDDI registry where Web services can be published and discovered. The Oracle UDDI registry supports both a private and public UDDI registry and can also synchronize information with other UDDI nodes.
- **Developer Simplicity:** Using Oracle Application Server Web Services, developers do not need to learn a completely new set of concepts. Web services are developed, deployed, and managed using the same programming concepts and tools as J2EE applications.
- **Business Logic Reuse:** Application developers can transparently publish their J2EE applications to new Web services clients with no change in the application itself. The existing business logic developed in J2EE can be transparently accessed from existing J2EE/EJB clients.
- **Common Runtime Services:** Oracle Application Server has a common runtime and brokering environment for J2EE applications and Web services. As a result, Web services transparently inherit various services available with the J2EE container, including transaction management, messaging, naming, logging, and security services.

Oracle Application Server Web Services Deployment and Management Features

Key Oracle Application Server Web Services deployment and management features include:

- **Packaging and Assembly:** The Web Services Assembly Tool assists with assembling Web services and producing a J2EE `.ear` file.

ORCL00370747

- **Deployment:** Oracle Enterprise Manager provides a comprehensive set of facilities for deploying Web services to Oracle Application Server through Oracle Enterprise Manager Application Server Control. Application Server Control provides a single, consistent *Deploy Applications* wizard for deploying Web services to Oracle Application Server. It accepts a J2EE `.ear` file, and takes you through the steps to specify information about the application you are deploying, and then deploys the application.
- **Register Web Service:** The *Deploy Applications* wizard also provides access to facilities for registering Web services in the UDDI registry.
- **Browse the UDDI Registry:** The Oracle UDDI registry provides the UDDI standards-compliant, pre-defined, hierarchical categorization schemes. Oracle Enterprise Manager can drill down through these categories and look up specific Web services registered in any category.
- **Monitoring and Administration:** Once deployed, Oracle Enterprise Manager provides facilities to deinstall a Web service, and also to monitor Web service performance as measured by response time and throughput, and to monitor status as measured by up time, CPU consumption, and memory consumption. Oracle Enterprise Manager also provides facilities to identify and list all of the Web services deployed to a specific Oracle Application Server instance.

Oracle Application Server Forms Services

Oracle Forms applications combine interactive graphical interfaces with strong support for data validation. Forms developers can quickly create applications with powerful data manipulation features.

An Oracle Forms application gives you the power of a desktop application deployed in a browser. It provides a very rich and productive user interface by allowing the following:

- Immediate data validation as clients enter data into the form instead of after submitting the form
- Automatic completion and list of value searches for fields that enable users to enter correct information quickly

An Oracle Forms application is analogous to a traditional HTML form application. In the traditional application, the user enters data into a form on a Web page and submits the data. The Web server processes the data from the form in a CGI application. The application then performs any necessary transactions and returns feedback to the user's browser. JavaScript and the database perform data validation tasks to ensure that the data from the form is correct and complete.

Oracle Application Server Forms Services deploys Forms applications to Java clients in a Web environment. Oracle Application Server Forms Services automatically optimizes class downloads, network traffic, and interactions with the Oracle database. Applications are automatically load-balanced across multiple servers and, therefore, can easily scale to service any number of requests.

Oracle Application Server Forms Services consists of four components:

- **Forms Servlet:** The Forms Servlet is used to establish the initial connection when a user starts a Forms application. It returns to the client the HTML file that contains the Forms Client applet.
- **Forms Client Applet:** A Java applet running in the client browser. It provides the user interface for the corresponding Forms Runtime Process.

ORCL00370748

- **Forms Listener Servlet:** A servlet running in the OC4J servlet container. It manages the lifecycle of the Forms Runtime Process and network connections to the runtime process.
- **Forms Runtime Process:** A process running in an Oracle Application Server instance. The process maintains the connection between the client applet and the database.

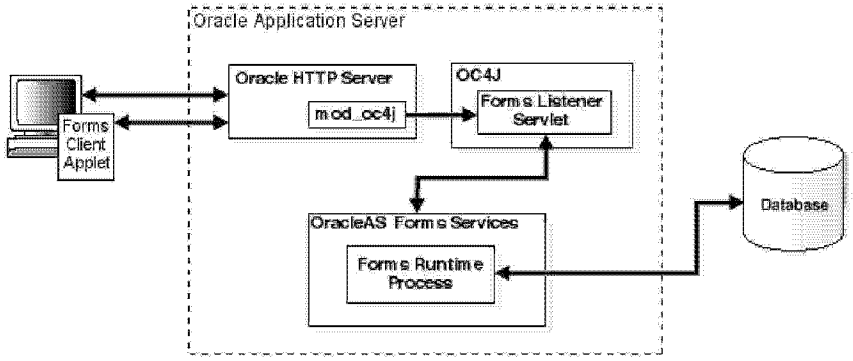
Oracle Application Server Forms Services Architecture

When a client requests an Oracle Application Server Forms application, the request has the following flow:

1. The user launches a browser and navigates to the URL for the Forms application.
2. The HTTP Listener interprets the URL and displays an HTML page containing an `<APPLET>` tag that describes the Forms Java Client to the browser. The URL calls the Forms servlet to create the HTML page dynamically.
3. The client receives the HTML file served by the HTTP Listener. The HTML file supplies the information required to locate the Java class files that make up the Forms Java Client. The HTML file also contains information about the requested Forms application, as well as parameters that get passed to the Forms session, such as login information.
4. The browser downloads the Forms Java Client and the Java applet starts.
5. The Forms Java Client applet sends a request to start a Forms session through the HTTP Listener to the Forms Listener Servlet.
6. After receiving the connection request from the Forms Java Client, the Forms Listener Servlet starts a new Forms Runtime process for the client.
7. The Forms Runtime process allocated to the client loads the module specified in the HTML file and any libraries and menus that are required by that form. All communication between the Forms Java Client and the Forms Runtime process is passed through the Forms Listener Servlet.
8. The user is prompted for database login information, and the connection to the database server is established.
9. The user is now ready to work with the Forms application.

Figure 2–11 illustrates this flow in terms of the Oracle Application Server Forms Services architecture.

Figure 2–11 Oracle Application Server Forms Services Request Flow



ORCL00370749

Oracle Application Server Forms Services Features

Oracle Application Server Forms Services provides the following features to allow a robust and standard deployment of Forms applications on the Internet.

Automatic Internet Scalability and High Performance

Oracle Application Server Forms Services is an optimized framework for deploying Oracle Forms applications to a multi-tiered environment. OracleAS Forms Services delivers out-of-the-box functionality and native services to ensure that Forms applications automatically scale and perform over any network. OracleAS Forms Services allows organizations to re-deploy existing applications to the Internet without changing the application code.

Native Services and Out-of-the-Box Functionality With OracleAS Forms Services, the scalability and high performance of Internet deployment do not rely on the quality of the application or on the developers' skills; applications developed with Oracle Forms automatically scale and perform through the use of built-in services that Oracle Forms provides. These services natively deliver the building blocks and the infrastructure of an e-business solution. Business developers do not have to implement complex, low-level infrastructure code and therefore can focus on implementing business requirements. Examples of these native services include record caching and record locking, database management, exception handling, and load balancing.

Application Server Integration As a fully-integrated component of Oracle Application Server, OracleAS Forms Services utilizes the Oracle Application Server Infrastructure. The deployment architecture of OracleAS Forms Services uses Oracle Application Server Containers for J2EE (OC4J). Codeless integration with Oracle Application Server Single Sign-On Server eliminates the need for users to remember multiple passwords. And leveraging the power of Oracle Enterprise Manager Application Server Control means that system administrators can use a single point of control to start, shutdown, monitor, trace, and diagnose OracleAS Forms Services runtime processes.

Network Traffic Optimizations OracleAS Forms Services is highly optimized to reduce network traffic and to minimize database response time. For example, OracleAS Forms Services renders application interfaces as highly-optimized generic Java applets and uses mechanisms such as message diffing, message bundling, and JAR file caching to reduce the number of round trips between the client and the middle tier. At the same time, OracleAS Forms Services leverages the capabilities of the Oracle database to minimize interactions with the database server.

HTTP/HTTPS Support The deployment with OracleAS Forms Services is optimized for corporate intranets, extranets, and for the Internet using socket connections or HTTP/HTTPS communications between the client and the middle tier. The support for HTTP/HTTPS allows Oracle Forms applications to be deployed across industry firewalls in a highly secure and optimized fashion.

Load Management Ensuring that your application has no single point of failure is essential to guaranteeing your application is highly available. The OracleAS Forms Services runtime architecture is based on standard servlets, meaning you can use standard load balancing techniques such as hardware load balancing, reverse proxy servers, and standard load balancing through Oracle Application Server.

ORCL00370750

Open Access to Internet Standards

Through code-based integration, OracleAS Forms Services gives you the flexibility to leverage emerging standards, including Java and XML, within your Oracle Forms applications. This gives you the advantages of both the application framework and built-in scalability and performance of OracleAS Forms Services, along with the flexibility to leverage Java and XML to extend your applications.

Server-Side Openness OracleAS Forms Services gives you the ability to access components written in Java and make use of them from Oracle Forms applications on the middle tier. Using this feature, OracleAS Forms Services is able to communicate and integrate with other applications or technologies, such as Web services, XML, Enterprise JavaBeans, and non-standard devices.

Client-Side Openness OracleAS Forms Services renders the application presentation as a generic and optimized Java applet. This applet can be extended using Java components (such as JavaBeans and Pluggable Java Components) so that business developers can quickly and easily deliver sophisticated interfaces that maximize end user productivity. The Java client extensibility, along with the new look and feel, leverages the strength of Java without the need for business developers to learn the language.

Oracle XML Developer Kit

XML makes data portable and interoperable across heterogeneous systems. XML is a metamarkup language that supports markup tags that have been defined by the user to encapsulate and describe data in a Web page. Because it is a self-defined language, it is extremely flexible.

Oracle Application Server includes the **Oracle XML Developer Kit** (XDK) to support development of any applications that use XML. The XDK provides basic XML infrastructure components for manipulating and transforming XML documents. The XDK contains component libraries and utilities that are based on World Wide Web Consortium (W3C) specifications. You can use these components to generate, manipulate, render, and store XML-formatted data in an Oracle database or to share data between applications written in diverse programming languages.

Oracle XML Developer Kit Tools

The following sections describe the tools included in the Oracle XML Developer Kit.

XML and XSLT Parsers

The XML parser **APIs** are defined by the W3C specifications so that application developers can use standard programming interfaces. However, if some functionality is not specified, then Oracle Corporation may implement its own enhancements to the specifications.

The Oracle XML and XSLT parsers provide international **character set** and multithreaded support, allow optional validation, and cache document type definitions (**DTDs**) and stylesheets for performance.

Oracle XML Developer Kit provides the following APIs:

- **Document Object Model (DOM) APIs:** Because an XML document is structured, in the sense that "start" tags have corresponding "end" tags and these tags are nested in an ordered fashion, an XML document can be viewed as a tree whose hosts consist of these tags and information between and corresponding to the tags.

ORCL00370751

When the DOM APIs are used to navigate an XML document, the XML parser parses the document and forms a tree representation of it in memory.

- **Simple APIs for XML (SAX) APIs:** SAX APIs are event-based, meaning that notification of certain events and data encountered during the parsing of an XML document can be reported with callback functions to the application program. One advantage over DOM is that an in-memory representation of the parse tree does not have to be built, thus saving memory and improving performance. When application programs are notified of these events, then they must handle them.
- **Namespace APIs:** These interfaces resolve namespace prefixes that are used to qualify element and attribute names in XML documents. XML document namespaces are identified by uniform resource identifier (URI) references that qualify element or attribute names and locate resources that may be on different machines or in different XML documents. Namespaces make it possible to have identical names for elements and attributes by qualifying them with URIs that differentiate the names.
- **Parser APIs:** Application programmers invoke the parser API to read an XML document and to provide access to its content and structure through DOM or SAX APIs. Typically, initialization and termination functions must also be invoked in association with the parse function. Various flags, such as to discard whitespace and turn on validation, can be set with initialization functions before the application program invokes the parse function.
- **XSLT APIs:** These interfaces read the input stylesheet file and transform the input XML document according to the stylesheet. A command-line interface to the integrated XSLT processor is provided. This allows application programmers to specify the number of threads to use to parse and transform XML documents and other useful options. To enhance performance, the XSLT engines allow caching of both stylesheets and Document Type Definitions (DTDs) so they can be reused for multiple XML documents.

XML Schema Processors

Oracle XML schema processors comply with the Structures and Datatypes sections of the W3C XML Schema Working Drafts, except for certain features (such as unique, key and keyref constraints) when the SAX parser is used. XML schemas are a superset of DTDs, except for the support of primitive and complex datatypes in XML schemas. This allows you to validate XML documents, with embedded datatype information, against XML schemas.

XML Class Generators

The XML class generators create a set of Java or C++ classes that create XML documents corresponding to a DTD or to XML schema definitions. This is useful when an application wants to send an XML message to another application based on an agreed-upon DTD, or as the back end of a Web form to construct XML documents. You can use the generated classes to programmatically construct XML documents that comply with the DTD or schema definition.

XSQL Servlet

XSQL Servlet is a tool that processes SQL queries and outputs the result set as an XML document. This processor uses an XML file with embedded SQL queries as its input. You can use XSQL Servlet to perform the following tasks:

ORCL00370752

- Build dynamic XML datapages from the results of one or more SQL queries and serve the results over the Web as XML datagrams or HTML pages using server-side XSLT transformations
- Receive XML posted to your Web server and insert it into your database

XML Transviewer Beans

XML transviewer beans are a set of XML components that constitute XML for Java beans. These are used by Java applications or applets to view and transform XML documents.

These beans are visual and non-visual Java components that are integrated into Oracle JDeveloper to enable the fast creation and deployment of XML-based database applications.

Oracle Application Server PL/SQL Platform

Oracle Application Server provides support for building PL/SQL-based applications on the Web. PL/SQL stored procedures retrieve data from a database and generate HTTP responses containing data and code to display in a Web browser. There are many techniques for coding dynamic pages, but PL/SQL is particularly suited for producing dynamic pages based on database processing. Its support for DML statements, dynamic SQL, cursors, and tight server integration provide both power and flexibility for Web applications. The dynamic pages can contain links and HTML forms that call more stored procedures, for example to drill down on the displayed data. The set of interlinked HTML pages forms the user interface of the Web application. Producing dynamic content using PL/SQL stored procedures gives you the flexibility and interactive behavior of CGI programs, without the memory overhead of starting a new CGI process each time.

Oracle Application Server PL/SQL Tools

Oracle Application Server PL/SQL developers can create applications using either Oracle PL/SQL Server Pages or Oracle PL/SQL Web Toolkit.

mod_plsql

Mod_plsql is an Apache plug-in that provides support for building PL/SQL-based applications on the Web. It maps browser requests into PL/SQL stored procedures, which retrieve data from a database and generate HTTP responses containing data and code to display in a Web browser.

Oracle PL/SQL Server Pages

To include dynamic content inside Web pages, including the results of SQL queries, you can use server-side scripting through PL/SQL Server Pages (PSPs). You can author the Web pages in a script-friendly HTML authoring tool and drop the pieces of PL/SQL code into place. You may find this technique more convenient than using the HTP and HTF packages to write out HTML content line by line.

Because the processing is done on the database server rather than the Web server, the browser receives a plain HTML page with no special script tags, and you can support all browsers and browser levels equally. It also makes network traffic efficient by minimizing the number of server roundtrips.

ORCL00370753

Embedding the PL/SQL code in the HTML page that you create lets you write content quickly and follow a rapid, iterative development process. You maintain central control of the software, with only a Web browser required on the client machine.

Oracle PL/SQL Web Toolkit

The PL/SQL Web Toolkit contains PL/SQL packages that allow you to create applications that generate dynamic HTML. Using the toolkit, you can access data and insert it into Web pages. The packages remove the need for you to know the specifics of HTML syntax and allow you to focus on writing applications.

Oracle Application Server PL/SQL Architecture

Oracle Application Server PL/SQL leverages the mod_plsql module to deliver Web content, as follows:

- Visiting a Web page, following a hypertext link, or submitting an HTML form causes the database server to run a stored procedure.
- Any choices that a user makes on an HTML form are passed as parameters to the stored procedure. Parameters can also be hardcoded in the URL used to invoke the stored procedure.
- The results of the stored procedure are printed as tagged HTML text and are displayed in the browser as a Web page. Web pages generated this way are dynamic: code runs inside the database server, producing HTML that varies depending on the database contents and the input parameters.

  This kind of dynamic content is different from dynamic HTML (DHTML). With DHTML, the code is downloaded as JavaScript or some other scripting language, and processed by the browser along with the HTML. A PL/SQL Web application can print JavaScript or other script code in its output to produce complex DHTML that would be tedious to produce manually.

Oracle Content Management Software Developer Kit

Oracle Application Server includes the Content Management Software Development Kit (CM SDK), a robust document lifecycle system that is built in Java and integrated with other Oracle content management API products. Oracle customers and partners have built successful solutions on this Java-based platform that leverages the features and capabilities of the Oracle platform.

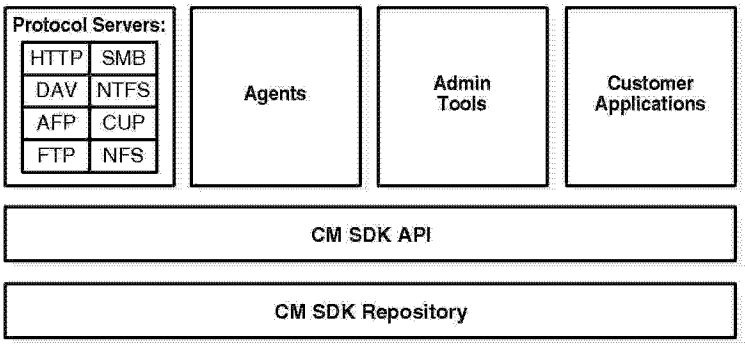
Oracle Content Management SDK Architecture

Oracle Content Management SDK provides, out of the box, the content and metadata repository, protocol servers, and class libraries that allow developers to quickly create content management solutions.

Figure 2–12 shows the basic architecture of Oracle Content Management SDK. Once installed, developers can immediately start storing content into the repository via the Web DAV, SMP, FTP, or IMAP interface. They can then provide the business logic for their specific requirements and create a Web-based user interface. Also included with the CM SDK installation is a Web starter application that demonstrates how to create Web-based content management applications.

ORCL00370754

Figure 2–12 Oracle Content Management SDK Architecture



Oracle Content Management SDK Features

Important features and capabilities of the CM SDK include the following:

- Integration with Oracle Text, Oracle Workflow, Oracle Ultra Search, Oracle *inter*Media, and OracleAS Portal
- Complete document lifecycle API with a variety of options for security, check-in/check-out, versioning, searching, extensible metadata, and other standard content management operations
- Easy extensibility to help you develop and deploy your content management solutions faster
- Managed through Oracle Enterprise Manager Application Server Control
- Synchronous and asynchronous trigger mechanisms to enforce business rules
- Built-in support for standard protocols such as HTTP/WebDAV, SMB/NTFS, FTP, AFP, NFS, IMAP4, and SMTP
- Support for standard Oracle performance, scalability, and high availability features

Oracle Application Server MapViewer

The Oracle Application Server MapViewer is a programmable tool for rendering maps using spatial data managed by Oracle Spatial or Oracle Locator. MapViewer provides tools that hide the complexity of spatial data queries and cartographic rendering, while providing customizable options for more advanced users. These tools can be deployed in a platform-independent manner and are designed to integrate with map-rendering applications.

There are five concepts that are fundamental to understanding MapViewer:

- **Style:** A style defines the rendering properties for features that are associated with styles. For example, a text style determines how a feature is labeled on a map, while a line style determines the rendition of a linear feature such as a road.
- **Theme:** A theme is a collection of features (entities with spatial and nonspatial attributes) that are associated with styles through the use of styling rules.
- **Base map:** A base map consists of one or more themes.
- **Mapping metadata:** Mapping metadata consists of a repository of styles, themes, and base maps stored in a database.

ORCL00370755

- **Map:** A map is one of the components that MapViewer creates in response to a map request. The map can be an image file, the object representation of an image file, or a URL referring to an image file.

When an application uses MapViewer, it applies specific styles, such as colors and patterns, to specific themes, such as cities, rivers, and highways, to render a map. For example, the application might display a map in which state parks appear in green and restaurants are marked by red stars. A map typically has several themes representing political or physical entities, or both. For example, a map might show national and state boundaries, cities, mountain ranges, rivers, and historic sites. When the map is rendered, each theme represents a layer in the complete image.

MapViewer lets you defines styles, themes, and base maps, including the rules for applying one or more styles to each theme. These styles, themes, base maps, and associated rules are stored in the database in map definition tables under the MDSYS schema, and they are visible through metadata views. All styles in a database instance are shared by all users. The mapping metadata that you can access is determined by MapViewer metadata views. The set of map definition objects that a given user can access is sometimes called that user's mapping profile.

Oracle Application Server MapViewer Architecture

With MapViewer, the basic flow of action involves two steps, regardless of whether the client requests a map or a MapViewer administrative action.

For a map request, the following occurs:

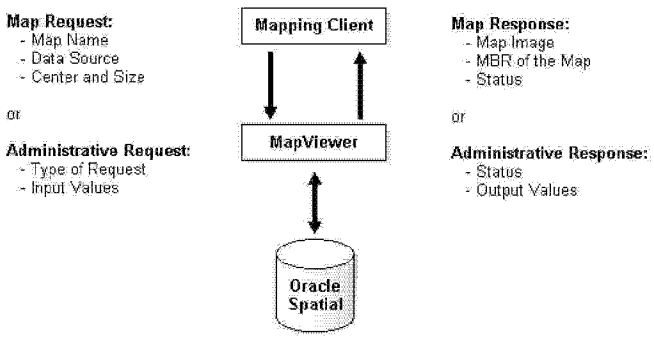
1. The client requests a map, passing in the map name, data source, center location, map size, and optional other data to be plotted on top of a map.
2. The server returns the map image (or a URL for the image), the minimum bounding rectangle (MBR) of the map, and the status of the request.

For a MapViewer administrative request, the following occurs:

1. The client requests a MapViewer administrative action, passing in the specific type of request and appropriate input values.
2. The server returns the status of the request and the requested information.

Figure 2–13 shows the MapViewer architecture, which supports this flow of action.

Figure 2–13 MapViewer Architecture



As shown in Figure 2–13:

- MapViewer is part of the Oracle Application Server middle tier.

ORCL00370756

- MapViewer includes a rendering engine.
- MapViewer can communicate with a client Web browser or application using the HTTP protocol.
- MapViewer performs spatial data access (reading and writing Oracle Spatial and Locator data) through JDBC calls to the database.
- The database includes Oracle Spatial or Locator, as well as mapping metadata.

Oracle Application Server MapViewer Features

MapViewer includes the following main components:

- A rendering engine (Java class library) that provides cartographic capabilities (a map renderer)
- An XML API that provides a programmable interface to MapViewer

The rendering engine connects to the Oracle database through Java Database Connectivity (JDBC). It also loads the map metadata from the database, and applies it to the retrieved spatial data.

The XML API provides high-level application developers with a convenient interface for submitting a map request to the middle-tier MapViewer and handling its map response.

ORCL00370757

ORCL00370758

Exhibit 20

Oracle9*i* Application Server

An Oracle White Paper
September 2002

ORACLE

ORCL00010730

EXECUTIVE SUMMARY

The Internet has played a vital role in transforming the way companies do business - enabling them to reach new customers and retain existing ones, implement new self-service business processes, streamline their supply chains - ultimately making them more efficient and lowering their costs. As companies have sought to make themselves more efficient by deploying their Enterprise Applications and Business Processes on the Internet, they face three fundamental systems challenges - (i) *Application Development Complexity* - The set of tools, standards, and technologies to develop business applications based on Internet technologies continues to evolve rapidly; (ii) *Application Deployment Complexity* - The infrastructure required to deploy and manage business applications and processes on the Internet needs to mature to meet mission-critical needs for performance, reliability, availability and security; (iii) *Middleware Cost and Complexity* - The number of pieces of application infrastructure, called "middleware," required to maintain new applications on the Internet and to web-enable existing applications continues to grow in both cost and complexity.

To resolve these challenges, Oracle recommends that Organizations design all Applications to Internet standards so that they can be deployed on corporate Intranets, Extranets or Internet. In doing so, Oracle recommends that organizations consolidate applications off desktop computers and multiple disparate pieces of middleware onto Application Servers - Applications will run more reliably at lower cost.

Oracle offers Oracle9*i* Application Server (Oracle9*i*AS) Release 2 - the second major release of its 100% standards-compliant Application Server to simplify how Internet Applications are developed; to make Internet Applications fast, reliable, manageable, and secure; and to substantially lower the Total Cost of Ownership associated with developing, deploying and operating Internet Applications. Oracle9*i*AS Release 2 has a number of new capabilities in three primary areas -

Most Productive to Develop Internet Applications - Oracle9*i*AS Release 2 provides the most productive Application Server environment for developers to develop and deploy Internet Applications. This includes Dynamic Web Sites, J2EE Applications, Web Services, Enterprise Portals, Wireless and Business Intelligence Applications. Further, Oracle9*i*AS Release 2 is 100% Standards Compliant supporting standards such as J2EE 1.3, Web Services, SOAP, WSDL, UDDI, RosettaNet 1.1/2.0, ebXML, WebDAV, LDAP v3, SSL v3, and several XML standards.

Most Reliable to Deploy and Manage Internet Applications - Oracle9*i*AS Release 2 provides the most reliable platform to deploy Internet Applications. This includes the industry's most optimized J2EE and Web Services environment; excellent scalability on single CPUs and multiple CPUs; several new high availability features including fault tolerance, clustering, online upgrade and maintenance; and integrated systems and security management for Internet applications and users.

Lowest Total Cost of Ownership - Finally, Oracle9*i*AS Release 2 is a single integrated product and designed to lower the total cost of ownership to develop, deploy, and operate Applications. Oracle9*i*AS Release 2 is a single product, not a single brand with multiple poorly integrated products offering users lower software costs, lower hardware costs, lower integration costs, lower support costs; and lower operating costs.

This technical whitepaper provides a detailed overview of Oracle's Vision for how Internet Applications should be developed, deployed, and managed; and provides specific details on the facilities in Oracle9*i*AS Release 2 to fulfill this vision.

EXECUTIVE SUMMARY 2

INTRODUCTION 5

E-BUSINESS INFRASTRUCTURE CHALLENGES 6

DEVELOPMENT CHALLENGES 6
DEPLOYMENT CHALLENGES 7
COST OF OWNERSHIP CHALLENGES 8

ORACLE9*I* APPLICATION SERVER OVERVIEW 9

ORACLE'S APPLICATION SERVER PRODUCT STRATEGY 9
ORACLE9*I*AS PRODUCT CAPABILITIES 11
ORACLE9*I*AS ARCHITECTURE 14

DYNAMIC WEB SITES, J2EE, AND WEB SERVICES 15

ORACLE9*I*AS APPLICATION DEVELOPMENT STRATEGY 15
ORACLE HTTP SERVER 15
DEVELOPING DYNAMIC WEB SITES 18
DEVELOPING J2EE APPLICATIONS 19
WEB ENABLING EXISTING APPLICATIONS 28

CREATE ENTERPRISE PORTALS 29

WIRELESS ENABLE INTERNET APPLICATIONS AND PORTALS 31

INTEGRATE YOUR ENTERPRISE 35

ORACLE9*I*AS INTEGRATION CAPABILITIES 36
COMMUNICATING WITH STAKEHOLDERS 38

EXTRACT AND ANALYZE BUSINESS INTELLIGENCE 39

ORACLE9*I*AS BUSINESS INTELLIGENCE STRATEGY 39
ORACLE9*I*AS CLICKSTREAM INTELLIGENCE 39
ORACLE9*I*AS PERSONALIZATION 40
ORACLE9*I*AS REPORTS SERVICES 41
ORACLE9*I*AS DISCOVERER 42
ORACLE9*I*AS WEB CACHE 44

RELIABILITY, AVAILABILITY, AND SCALABILITY 46

SCALABILITY 46
HIGH AVAILABILITY AND RELIABILITY 48
ORACLE9*I*AS CLUSTERING 52

ORCL00010732

MANAGE AND SECURE YOUR WEB INFRASTRUCTURE 53

ORACLE9*i*AS MANAGEMENT STRATEGY 53
MANAGEMENT, ADMINISTRATION AND MONITORING 54
SECURITY SERVICES 56
DIRECTORY SERVICES 58

PARTNER SOLUTIONS 59

SUMMARY 60

ORCL00010733

INTRODUCTION

The Internet has played a vital role in transforming the way companies do business - enabling them to reach new customers and retain existing ones, implement new self-service business processes, streamline their supply chains - ultimately making them more efficient and lowering their costs. As the Internet and corporate Intranets become almost universally ubiquitous, leading organizations around the world have utilized the Internet to bring their enterprise wide applications and business processes online, and have begun to use the Internet to build direct relationships with their suppliers, partners, government bodies and large customers. Today, these facilities continue to largely rely on traditional forms of business-to-business communication - fax, e-mail and EDI and really only addresses specific business functions and relationships - in effect "hardwiring" these specific interactions. In reality, most organizations still operate using information systems that limit their ability to exploit new market opportunities and to make themselves more efficient for competitive advantage. The true potential of the Internet is to change the e-Business landscape to create an environment where virtual enterprises can seamlessly collaborate and transact business - locating and syndicating services, sharing applications, and conducting electronic transactions without having to pre-define specific business functions, connections, clients or access points.

To be successful in this environment, enterprises must organize themselves around a set of fundamental business objectives - the desire to understand every customer to attract and retain them; the ability to have zero latency business processes that span a company and its supply chain partners; the facility to have up-to-the minute access to business intelligence to enable fact-based decisions - and design their information systems to use the Internet to meet these objectives. Leading organizations that are attempting to make this transition, however, face three fundamental systems challenges - (i) Application Development Complexity - The set of tools, standards, and technologies to develop business applications based on Internet technologies continues to evolve rapidly; (ii) Application Deployment Complexity - The infrastructure required to deploy and manage business applications and processes on the Internet needs to mature to meet mission-critical needs for performance, reliability, availability and security; (iii) Middleware Cost and Complexity - The number of pieces of application infrastructure, called "middleware," required to maintain new applications on the Internet and to web-enable existing applications continues to grow in both cost and complexity.

This has lead to the following dilemma for organizations - how does a company create a competitive advantage? By spending its dollars where it counts the most. Studies show that companies spend approximately 80% of their IT budget to provide standard business operations and only 20% on areas where they can differentiate themselves from the competition. This spending mix is driven by two factors - first, integrating diverse applications into a consolidated set of business processes is difficult to implement and difficult to maintain and manage; and second, organizations continue to spend enormous amounts of money in integrating the infrastructure required to run business applications rather than creating new business applications.

Oracle has a clear vision for organizations can use the Internet to fundamentally change how they do business. To enable this vision, Oracle believes that all business applications should be developed to open standards; be deployed on the Internet; and should be managed using a software infrastructure that eliminates the need to tie-together many different piece-part middleware components. This software infrastructure - Oracle9*i* Application Server - will allow organizations to develop applications faster, to deploy and manage them with greater reliability, and will lower the total cost of ownership for organizations.

This white paper provides a detailed technical overview of Oracle9*i* Application Server (Oracle9*i*AS) Release 2 and explains how it addresses the challenges discussed above. It is complimented by a number of other white papers that provide greater detail in a variety of specific product areas. It is organized into the following major sections:

ORCL00010734

- First, we discuss the primary e-Business Infrastructure Challenges facing organizations seeking to use the Internet to deploy mission critical applications and business processes
- We then provide an overview of Oracle's Vision to address these challenges and an overview of the Oracle9*i*AS Product Strategy
- Next, we discuss in detail the facilities in Oracle9*i*AS to Develop Internet Applications
- We follow this by discussing facilities in Oracle9*i*AS to Deploy and Manage these Applications
- Finally, we summarize the benefits of using Oracle9*i*AS as an e-Business Software Infrastructure

E-BUSINESS INFRASTRUCTURE CHALLENGES

Let us begin by first discussing the primary software infrastructure challenges facing organizations seeking to use the Internet to deploy mission critical applications and business processes. From Oracle's experience working with leading organizations around the world and in undergoing our own e-business transformation, these challenges typically fall into three broad categories:

- *Development Challenges* - Organizations face challenges in quickly and efficiently developing business applications based on Internet standards. Further, as more complex business applications are deployed on the Internet, application development may evolve to include other requirements such as integrating the application with other systems, making it accessible from an Enterprise Portal from a Wireless Device, and providing information about which users are accessing the Application.
- *Deployment and Management Challenges* - Organizations also face challenges in ensuring that the applications developed to the Internet model can be deployed and managed to meet mission-critical requirements of performance, scalability, availability, security, and manageability.
- *Cost of Ownership Challenges* - Finally, organizations need to continue to lower the cost of ownership associated with the software infrastructure necessary to develop, deploy, and manage Applications on the Internet.

Let us now consider each of these issues in greater detail.

Development Challenges

As organizations seek to redefine their business processes and implement new Internet-enabled ways of working, they must address several Application Development challenges:

- *Rapid Development Cycles based on Industry Standards* - Competitive pressures dictate that e-Business applications be designed and delivered with short turnaround times and provide vendor independence by leveraging industry standards such as Java2 Enterprise Edition (J2EE). Can you achieve this without significantly growing your software development staff?
- *Making Business Applications Accessible as Web Services* - Applications deployed on the Internet need to be accessed by both internal departments and business partners. To be seamlessly integrated with other applications, can you develop applications on a service-based architecture to enable them to communicate with other "Web Services?"
- *Integrating Applications with Existing Systems and Business Partners* - Existing Applications, Legacy Systems, and Data Sources need to be integrated into new business processes. The need for integration may span organizational boundaries. Can you accomplish this without rewriting your applications or building extensive customization code to connect with specific partners? Can you make the data you have in different systems available to all of your constituencies?
- *Simplifying Information Access* - As the number of Internet Applications proliferates, organizations need a single place for users accessing these applications to go to find information more easily - an organization's Enterprise Portal. Does your organization have too many Web sites? Can your

ORCL00010735

employees, customers, and business partners find the information that they need in a unified manner from these different systems?

- *Extending Information Reach to Mobile Devices* - Business users accessing these enterprise applications are increasingly conducting business out of the office. Up-to-the minute access to information is becoming ever more important. Can you make all the relevant business applications that your employees use available to any user, anytime, anywhere in the world, via a mobile device? As the number of different devices grows, can you accomplish this without creating multiple versions of the same applications and Web sites?
- *Delivering tailored experiences and extracting information from the Web Site* - Visitors accessing Web Sites provide valuable information about themselves, their interests, and how easy it is to use the Applications they access. Can you extract this valuable information to deliver tailored customer experiences and drive business decisions?

These are some of the primary challenges that organizations face in developing Internet Applications.

Deployment Challenges

Once Internet Applications have been developed, deploying and managing them with extremely high quality service brings another set of challenges.

- *Performance* - User loads on Internet-based Applications have grown dramatically as business critical applications are increasingly deployed on the Internet. Most organizations are plagued by complaints from their users and employees that their Applications do not perform adequately - are the performance of your Web Sites adequate? Can you make your applications run faster without redesigning and rebuilding them?
- *Scalability* - As the volume of users accessing Internet Applications grows, managing the scalability of the Applications without requiring enormous amounts of hardware becomes a necessity. Can you ensure your systems will scale when faced with growing numbers of users? Can you maintain excellent "quality of service" with growing demand?
- *High Availability* - Business-critical Internet Applications need to be highly available, up-and-running all the time, and accessible 24/7 from anywhere in the world. Can you keep your systems with zero "unplanned downtime" from system failures? Can you keep your applications up-and-running without taking down your systems for maintenance purposes - "planned downtime"?
- *Manageability* - For cost reasons, Internet Applications are frequently deployed on less expensive commodity hardware making comprehensive management of these systems from a centralized facility critical to overall cost of ownership. Can you manage your complete Web Site infrastructure and monitor system status and faults from one integrated console? Can you monitor end-to-end performance of your systems including that experienced by individual users from a single place?
- *Security* - Finally, with the move to self-service business processes and direct access by customers and business partners to business critical applications, security requirements are escalating in complexity. Can you ensure that your Internet Applications are secure and your e-business transactions are not compromised? Can you manage your enterprise's security from a single place and grant and revoke a user's access control privileges from a single console?

These are some of the important challenges that organizations face in deploying and managing Internet Applications.

ORCL00010736

Cost of Ownership Challenges

Finally, Information Technology (I/T) organizations continue to face increasing cost pressures - to reduce investment in new software projects and to lower costs associated with managing and operating e-Business Applications. There are three factors that need to be considered in addressing total cost of ownership:

- *Middleware Growing in Complexity* - First, the number of pieces of standalone "middleware" required to deploy and operate Internet Applications is growing rapidly - Application Servers, Integration Servers, Directory Servers, Security Servers, Portal Servers, and others. The cost associated with purchasing, operating and managing these different pieces of middleware is extremely high.
- *Applications Difficult to Develop and Deploy* - Further, this proliferation of specialized pieces of middleware causes further complexity in application development - users need to use many different tools to develop applications - and in application deployment - system administrators face reliability problems from these different infrastructure pieces and find it very difficult to diagnose and fix problems that stem from multiple pieces not working together.
- *Middleware Expensive to Maintain* - Finally, middleware complexity has lead to rapidly escalating total cost of ownership - software costs, hardware costs, integration costs, support costs, and maintenance and operating costs. Today corporate I/T organizations spend as much as 30% of their budgets purely on integrating different systems together without adding any new e-business capabilities.

Lowering the total cost of ownership for e-Business Software Infrastructure is a fundamental challenge for every organization today.

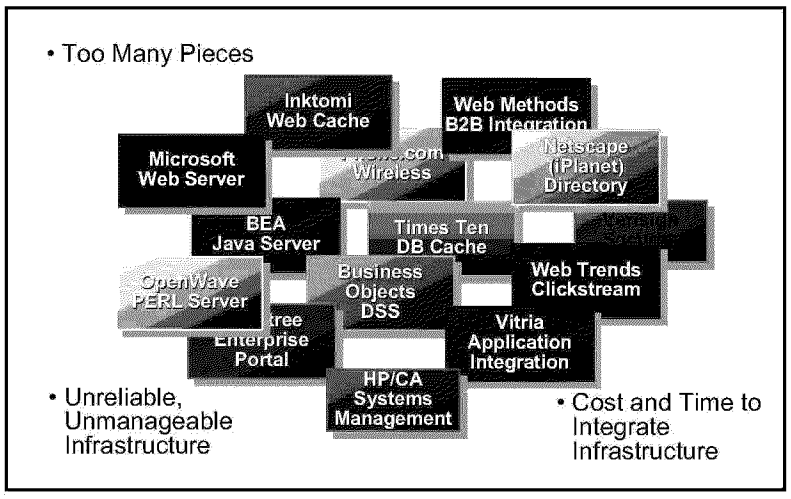


Figure 1- Middleware Complexity

Now that we have discussed the challenges facing organizations seeking to transform themselves into e-Businesses, let us examine how Oracle's Product Strategy addresses these challenges.

ORCL00010737

ORACLE9*i* APPLICATION SERVER OVERVIEW

To develop, deploy, and manage Internet Applications, Oracle offers Oracle9*i* Application Server - a 100% standards-based Application Server that is designed to address the challenges discussed above. Oracle9*i*AS simplifies how Internet Applications are developed; Oracle9*i*AS makes Internet Applications fast, reliable, manageable, and secure; and Oracle9*i*AS lowers the Total Cost of Ownership associated with developing, deploying and operating Internet Applications. Before we examine the specific capabilities of Oracle9*i*AS, let us first discuss Oracle's vision of a software infrastructure for e-Business Software and delineate our product strategy objectives in greater detail.

Oracle's Application Server Product Strategy

Oracle has designed its systems software to enable companies to develop and deploy Applications on the Internet model. In doing so, Oracle firmly believes that organizations can sharply reduce the cost and complexity associated with both developing, deploying, and managing this systems software by following two simple guiding principles:

- *Data Consolidation* - By consolidating data from multiple disparate data sources into fewer, database servers, users have access to better quality information at lower cost.
- *Application Consolidation* - Similarly, by consolidating applications off the desktop onto Application Servers, users have access to more reliable applications at lower cost.

Keeping these principles in mind, Oracle is also cognizant of several important trends in the way organizations are adopting the Internet and has a clear vision for how Internet Applications should be designed.

Oracle's Vision of Internet Applications

While there are several facets to Oracle's Vision for Internet Applications, our product direction is guided by three fundamental design principles:

- *Internet Becomes Ubiquitous* - Oracle believes that with the continued rapid growth of networks based on Internet standards - corporate Intranets within organizations, extranets that connect organizations with their business partners, and the public Internet itself - these networks will be ubiquitous. As a result, every organization will be reachable over the Internet, and every employee within these organizations will access information both within the organization and from external sources over the Internet. Oracle believes that the growth and reach of this global network based on a set of open standards provides a server platform on which all information and applications will be deployed. All applications will migrate off desktop systems and be deployed on Internet networks.
- *All Applications Deployed on Application Servers* - As applications migrate off desktops to be deployed on Internet networks, they need a server environment to run called an Application Server. Oracle believes that as applications deployed on Internet networks will be deployed in a three-tier architecture where information or content is managed in database servers; applications are managed in application servers; and access to this information is provided from an Internet browser that runs either on a desktop machine or on a mobile device. Further, Oracle also believes that as more complex business applications are deployed on the Internet, application servers will extend their capabilities to encompass facilities today offered by proprietary piece-part middleware solutions. This will make applications easier to develop, deploy, and maintain. This belief is also shared by many leading industry analysts who describe middleware converging on application servers.

ORCL00010738

- *Internet Becomes Programmable* - Finally, with all applications being deployed on Internet networks, and these networks becoming ubiquitous, Oracle believes that many applications will no longer run as standalone systems but will also communicate with each other over the Internet - the Internet will no longer be a medium for person-to-person communication (for instance, via e-mail), or for person-to-application communication (for instance, downloading information), but also for application-to-application communication. This communication will make business processes much more efficient by eliminating human intervention and will be accomplished by the adoption of a set of distributed computing standards based on Internet technologies that enable both existing applications and new applications to be accessed in a service-based architecture as "Web Services." These standards today involve an XML-based network protocol (SOAP - Simple Object Access Protocol), XML-based interface specifications (WSDL - Web Services Description Language), and service registries (UDDI - Universal Description, Discovery, and Integration). They will need to evolve to include other facilities such as messaging, transactions, security, composition or aggregation of services and others. Further, Oracle also believes that simple web services which are focused on point-to-point, request-response style connectivity between applications will be complimented by more complex web services with multi-party, long-running transactions (or business "conversations"), sophisticated security, such as non-repudiation and digital signatures, and as business-to-business collaboration and business process management.

The figure below summarizes Oracle's overall vision for how the Internet is evolving.

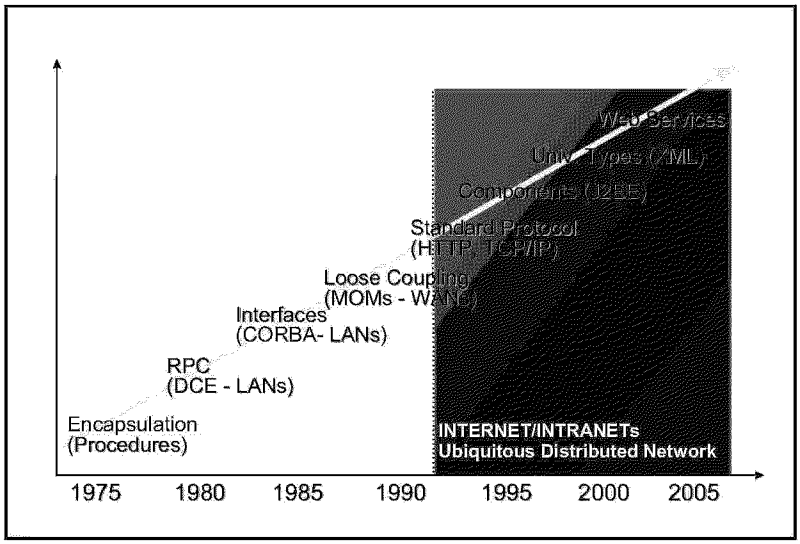


Figure 2 - The Evolution of the Internet

Oracle offers the second release of Oracle9*i* Application Server with three specific objectives - first, to help organizations develop applications that exploit this specific vision of Internet Computing; second, to address the challenges associated with developing and deploying Internet Applications discussed earlier; and third, to lower the total cost that organizations incur in developing, operating, and maintaining these applications. Let us now examine the specific product facilities of Oracle9*i*AS Release 2.

ORCL00010739

Oracle's Application Server Strategy

Oracle has designed Oracle9*i* Application Server Release 2 to provide a comprehensive infrastructure for e-Business Applications based on open standards. Specifically, Oracle has designed the product to address three specific objectives:

- *Most Productive to Develop Internet Applications* - Oracle9*i*AS Release 2 provides the most productive Application Server environment for developers to develop and deploy Internet Applications. This includes Dynamic Web Sites, J2EE Applications, Web Services, Enterprise Portals, Wireless and Business Intelligence Applications. Further, Oracle9*i*AS Release 2 is 100% Standards Compliant supporting standards such as J2EE 1.3, Web Services, SOAP, WSDL, UDDI, RosettaNet 1.1/2.0, ebXML, WebDAV, LDAP v3, SSL v3, and several XML standards.
- *Most Reliable to Deploy and Manage Internet Applications* - Oracle9*i*AS Release 2 provides the most reliable platform to deploy Internet Applications. This includes the industry's most optimized J2EE and Web Services environment; excellent scalability on single CPUs and multiple CPUs; several new reliability and failure detection facilities including fault detection, clustering, online upgrade and maintenance; and integrated systems and security management for Internet applications and users.
- *Lowest Total Cost of Ownership* - Finally, Oracle9*i*AS Release 2 is a single integrated product and designed to lower the total cost of ownership to develop, deploy, and operate Applications. Oracle9*i*AS Release 2 is a single product, not a single brand with multiple poorly integrated products offering users lower software costs, lower hardware costs, lower integration costs, lower support costs; and lower operating costs.

Now that we have explained Oracle's Vision for Internet Computing and the specific objectives of Oracle9*i*AS Release 2, let us understand its capabilities in greater detail.

Oracle9*i*AS Product Capabilities

Oracle9*i*AS Release 2 is a single integrated product that provides facilities to develop applications based on Internet standards and to deploy and manage them reliably. To understand its specific product facilities, let us look at its capabilities from the point of view of application development and deployment.

Application Development

Oracle9*i*AS Release 2 provides a productive Application Server environment for developers to develop and deploy Internet Applications based on open standards such as full support for **Java2 Enterprise Edition (J2EE)**, XML and Emerging **Web Services** standards. With Oracle9*i*AS, you can simplify access to these applications by creating **Enterprise Portals** that provide a single point for users to access these applications either from desktop browsers or **Wireless** Devices.

ORCL00010740

Through a comprehensive **Enterprise Integration** framework, modeling tools, Pre-built adapters and Web Services, you can redefine your business processes and integrate your Applications with legacy systems, databases, packaged applications, and business partners. You can deliver tailored 1:1 customer experiences via real-time Personalization, assess and correlate web site traffic patterns using Clickstream Analysis, and satisfy demands for up-to-the minute **Business Intelligence** about your applications using Oracle9*i*AS. The figure below illustrates the capabilities in the product.

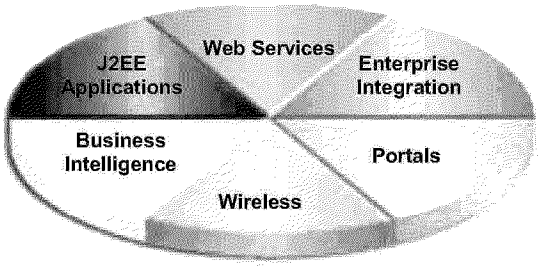


Figure 3 - Oracle9*i*AS Application Development

Specifically, Oracle recognizes that Oracle9*i*AS Release 2 offers the following specific facilities:

- *J2EE Facilities* - Oracle9*i*AS is (i) Integrated with the best-of-breed HTML Tools to design Web Sites; (ii) Provides facilities to create Dynamic Web Sites using Java Server Pages and Servlets; and (iii) Enables you to develop transactional Internet Applications using Enterprise JavaBeans.
- *Web Services* - Oracle9*i*AS enables you to (i) Publish J2EE Applications and existing legacy Applications as Web Services by generating WSDL interfaces; (ii) Invoke Web Services using the SOAP Protocol both synchronously and asynchronously; and (iii) Publish and Import Web Services from a UDDI Registry. Oracle9*i*AS allows you to integrate Web Services written in J2EE with those developed to the Microsoft's proprietary .NET specification.
- *Portals* - As the number of Web Sites and Web Services proliferate within your organization, Oracle9*i*AS allows you to create Enterprise Portals that offer a (i) Single Point to access information personalized to individual user needs; (ii) Single Point to Aggregate Information as "portlets"; and a (iii) Single Place to search for Information across the Enterprise.
- *Wireless* - Recognizing that users need to be able to access Applications all the time, Oracle9*i*AS allows you to (i) Make any Application, Accessible Across Any Wireless Network, and Any Gateway; (ii) Make these applications Accessible from Any Wireless Device; and (iii) To Exploit Wireless facilities such as Voice, Location-based Services, and Wireless Security to make applications easy to use.
- *Integration* - Oracle9*i*AS enables you to integrate Web Sites, Internet Applications, and Enterprise Portals with (i) Legacy Systems; (ii) Oracle and non-Oracle Databases; (iii) Packaged Applications; and (iv) Business Partners using a set of open standards such as J2EE, JMS, JCA, and XML messaging standards such as ebXML and RosettaNet.
- *Business Intelligence* - Finally, Oracle9*i*AS offers users comprehensive business intelligence about their applications - (i) Provide real-time "personalized" recommendations based on visitors behaviors; (ii) Analyze clickstream logs and extract intelligence about overall web traffic on Web Sites; and (iii) Generate Web-based Reports and Decision Analysis using powerful Business Intelligence Tools.

ORCL00010741

Application Deployment and Management

With Oracle9*i*AS, you can deploy and manage your Applications reliably. With Oracle9*i*AS, you can deploy any of these applications with excellent **Performance** and **Scalability**, reducing the amount of hardware you need to provide the same quality of service to users. You can further improve the performance, scalability, and availability of your systems through built in **Web Caching.** Your applications will remain **Highly Available** across system failures, and when software maintenance operations are carried out. Finally, you can implement a centralized **Systems Management**, **Security** and **Directory** framework to monitor, manage and secure all of your distributed systems and users.

Specifically, Oracle recognizes that Oracle9*i*AS Release 2 offers six specific facilities that are illustrated below:

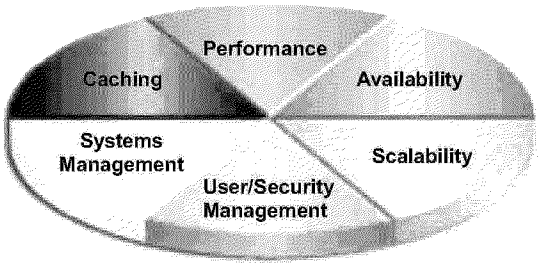


Figure 4 - Oracle9*i*AS Application Deployment

Lowest Total Cost of Ownership

Finally, Oracle9*i*AS Release 2 is designed to lower your Total Cost of Ownership (TCO) in developing, deploying and operating Applications by offering users a single integrated product. There are a number of features that unify the product:

- *Single Integrated Product* - Oracle9*i*AS is a single integrated product that is unique relative to competitors in a number of different ways. Users (i) Install and configure all of its elements as a single product; (ii) Secure it by defining users and their security privileges in a single place; (iii) Manage all the systems from a single management console; (iv) Globalize applications in a single way; (v) Use a common, consistent certified technology stack; (vi) Develop applications to a set of open standards - J2EE, Web Services and others; (vii) Have these applications integrate automatically into an Enterprise Portal; and (viii) Make these applications and services transparently accessible from Browsers and Wireless Devices. By unifying all these of elements into a single integrated product, Oracle9*i*AS lowers the cost associated with developing, deploying and operating applications.
- *Lowest TCO* - Oracle9*i*AS provides lowest cost of ownership through - (i) Lowest Software Cost due to Oracle9*i*AS' competitive price and more capabilities "In-the-Box"; (ii) Lowest Hardware Cost due to the improved scalability and caching facilities that allow you to serve more users with less hardware; (iii) Lowest Integration Cost due to a broader, better Integrated Product; (iv) Lowest Administration Cost since users manage 1 Product as opposed to a Bundle of different products; (v) and Lowest Support Cost due to a single support contract for these different capabilities. This technical white paper highlights the inherent value in leveraging a single e-Business platform so companies can minimize spending on I/T infrastructure and focus on their core competencies to reduce their total cost of ownership and create a competitive advantage.

ORCL00010742

Oracle9*i*AS Architecture

Oracle9*i*AS delivers a unique value proposition bringing you the broadest functionality offering from a single platform. Oracle9*i*AS has a layered architecture that consists of the following services:

- *Communication Services* - The entry point to Oracle9*i*AS from any client is a set of protocol handlers that manage communication for a variety of protocols including HTTP, RMI, RMI-over-IIOP, SOAP, IMAP 4, Wireless Protocols (such as SMS), and Web Cache that can serve a reverse proxy servers. All of these protocol handlers are designed to be deployed in a redundant, horizontal scaling model (described later in this document) with no single point of failure.
- *Application Runtime Services* - From the protocol handlers, the requests are routed to the Application Server Runtime that is a highly scalable and highly available J2EE Container. This container provides a common runtime environment for Applications developed as JSPs, Servlets, EJBs, and Web Services.
- *System Services* - The Application Server is itself designed to leverage a common set of runtime services that are necessary for J2EE Applications and Web Services - request dispatch and scheduling, resource management, resource pooling (including memory, connection pooling), clustering, fault monitoring, transaction management, and messaging).
- *Management Services* - To manage one or more instances of Oracle9*i*AS and to manage applications deployed on Oracle9*i*AS, a common set of systems management services to monitor the status, performance and faults of the system; to monitor resource consumption and usage; to manage a single instance or cluster of instances; to centrally administer security for users and applications; and to provide a comprehensive directory service framework to manage users.
- *Connectivity Services* - Applications can connect to a variety of Systems from Oracle9*i*AS including Oracle and non-Oracle Databases; Legacy Environments including CICS and EDI systems; any JCA Compliant Datasource including any Packaged Application; any Messaging System; and any Web Service.
- *Solutions* - Finally, Oracle9*i*AS does not just provide a J2EE Container and Web Services facilities but provides a comprehensive set of solutions all built on the infrastructure described above including Enterprise Portals, Enterprise Integration, Business Intelligence, Wireless, and ISV Solutions.

The figure below describes conceptually Oracle9*i*AS' Release 2's architecture.

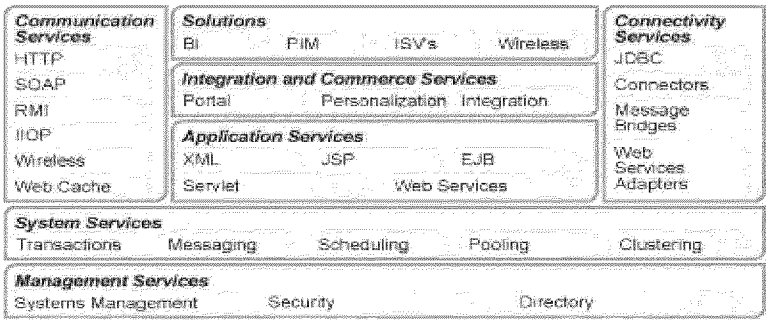


Figure 5 - Oracle9*i*AS Architecture

ORCL00010743

Now that we have provided an overview of Oracle9*i*AS' capabilities, we will examine each specific area in greater detail in the following sections:

- Dynamic Web Sites, J2EE Applications, and Web Services
- Enterprise Portals
- Wireless Access
- Enterprise Integration
- Business Intelligence

We follow these sections with a discussion of performance, scalability, high availability, systems management, and security.

DYNAMIC WEB SITES, J2EE, AND WEB SERVICES

Oracle9*i*AS Release 2 provides you with facilities to develop dynamic web sites; develop transactional e-business applications; and to make these applications accessible as Web Services.

Oracle9*i*AS Application Development Strategy

Oracle9*i*AS provides the industry's highest rated Application Server to simplify development of Internet Applications. Oracle9*i*AS supports development of the following kinds of applications:

- *Developing Dynamic Web Sites* - Oracle9*i*AS is seamlessly integrated with HTML Design Tools, and provides users with the ability to create dynamic web sites using J2EE Standards - Java Server Pages (JSPs) or Servlets - or using web scripting facilities in PERL or PL/SQL.
- *Developing J2EE Applications* - To develop Enterprise Applications, Oracle9*i*AS provides complete support for J2EE 1.3 APIs including JSPs, Servlets, and Enterprise JavaBeans. Oracle9*i*AS' J2EE Container interoperates with several best-of-breed solutions including Messaging Systems, Object-Relational Mappings, Security Services, Systems Management Facilities, Application Development Tools, and J2EE Development Frameworks.
- *Developing Web Services* - To publish existing legacy systems or new J2EE Applications as Web Services, Oracle9*i*AS provides complete support for Web Service standards including SOAP, WSDL, and UDDI. Oracle9*i*AS also provides support for complex Web Services standards such as ebXML.
- *Web Enabling Existing Applications* - Finally, Oracle9*i*AS Release 2 also offers facilities to integrate existing applications into this environment including - existing Forms Applications; existing Microsoft ASP Applications; and existing PL/SQL Applications.

  In this section we will examine these capabilities in more detail beginning with an understanding of Oracle's HTTP Server.

Oracle HTTP Server

Serving Web content requires a Web server, and serving content reliably to users requires a robust and reliable Web server. Oracle9*i*AS provide a Web server that can reliably serve requests to thousands of users in a scaleable and predictable manner.

Oracle HTTP Server Overview

Based on the industry leading Apache Web Server, Oracle's HTTP Server is HTTP listening entry point to Oracle9*i*AS. In Release 2, Oracle HTTP Server provides a completely configured, tested, and supported version of the Apache Web Server version 1.3.22. Oracle HTTP Server incorporates extended Apache functionality to provide SSL and HTTPS support. Oracle's HTTP Server dispatches requests to invoke program logic written in Java, PL/SQL, PERL, or as CGI executables via a standard Apache mod architecture.

ORCL00010744

With Release 2, Oracle9*i*AS provides a number of additional facilities that extend the capability of the Apache HTTP Server including:

- *Session Routing, IP Load Balancing* - Oracle's HTTP Server provides a number of session routing optimizations to route requests between the web server and specific instances of the J2EE container in a highly optimized fashion. Oracle also provides a number of facilities for more sophisticated IP load balancing.
- *Dynamic Port Registration* - To make large number of HTTP Server's easy to configure, manage and operate, Oracle9*i*AS' HTTP Server can dynamically identify and register a set of ports when started up or restarted, either as a single instance or within a cluster.
- *High Availability Facilities* - Oracle has added two facilities in Release 2 for high availability - (i) Fast Restart High Availability architecture designed to detect instance failure and automatically restart Web Servers eliminating downtime due to unplanned downtime; and (ii) Clustered Apache Servers that allow multiple HTTP Servers to share a single virtual IP-address, to load balance requests more effectively, and to be started, shut down, upgraded, and cloned as a single set of instances.
- *Oracle Specific Apache mods* - Finally, Oracle also provides a number of specific Apache mods including mod_OC4J (to invoke J2EE Applications); mod_PL/SQL (to invoke PL/SQL); mod_osso (for single sign-on); and mod_oraDAV (for Web DAV support). These are explained below.

Oracle HTTP Server Modules (mods)

Modules (mods) are plug-ins to the HTTP Server that extend its functionality either by offering native services (e.g. mod_ssl) or by dispatching requests to external processes (e.g. mod_plsql dispatching to the PL/SQL engine). mods are implemented in C code and run within the process space of the Oracle HTTP Server. In addition to the standard, compiled Apache mods provided with Oracle HTTP server, Oracle has enhanced several of the standard mods and has added Oracle-specific mods. Some of the important mods are:

- **mod_oc4j** : Mod_oc4j is new in Oracle9*i*AS Release 2. This module acts as a connector to route requests to and responses from Oracle9*i*AS Containers for J2EE (OC4J). Mod_oc4j uses Apache JServ Protocol (AJP) with binary format, providing more efficient communication and message header processing than HTTP. Mod_oc4j supports multiple OC4J instances, providing load balancing and failover across the instances. Routing information is maintained in user's cookies to ensure that stateful servlet requests are always routed to the same OC4J instance, and this eliminates any single point of failure in the OC4J load-balancer mechanism. OC4J is discussed in more detail later in this paper.
- **mod_osso** : Mod_osso is new in Oracle9*i*AS Release 2. This module enables the transparent use of Oracle9*i*AS Single Sign-On across all Oracle9*i*AS components. Through mod_osso, the HTTP Server becomes an Single Sign-On (SSO) partner application enabled to use SSO to authenticate users and obtain their identity, and to make user identities available to web applications as an Apache header variable. SSO is discussed in more detail in the Security section of this paper.
- **mod_ossl** : This module provides standard support for HTTPS protocol connections to Oracle9*i*AS. It enables secure connections between the HTTP server and a browser client by using an Oracle-provided encryption mechanism over a Secure Sockets Layer (SSL). It may also be used for authentication over the Internet through the use of digital certificate technology.
- **mod_fastcgi:** Oracle9*i*AS enables the execution of both CGI and FastCGI programs, which may be written in C, C++ or Java. While straight CGI programs result in the creation and tear-down of a process for every Web request, FastCGI provides for the reuse of these processes to greatly reduce the resource cost of running CGI.

ORCL00010745

- **mod_rewrite:** Oracle9*i*AS provides for the server-side rewriting of Web requests. While Internet infrastructures may be deployed in a variety of architectures based on security concerns, organizational factors, or other reasons, Oracle9*i*AS enables the maintenance of a simple and consistent Web site that hides this architecture from the end user.

Three other mods - mod_PL/SQL, mod_PERL, and mod_oraDAV are discussed later in this section.

Proxy Plug-In Support for Netscape and Microsoft IIS 

Many customers have standardized on either IIS (Microsoft's Internet Information Server), or on NES (Netscape Enterprise Server) as their corporate Web Server standard. With Release 2, Oracle offers a Proxy Plug-in to enable the use of these Web Servers with Oracle9*i*AS. The Proxy Plug-in provides users four important benefits:

- *No Separate Proxy Server Required* - Oracle provides a completely certified and tested proxy plug-in to eliminate the need for users to license a separate proxy server when using Oracle9*i*AS with other Web Servers. Additionally, Oracle's proxy solution is designed to invoke any program logic from Microsoft's and Netscape's Web Servers including J2EE Applications, PL/SQL, PERL, CGI and others.
- *Optimized Performance* - Further, Oracle's proxy plug-in provides facilities such as cookie propagation, socket connection keep alives (for instance, for HTTP 1.1 support), and session keep alives designed to optimize performance when routing requests to Oracle's Application Server from either IIS or NES.
- *Integrating ASP/ISAPI and NSAPI Applications* - The proxy also provides an efficient architecture for users to combine Microsoft ASP Applications or Applications designed to Netscape NSAPI interface with Applications deployed on Oracle9*i*AS as shown in the figure below.
- *Integrating Security* - Finally, the proxy is also integrated with Oracle's mod_osso facility so that Oracle's single sign-on security continues to function even with other Web Servers.

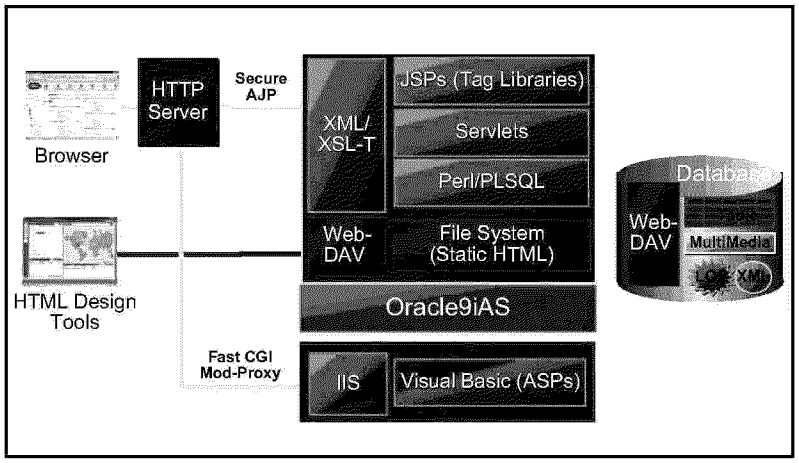


Figure 6 - Oracle Proxy Plug-in with Netscape and Microsoft Web Servers

ORCL00010746

Certification with Firewalls and Load Balancers

To ensure that Oracle9*i*AS' HTTP Server, Proxy Plug-in, and Application Server can be deployed in a variety of configurations including Internets with one or more firewalls, simple or complex DMZ architectures, and Intranets, Oracle has certified Oracle9*i*AS Release 2 with a number of the leading Firewalls and Load Balancers including - (i) Firewalls: Cisco PIX and Checkpoint Firewall 1; and (ii) Load Balancers: Cisco CSS, F5 BigIP, Nortel Networks, and Alteon. Based on certification and testing with these firewalls and load balancers in various configurations - a collection of Oracle9*i*AS Availability Best Practices have been published on Oracle Technology Network (technet.oracle.com). Note that certification with these solutions does **not** mean that Oracle's solution does **not** work with other firewalls or load balancers.

Developing Dynamic Web Sites

Complementing Oracle's robust HTTP Server, Oracle9*i*AS provides three important facilities to make developing content-rich, compelling, high performance, and highly available dynamic web sites easy and productive:

Integration with HTML Design Tools

Through support for an open repository standard called Web-based Distributed Authoring and Versioning (WebDAV), Oracle9*i*AS is seamlessly integrated with HTML Design Tools including Macromedia, Adobe, and Frontpage. This allows users to design HTML Pages and transparently save or deploy their templates/content to Oracle9*i*AS. Specifically, new in Release 2:

- *WebDAV and mod_oraDAV* - Oracle9*i*AS supports Web-based Distributed Authoring and Versioning (WebDAV). WebDAV is a protocol extension to HTTP 1.1 that supports distributed authoring and versioning. With WebDAV the Internet becomes a transparent read and write medium where content can be checked out, edited, and checked in to a URL address. Via an Apache mod called mod_oraDAV, Oracle9*i*AS Release 2 supports WebDAV to perform read/write activity to content stored both in local files and in Oracle databases (for instance in the Portal Repository or in the Internet File System). If the request is for content stored in the file system (not in an Oracle database), mod_oradav handles the access. If the request is for content stored in an Oracle database, the OraDAV API handles the access.
- *Pluggable Integration with HTML Authoring Tools* - WebDAV support makes Oracle9*i*AS pluggable with popular Web Authoring tools allowing users to seamlessly access files and database content from these tools. For example, Dreamweaver users can use WebDAV as the protocol for transferring files between a local folder on their development system to the remote site where their Web site is deployed on Oracle9*i*AS.

Web Page Scripting Support

To add dynamic content to static Web sites, Oracle supports Web Page scripting in a variety of scripting languages, most importantly - Java (JSP, Servlet,), PERL and for existing Oracle customers PL/SQL. Oracle also supports the ability to write web applications in C (CGI, FastCGI), and C++ (FastCGI).

- *Java Server Pages and Servlets* - Oracle provides a complete JSP and Servlet environment to make Web Page design extremely productive. JSP and Servlets are discussed later in this section.
- *Perl Interpreter, mod_Perl* - For users, wanting to script web pages in Perl, Oracle9*i*AS Release 2 provides a highly optimized Perl interpreter and runtime environment. The Perl runtime environment is linked into and embedded within the HTTP Server process, reducing the overhead of starting an external interpreter or making IPC calls to execute Perl scripts when they are requested. When the Oracle HTTP server receives an HTTP request for a Perl script, the request is routed to mod_perl, which passes the request to the Perl interpreter for processing.

ORCL00010747

The Perl interpreter returns the results of the request back to the HTTP server via mod_perl. The embedded Perl interpreter saves the overhead of starting an external interpreter process. Oracle's Perl environment also provides a code-caching feature, in which modules and scripts are loaded and compiled only once, allowing the server to run pre-loaded and compiled code, therefore incurring less overhead costs in repeated executions of the same Perl script.

- *PL/SQL and mod_plsql* - For existing Oracle users, Oracle provides the ability to create dynamic web pages using PL/SQL. Mod_PL/SQL is an Apache module that routes HTTP requests to PL/SQL stored functions and stored procedures in an Oracle database. The PL/SQL stored procedures generate dynamic content, using the PL/SQL Web toolkit, which is then returned to the browser via Oracle HTTP Server. Oracle also provides facilities to create rich web pages using PL/SQL Server Pages, which are analogous to JSPs, but uses PL/SQL to generate dynamic content.

Comprehensive XML Support

To customize Web Page presentation and delivery, Oracle provides a comprehensive set of XML libraries called the XML Developer's Kit (XDK).

- *XML Developers Kit* - The XDK contains XML component libraries and utilities that can be used to XML-enable applications and Web sites including the XDK for Java contains the (i) Oracle XML Parser, (ii) XSL Translator, (iii) XML Class Generator, (iv) XSQL Servlet which provides the results of SQL queries in XML form; and (v) XML Transviewer Beans. XDKs are also included for C, C++ and PL/SQL languages.
- *XML Standards Compliance* - These XDK's are compliant with the following W3C XML standards - XML 1.0, XML Namespaces 1.1, DOM 1.0/2.0, XSLT 1.0, XML Schema 1.0, SAX 1.0/2.0, SOAP 1.1, XPath 1.1, JAXP 1.0. Other new features in Release 2 include new XDK components that include an XML Parser, XSLT Processor, XML Schema Processor, XML Class Generator, and XDK Java Beans. Updated XDK Server components in Release 2 include an XML SQL Utility and XSQL Pages Publishing Framework.

Developing J2EE Applications

Oracle9*i*AS provides a fast, lightweight, highly scalable, easy-to-use and complete Java2 Enterprise Edition (J2EE) container written entirely in Java, which executes on the standard Java Development Kit (JDK) Virtual Machine (Java VM) available on the operating systems and hardware platforms on which Oracle9*i*AS is certified. Oracle9*i*AS Release 2 is J2EE 1.3 compliant. Before we go into specific details about the product's capabilities, let us first discuss Oracle's J2EE Product Strategy.

Oracle's J2EE Product Strategy

Oracle's J2EE Product Strategy is predicated on four specific objectives:

- *J2EE Standards Compliance* - First, Oracle is completely committed to supporting open J2EE Standards for application development. Oracle's Application Server will continue to provide the most up-to-date support for J2EE Standards. Oracle9*i*AS Release 2 provides complete support for J2EE 1.3 APIs including JSPs, Servlets, and Enterprise JavaBeans. Oracle9*i*AS Containers for J2EE (OC4J), supports Enterprise Java Beans (EJB) 2.0; Servlets 2.3 JavaServer Pages (JSP) 1.2; JTA 1.0; JNDI 1.2; JMS 1.0; JDBC 2.0 Extension; JavaMail 1.2, JAF 1.0, JAXP 1.1, Connector 1.0 (JCA) , and JAAS 1.0.
- *Interoperability with Best of Breed Products* - Recognizing that J2EE Applications will need to co-exist with a number of existing pieces of systems infrastructure, Oracle9*i*AS' J2EE Container interoperates with several best-of-breed solutions including Messaging Systems, Object-Relational

ORCL00010748

Mappings, Security Services, Systems Management Facilities, Oracle and non-Oracle Databases and others. These are listed in this document and in greater detail in the J2EE Technical Whitepaper.

- *Productive Application Development* - In addition, to make J2EE Application Developers highly productive, Oracle9*i*AS is designed with a number of features to improve productivity; and is integrated with the best-of-breed development tools and J2EE Application Frameworks.
- *Reliable Application Deployment* - Finally, Oracle is committed to providing users the fastest, most scalable, reliable and highly available platform to deploy J2EE Applications. A number of new facilities in Oracle9*i*AS Release 2 for RAS are discussed later in this document.

Now let us consider each of these capabilities in greater detail.

Java Servlets

A Java servlet is a program that extends the functionality of a Web server. A servlet receives a request from a client, dynamically generates the response (possibly querying databases to fulfill the request), and then send the response containing an HTML or XML document to the client. Oracle9*i*AS Release 2's Servlet engine has a number of facilities:

- *Support for Servlet 2.3* - Oracle9*i*AS' Servlet Engine provides complete support for the Servlet 2.3 specification including simple, complex and parameterized filters and chaining, application life cycle events, new class loading rules, advanced globalization/NLS support, full WAR file-based deployment, deployment descriptors, inter-JARs, auto-compile and auto-deployment of Servlets, and stateful failover and clustered deployment of Servlets.
- *100% Compatible with Tomcat* - Oracle9*i*AS' Servlet Engine is a 100% compatible with the Tomcat Servlet Engine delivered by the Apache consortium. As a result, developers who have used Apache and Tomcat to develop their applications can easily deploy them to Oracle9*i*AS' servlet engine. Oracle will ensure that its Servlet facilities remain compatible with subsequent Apache consortium efforts including the Jakarta and Catalina projects.
- *Complete Support for Filters* - Servlet filters are probably the most important new aspect of the Servlet 2.3 specification. Specifically, a filter is a component that is invoked when a client requests a resource that the Filter is mapped to such as a URL pattern or a Servlet Name. Normally a filter is used to wrap and manipulate a request, response, or header values before or after the execution of the original request target and does not itself return any response to the client. With Oracle9*i*AS, filters can be simple where they are configured to act upon a single request, complex where they act upon multiple requests, parameterised where they can receive input and output arguments; and chained where they can be placed into a chain where are a number of Filters can be invoked one after the other.
- *Full WAR file-based Deployment* - Servlets are packaged and deployed to J2EE Containers using a standard format called a Web Application Archive (WAR) file. Oracle provides WAR file Packaging and Deployment Tools.
- *Auto-Compile, Auto-Deployment of Servlets* - To make developers more productive, Oracle9*i*AS also provides automatic compilation of servlets and automatic deployment where the server receives a .WAR archive and automatically decompresses it and installs the application.

ORCL00010749

JavaServer Pages (JSP)

JavaServer Pages are a text-based, presentation-centric way to develop Servlets. JSPs allow Web developers and designers to rapidly develop and easily maintain information-rich, dynamic Web pages that leverage existing business systems. JSPs separate the content presentation from content generation enabling Web designers to change the overall page layout without altering the underlying dynamic content. Oracle9*i*AS' JSP Translator provides a number of features including:

- *Support for JSP 1.2* - Oracle provides a JSP 1.2 compliant translator and runtime engine including support for all JSP Directives and all core/standard JSP Tags, iterator support, and buffer management. It includes support for embedding SQL within Java using SQLJ and national language support (NLS).
- *Productive Development with Tag Libraries* - Oracle9*i*AS also provides a number of JSP Tag libraries to make developers productive including Tag libraries to interface with EJBs, to support connection pooling, to send and receive e-mail messages, to access files either in the file system or in Oracle's Internet File System, to embed XML result sets into JSP pages, to access a range of multi-media content, to access the personalization recommendation engine API, and to execute Web searches/queries.
- *Caching to Improve Performance* - Finally, Oracle's JSP Translator enables enhanced performance and efficiency by providing standard caching syntax that indicates whether a specific JSP tag is cacheable either in a shared Java cache (when additional XSL-T transformations for instance may need to be applied) or in the Oracle9*i*AS Web cache using JESI tags (Java Edge Side Includes). This substantially improves the performance and scalability of JSP Driven Web Sites.

Enterprise JavaBeans (EJB)

Enterprise JavaBeans (EJB) components are designed to encapsulate business logic, so that the developer does not have to worry about programming code for such mundane things as database access, transaction support, security, caching, and concurrency. In the EJB specification they are the responsibility of the EJB container. Oracle9*i*AS's OC4J provides a JDK-based EJB Container with complete support for the EJB 2.0 specification including full support for session beans, entity beans both Bean Managed Persistence (BMP), Container Managed Persistence (CMP), Message-driven Beans, and EJB Query Language. It includes support for dynamic EJB stub generation, full EAR file-based deployment, and simplified and automatic deployment of EJB applications including hot deployment. Oracle's EJB Container provides a number of facilities including:

- *Persistence* - BMP, CMP - For persistence, Oracle9*i*AS supports the EJB 2.0 standard entity-relational mappings (simple and complex mappings), providing facilities to auto-generate the mapping from EJBs to Database Tables. In addition to providing the EJB 2.0 mapping, Oracle9*i*AS Release 2 also provides integration with Oracle9*i*AS TopLink (TopLink) and with Oracle Business Components for Java (BC4J). TopLink provides advanced mapping capabilities including bean/object identity mapping, type and value transformation, relationship mapping, object caching and locking, batch writing and advanced and dynamic query capabilities and a GUI mapping tool.
- *Queryability via EJB QL* - Oracle9*i*AS provides full support of the EJB QL specification.[1]
- *Local Interfaces*[2] – Oracle9*i*AS provides support for local interfaces and container managed relationships between entity beans.
- *Transactions via JTA* - For transaction management, Oracle9*i*AS provides full support of the JTA specification. Oracle9*i*AS' JTA supports 1PC and 2PC, Container Managed and Bean Managed

[1] This feature will be available in a product update shortly after the initial delivery of Release 2.

[2] This feature will be available in a product update shortly after the initial delivery of Release 2.

ORCL00010750

transactions; and support of both Oracle and non-Oracle XA Compliant Resource Managers. For performance reasons, Oracle9*i*AS JTA also auto-detects transactions.

- *Messaging via JMS* - Oracle9*i*AS Release 2 supports Java Messaging Service (JMS). There are three important messaging capabilities that it provides - (i) A lightweight, volatile, database independent message provider (ii) Complete integration with Oracle's Advanced Queuing (OracleJMS) which supports both Publish/Subscribe and Request/Reply or Point-to-Point messaging. OracleJMS provides guaranteed, in-order, once-only delivery of messages over a variety of transports including HTTP, SMTP, FTP, SOAP, and Oracle Net Services (SQL*Net*). Oracle's Advanced Queuing facilities also support routing based on both topic and content of messages, and can support structured, unstructured, and XML payloads. Since messages with Oracle's Advanced Queuing are automatically persisted in the Oracle database, users can also query the state of the messages using standard SQL. (iii) Oracle9*i*AS JMS is also certified with support for third-party messaging systems such as IBM MQ Series, SonicMQ, and SwiftMQ.

- *Message Driven Beans (MDBs)* - New in the EJB 2.0 specification, MDBs are EJBs, which serve as asynchronous message consumers, namely EJBs, which are invoked not by the receipt of a synchronous RMI or RMI-over-IIOP message but via a JMS message. With Oracle9*i*AS Release 2, Oracle provides a complete MDB implementation. MDBs are certified with the new JMS infrastructure facilities including any certified JMS provider and can be invoked either in a publish-subscribe or request-reply fashion.

- *Active Components* - Further, note that the J2EE 1.3 framework does make it simple to develop applications that manage long-lived interactions between autonomous applications, such as B2B interactions. To address these issues, Oracle9*i*AS Release 2 introduces Active EJBs to declaratively specify loosely coupled applications, which are consistent, scaleable, and recoverable. Active EJBs are stateless session and entity beans with some additional, declaratively specified properties to support reliable asynchronous, disconnected interactions. Active Components also provide a host of automatic infrastructure capabilities including the ability to provide automatic message and exception routing and handling, automatic state management and state transition management, scheduling and sequencing, security propagation, restart, and load balancing. Oracle has submitted several elements of this specification to the JCP Standards process.

- *CORBA Interoperability*[3] - In Oracle9*i*AS Release 2, CORBA interoperability provides the capability to build J2EE Applications/EJBs and access them as CORBA services from CORBA clients. These facilities include support for the RMI-over-IIOP Protocol facilities and support for the Interoperable Naming, Security and Transaction specifications.

- *J2EE Connector Architecture via JCA* - J2EE Connector Architecture that defines a standard architecture for connecting J2EE Applications to heterogeneous Enterprise Information Systems (EISs). Examples of EISs could include ERP applications, mainframe transaction processing middleware, database systems, and legacy applications. Specifically, JCA consists of the following key components - Resource Adapters that connects an enterprise information system resource to the JCA infrastructure; a Common Client Interface (CCI) that provides a common programming interface model for interacting with enterprise information systems via resource adapters; and Quality of Service Facilities – services provided by the JCA infrastructure implementation to address connection management, transaction management, and security. Oracle9*i*AS Release 2 supports the J2EE Connector Architecture (JCA) 1.0 standard.

- *Oracle, Non-Oracle Database Access via JDBC* - Oracle9*i*AS provides Type 2 and Type 4 JDBC Drivers to access Oracle 8.0, 8*i*, and 9*i* databases including 9*i* Real Application Clusters. In addition to Oracle's own Oracle JDBC drivers, Oracle9*i*AS Release 2 includes DataDirect

[3] This feature will be available in a product update shortly after the initial delivery of Release 2.

ORCL00010751

Connect Type 4 JDBC drivers, to provide highly scalable and reliable connectivity to non-Oracle data sources. Connect JDBC drivers deliver very high performance drivers that support the J2EE 1.3 specification. Data sources include - (i) DB2 UDB 6.1, 7.1, 7.2; (ii) DB2 UDB OS/390 ver 5, 6.1, 7.1; (iii) Informix Dynamic Server w/ Universal Data Option 9.x; (iv) Informix Dynamic Server 2000 (9.20); (v) Microsoft SQL Server 7.0, 2000; and Sybase (Adaptive Server version 11.5, 11.9, 12.0, 12.5)

- *Security via Java Authentication and Authorization Services (JAAS)* - JAAS, part of the J2EE 1.3 specification, extends the security architecture of the Java™ 2 Platform with additional support to authenticate and enforce access controls upon users. Oracle9*i*AS Release 2 provides a complete JAAS implementation. Oracle's JAAS Provider is used to enforce security constraints for Servlets, JSPs, EJBs, and Web Services. (i) It provides facilities to authenticate users (identify users), authorize them with access control privileges (limit what they can do), delegate privileges (enabling code to run securely, with privileges of other users). Applications can retrieve roles for users, and roles can be hierarchical (that is, a role can be assigned to other roles). (ii) Principal and code-based policies can require that a user only assume particular privileges when accessing particular Java applications from particular locations, allowing Java developers to fine-tune their security policies. Privileges can be specified in a fine-grained manner to a class or method level. (iii) A J2EE Application may be configured to run with the permissions associated with the current client, or as a specified user (e.g. run as "DBAdmin"). This allows developers to enforce "least privilege" in their applications, allowing users only those privileges needed to perform a function, because users can only exercise privileges in the context of a well-formed business rule (e.g. an enterprise bean). (iv) The JAAS Provider can store and manage user information - identity and access control privileges - either in an LDAP Directory (Oracle Internet Directory) or in an encrypted file system.

As described above, Oracle9*i*AS provides comprehensive support for J2EE Application Development.

Java Development Tools

Oracle9*i*AS provides a number of facilities to improve productivity of J2EE developers.

Oracle9iAS Facilities - Oracle9*i*AS' J2EE Container has been designed to improve developer productivity in a number of ways - (i) First, it has a small disk and memory footprint and is implemented in pure Java making it possible to run it in-process with a Java IDE for development and debugging purposes. (ii) The entire system is configured using simple XML configuration files. (iii) There are a number of facilities to make the edit-compile-package-deploy-debug process extremely efficient including auto-compile of JSPs and Servlets, auto-deployment of applications to the container, hot-deployment which eliminates the need to restart the container when applications are deployed; and auto-generation of finder methods, O-R mappings, and other elements. (iv) Oracle offers a large number of command line tools for WAR and EAR file packaging, CMP mapping, and has made extensions to ANT (Java make). (v) Pre-Packaged Tag Libraries, JSP Source level debugging and other facilities make JSP Developers very productive.

The product is also seamlessly integrated with Oracle9*i* JDeveloper and with a number of leading HTML, Java and UML development tools.

- *Oracle9i JDeveloper* - Oracle9*i* JDeveloper is a J2EE development environment with end-to-end support for developing, debugging, and deploying e-business applications. JDeveloper empowers users with highly productive tools, such as the industry's fastest Java debugger, sophisticated performance profilers, integrated software configuration management, built-in UML modeling and an innovative J2EE framework, Business Components for Java, for building applications and Web services based on industry standard J2EE design patterns. JDeveloper provides a rapid application development environment based entirely on industry standards. (i) For Java

ORCL00010752

developers, the latest J2EE APIs are supported including graphical tools for building Enterprise Java Beans, Java Server Pages and Servlets. (ii) For XML developers, an XML Schema driven code editor offers guided XML editing and XML processing tools include an XML parser, XSLT Processor and XML Schema Processor. (iii) For SQL developers, a database browser enables inspection of any SQL92 compliant database and development of PL/SQL in Oracle databases. (iv) For Web Service developers, wizards provide publishing and consumption of J2EE applications as Web Services using SOAP, WSDL and UDDI.

- *3rd Party Development Tools* - As a result of Oracle's commitment to the open standards and to developer productivity, Oracle has actively worked with a number of development tools vendors to integrate their solutions with Oracle9*i*AS. Since it's initial release, Oracle9*i*AS has been gaining momentum in the market place and today there is a growing number of vendors supporting Oracle9*i*AS - the most up-to-date list of certified tools can be found from Oracle Technology Network. Today the tools fall into three categories - (i) HTML Design Tools -: Macromedia Ultradev, Adobe GoLive, and Microsoft Frontpage; (ii) Java Development Tools - Borland J-Builder, WebGain Visual Café, Sitraka JProbe, TogetherSoft Together ControlCenter, Neuvis NeuArchitect, Forte for Java, Computer Associates Cool Joe, Empirix, Cacheon, Compuware and several others; (iii) UML Modeling Tools - Rational Rose, TogetherSoft Together ConrolCenter, and Oracle9*i* JDeveloper.

J2EE Developer Frameworks

Finally, Oracle9*i*AS also certifies and supports a number of J2EE Developer Frameworks to ensure that users can quickly and productively build J2EE Applications. These include:

- *Oracle9i Business Components for Java* - Business Components for Java (BC4J) is a 100% Java and XML framework that enables productive development, intuitive object relational mapping, portable deployment, and flexible customization of multi-tier, J2EE database applications from reusable business components. The framework is based on industry standard J2EE design patterns and is designed to enable developers to consistently deliver top quality code and high performance applications. Developers can use the BC4J framework to: (i) Author and test business logic in components that automatically map to relational databases and can be exposed as Web services; (ii) Reuse business logic through multiple SQL-based views of data that support different application tasks; (iii) Access and update views from Java servlets, JavaServer Pages, and thin Java Swing clients; (iv) Tailor application functionality by customizing XML metadata without having to re-compile the application. BC4J runtime components can be deployed to Oracle9*i*AS or any other standard Java Virtual machine a set of Java classes, Enterprise Java Beans or CORBA components. JDeveloper bundles a extensive set of JSP tag libraries and Swing controls for rapid construction of BC4J client applications. BC4J applications may be called via IIOP, HTTP, or by direct Java method calls when running locally. Release 2 includes a number of new BC4J scalability enhancements providing more fine-grained configuration of your application module pool and connection pools and runtime enhancements including support for discriminator attributes, polymorphic view objects, and support for Oracle object types.
- *Apache Frameworks* -  For developers who want to use J2EE Frameworks from the Apache community, Oracle also certifies and supports Apache Structs, Cocoon, and Turbine with Oracle9*i*AS Release 2.
- *Oracle9iAS MVC Framework for J2EE* -  Oracle9*i*AS MVC (Model-View-Controller) Framework for J2EE uses UML (Unified Modeling Language) modeling and declarative techniques to provide a common and systematic way of developing J2EE Applications. The components employed in one application or *service* can be rearranged to define another service or be redeployed to a different environment (from a servlet to a console based application, or even an EJB for example). The MVC Framework is designed to solve a number of recurring

ORCL00010753

development problems, chief among them is the issue of the stateless environment of the web. It solves this problem by providing a process/page flow engine, allowing developers to organize their applications into any number of processes, and transition between these processes in a logical manner. Oracle9*i*AS MVC Framework for J2EE is a developer utility available from the Oracle Technology Network.

- *Oracle9iAS TopLink* -  Oracle9*i*AS TopLink provides an advanced persistence architecture for rapid development and deployment of enterprise Java applications that access relational databases. The TopLink Mapping Workbench provides a visual environment for developers to manage the relationships between relational database schemas and Java Objects or Entity Beans. TopLink provides flexibility in the choice of database, application architecture (Java Objects, Entity Beans (CMP and BMP), Servlets, JSP, Session Beans, and Message Driven Beans), application server, as well as object model and database schema design.

  TopLink is standards-based and can be used with any J2EE compliant application server and JDBC-compliant database. Oracle9*i*AS TopLink is included with Oracle9*i* Application Server or can be purchased separately.

In addition to these features that make Oracle9*i*AS a highly productive platform to develop J2EE 1.3 Applications, Oracle is committed to providing users the fastest, most scalable, reliable and highly available platform to deploy J2EE Applications. The many new facilities in Oracle9*i*AS Release 2 for RAS are discussed later in this document.

Developing Web Services

Web services are a set of emerging standards that enable Internet Applications to be developed and deployed in a service-oriented architecture and to communicate with each other in standard ways. More specifically, Web Services are a set of Internet standard messaging protocols, programming standards, and network registration and discovery facilities that expose business Applications to other Services over the Internet. A Web Service can (i) Expose and describe itself – A Web Service defines its functionality and attributes so that other applications can understand it, and makes this functionality easily available to other applications; (ii) Allow other services to locate it on the Web – A Web Service can be registered in an electronic registry so that other applications can easily locate it; (iii) Be invoked – Once a Web Service has been located and examined, the remote application can invoke its service using an Internet standard protocol; and (iv) Returns a response – When the service is invoked, the results are passed back to the requesting application again over the same Internet standard protocol.

Oracle classifies Web Services into two categories: simple and complex.

- *Simple Web Services* - Simple Web Services provide basic "request/response " functionality, and are not transactional in nature or provide sophisticated security. Simple Web Services are defined to comprise three primary standards - (i) WSDL - an XML schema used to describe the interfaces for a Web Service; (ii) UDDI - a registry used to categorize, publish, and search for Web Services; and finally (iii) SOAP - a network protocol used to invoke a Web Service.

- *Complex Web Services* - Complex Web Services are services that will transform the way industries conduct business. Such services, largely not addressed to date by most e-business software vendors today, will provide the framework for collaboration among trading partners over the Internet. They can be characterized as multi-party, long-running transactions (or business "conversations ") that involve sophisticated security, such as non-repudiation and digital signatures, as well as business-to-business collaboration and business process management. Complex Web Services are defined to comprise primarily ebXML and RosettaNet standards.

ORCL00010754

Oracle's Web Services Strategy

Oracle's Web Services Product Strategy is predicated on four specific objectives:

- *Web Services Standards Compliance* - First, Oracle is completely committed to supporting open J2EE Standards for application development. (i) Oracle9*i*AS enables developers to develop and deploy Web Services in Java/J2EE. This allows Web Services to be deployed on any underlying OS and hardware platform. (ii) Oracle9*i*AS provides complete support for simple web services standards - SOAP, WSDL and UDDI. Oracle9*i*AS' SOAP implementation is compliant with the Apache SOAP 2.2 implementation, the WSDL binding is compliant with the J2EE JAX-B binding, and Oracle's UDDI Registry is complaint with UDDI v 1.0 and implements several parts of UDDI v 2.0 (which is still being finalized). (iii) Oracle9*i*AS also supports Complex Web Services standards - ebXML and RosettaNet 1.1, 2.0. Oracle is implementing key parts of ebXML standards as they get finalized - JAX-B, JAX-M, JAX-P, JAX-RPC, and JAX-R. Oracle9*i*AS also provides complete support for RosettaNet 1.1 and 2.0 including all the RosettaNet 2.0 collaborations/PIPs.[4]
- *Productive Application Development* - Second, Oracle's wants to provide application developers seeking to either develop new applications as web services or to externalize existing applications as web services with the easiest to use and most productive environment to develop Web Services. We have done so by unifying the J2EE and Web Services' Programming Models in Oracle9*i*AS Release 2 in line with the current specifications for Web Services in J2EE 1.4.
- *Reliable Application Deployment* - Since Web Services will be used to develop and integrate mission-critical applications, we have made a number of design optimizations to ensure that Oracle9*i*AS provides a mission-critical infrastructure to deploy Web Services. The number of new facilities in Oracle9*i*AS Release 2 for RAS is discussed later in this document.
- *Interoperability with Other Web Services Environments - .NET Interoperability* - Finally, since Web Services are primarily designed to provide interoperability, Oracle ensures that Web Services developed on Oracle9*i*AS can both invoke and be invoked from Web Services developed to Microsoft's .NET framework. Similarly Oracle's UDDI Repository can be browsed from Microsoft and IBM's UDDI Browsers.

Now let us consider each of these capabilities in greater detail.

Oracle9iAS' Web Services Infrastructure

There are four key pieces to Oracle9*i*AS' Web Services' infrastructure:

- *SOAP Message Processor* - Oracle9*i*AS provides a SOAP Message Processor that is derived from the Apache 2.2 SOAP Provider with a number of enhancements. It runs as a servlet on Oracle9*i*AS' J2EE Container and provides the following facilities: (i) SOAP Protocol Handling - It provides a complete implementation of the interoperable SOAP specification including support for SOAP requests with Attachments (XML payloads); support for Cookies and Sessions which are useful to pass state information for stateful Web Services' request/response; and support for SOAP Message delivery over a variety of transports including HTTP and SMTP. Oracle9*i*AS' SOAP Processor integrates the Oracle XML Parser with and provides a number of performance optimizations to process both RPC-style and Messaging requests. Before invoking a Web Service, the SOAP Processor authenticates the user using the standard JAAS-based User Manager. It can also work with Oracle's Single Sign-On Server and third-party authentication services to provide single-sign on for Web Services.

[4] This feature will be available in a product update shortly after the initial delivery of Release 2.

ORCL00010755

- *WSDL Stubs and Skeleton Generation Facilities* - Oracle9*i*AS provides automatic generation of WSDL 1.0 compliant Stubs and skeletons. There are three important elements to WSDL support - (i) Auto-Generation of WSDL and Client Stubs - With Oracle9*i*AS, an application developer does not necessarily need to statically generate the WSDL interfaces for a Web Service and pre-compile the client stubs into a client application. Rather, the Oracle9*i*AS Web Services' runtime automatically generates WSDL XML and clients stubs on demand as it needs them - this makes application and system maintenance significantly simpler than competitor products. (ii) WSDL Generation Tools - Optionally, Oracle9*i*AS also provides a set of tools to statically generate WSDL XML and client stubs given a Java Class or J2EE Application. These tools have also been integrated seamlessly with Oracle's JDeveloper development tool and other IDEs. (iii) Registration with UDDI Registry - When a Web Service is deployed on Oracle9*i*AS, a user can use Oracle Enterprise Manager to register the specific Web Service and publish its WSDL XML to the UDDI registry.
- *UDDI Registry* - Oracle9*i*AS provides a UDDI v 1.0 Compliant Registry to publish and discover Web Services. It has several important elements - (i) Complete UDDI v 1.0/v 2.0 Implementation - Oracle9*i*AS provides a completely UDDI v 1.0 compliant registry and also implement key aspects of the UDDI v 2.0 specification. Oracle's UDDI Registry can be either used as a private UDDI registry within an organization, or as a public/hosted UDDI node. An important element of standards compliance is that Oracle's UDDI registry can synchronize with public UDDI Operator Sites or nodes using the standard UDDI Publication API. (ii) Compliant Programmer API - Oracle9*i*AS' UDDI registry also completely implements the UDDI Programmer API specification which allow programmatic access to UDDI registries in two logical parts - the Inquiry API and the Publishers' API. (iii) Database-backed, Fast, Highly Reliable, Available - As Web Services' proliferate, UDDI registries need to be fast, highly scalable, and highly available. Oracle9*i*AS' uses an Oracle Database to store UDDI information persistently and reliably.
- *Web Services Runtime Environment* - Oracle9*i*AS provides a unified runtime environment for J2EE Applications and Web Services. This provides Web Services with a number of services including transaction management, messaging, naming, efficient request brokering, and lifecycle management. Web Services can be implemented in a number of different ways - Java Classes, Enterprise JavaBeans, Message Driven Beans, or PL/SQL and Java Stored Procedures There are two ways in which Web Services can be invoked on Oracle9*i*AS - RPC-style and Messaging. (i) For RPC-style requests Oracle supports both a highly efficient static binding (similar to a CORBA TIE implementation) and a Dynamic Binding (aligned with the proposed JAX-RPC standard). (ii) For Messaging Style Invocation, Oracle allows a JMS listener including Oracle AQ, to receive and process a SOAP message payload. A Message Driven Bean can in turn be invoked from the JMS listener. Oracle also supports guaranteed, once-only, in-order delivery of SOAP messages over HTTP and SMTP transports.

Now that we have explained the Web Services facilities in Oracle9*i*AS, we will discuss how they can be used to develop, publish, and consume Web Services.

Developing Web Services with Oracle9iAS

Oracle9*i*AS Web Services provides complete infrastructure for developing, deploying and managing Web Services. Oracle9*i* JDeveloper facilitates efficient development and deployment of Web service onto Oracle9*i*AS and its J2EE Container provides the runtime platform for Web Services. Oracle Enterprise Manager (OEM) provides a Web Services Management console. The UDDI registry provides the facility to publish, query and search a Web Service.

ORCL00010756

- *Developing Web Services* - Oracle9*i*AS provides the ability to create Web Services out of (i) Stateless Java Classes; (ii) Stateful Java Classes; (iii) Stateless Session Enterprise Java Beans (EJBs); (iv) Message Driven Beans (MDBs); (v) PL/SQL Stored Procedures. The underlying SOAP Protocol handler supports both cookies and sessions for stateful Web Services. Oracle9*i*AS provides an Assembly Tool to assemble the above-mentioned artifacts into a standard J2EE .WAR archive for deployment.
- *Generating Web Services Stubs* - When the application is deployed, Oracle9*i*AS can auto-generate the WSDL and Client Stubs necessary on-demand i.e. when the Web Service is invoked. This eliminates the need for users to manage many different clients. Users can also use the command line tools provided to statically generate the WSDL for the application and the client stubs. Further, users can also begin a WSDL definition and use the command line tools to generate the necessary stubs and skeletons that can be extended to implement the Web Service.
- *Publishing Web Services* - Once the WSDL is generated, the user can use Oracle Enterprise Manager to step through a set of HTML wizards in order to register the Web Service with Oracle's UDDI Registry. Users can create new categories in the UDDI registry, browse existing categories, discover existing Web Services, and register new services. Oracle also provides features to import Web Services published in external UDDI registries.
- *Consuming Web Services* - Oracle9*i*AS Web Services can be consumed either statically or dynamically and can be invoked from existing J2EE Applications - JSP, Servlets, or EJBs via the SOAP Protocol. For static invocation Oracle9*i*AS provides highly optimized Web Service's proxy client that can be downloaded and embedded into a Client's Java applications. Further, J2EE Application Developers see external Web Services as J2EE artifacts i.e. as Stateless Session Enterprise JavaBeans further simplifying the programming model.

In addition to these Web Services facilities, Oracle9*i*AS provides a number of other services - to aggregate Web Services into an Enterprise Portal; to enable access to Web Services from any fixed or wireless device; to integrate Web Services with other Web Services, Oracle or non-Oracle databases, legacy systems and Enterprise Applications; to Personalize Web Services for individual users; and to deploy, manage, and secure web services. Further, Oracle9*i* JDeveloper provides a complete J2EE and XML development environment to build, test, debug and deploy Web Services for Oracle9*i*AS.

More detailed information on Oracle's Web Services Strategy and Oracle9*i*AS' capabilities is available in a separate technical whitepaper.

Web Enabling Existing Applications

Oracle9*i*AS can also be used to web-enable existing Applications that have been developed to client-server architectures and to migrate them to a three-tier architecture. There are three kinds of applications that can be web enabled - Oracle's Forms Applications, Active Server Page Applications, and other 4GL-style Applications. Let us examine at each of these aspects in detail.

- *Oracle9iAS Forms Services* - Applications can be built for Oracle9*i*AS Forms Services using a set of integrated builders from Oracle9*i* Developer Suite that allow application developers to easily and quickly construct sophisticated database forms and business logic with minimal effort. Forms built in this environment are delivered through any Internet browser as a high performance, rich Java client. On the application server tier, Oracle9*i*AS Forms Services consists of a listener Servlet and a runtime engine. The runtime engine accesses data in an Oracle database or can connect and use non-Oracle data sources using the Oracle Gateways or ODBC drivers. The client tier consists of a generic, lightweight Java client Applet. The Forms runtime engine runs

ORCL00010757

the business logic defined in the Form application and provides the connection to the database. The generic Forms Java Applet provides the user interface for running the Forms application.

- *Active Server Page Applications (ASPs)* - As described earlier when discussing the Proxy Plug-in for Microsoft IIS, Oracle9*i*AS allows Active Server Pages to co-exist with J2EE Applications. Further, Oracle also offers a source-to-source translator to convert ASPs to JSPs.
- *Other Legacy Applications* - Finally, Oracle9*i*AS also allows other 4GL and legacy applications that have been designed as mid-tier executables to be accessed from browser clients through the Apache mod_FastCGI.

As described in this section, Oracle9*i*AS Release 2 provides you with facilities to design and develop dynamic web sites and transactional e-business applications; to make these applications accessible as Web Services; and to web-enable existing Applications.

CREATE ENTERPRISE PORTALS

As the number of Web Sites and Web Services proliferate within your organization, Oracle9*i*AS enables you to create Enterprise Portals that offer users (customers, employees, suppliers) with a secure, personalized, easy-to-use place to find the essential information and self-service applications they need to operate more effectively. Oracle9*i*AS Portal provides an out-of-the-box portal solution that does not require endless programming and maintenance.

Oracle9*i*AS Enterprise Portal Strategy

Oracle9*i*AS provides the industry's leading enterprise portal solution based on the following strategic objectives:

- *Easy-to-use, Self-Service Portal Creation* - Organizations need a way to enable their employees to easily publish and manage information on the Web. Oracle9*i*AS Portal is a browser-based software environment for building and deploying enterprise portals. Oracle9*i*AS Portal's self-service publishing features eliminates developer and administrator bottlenecks by allowing end users to post and share any kind of document or Web content with other users.
- *Open Interface to Aggregate Information* - Oracle9*i*AS Portal also provides a secure, manageable framework for delivering access to distributed software applications and information resources via an extensible interface called the "portlet" interface.
- *One Place to Find Information* - From their Enterprise Portal, users can execute enterprise-wide information searches to find information that may sit in multiple applications on many different internal systems.
- *One Place to Authenticate Users* - When accessing these different internal systems, users can go to a single place to authenticate themselves. Once they have Single Sign-On, they do not need to remember multiple user names and passwords to access different systems.
- *One Place to Personalize Information* - Finally, the easy-to-use portal interface offers an organized, personalized view of the business information, Web content, and applications required by each user. This information can be personalized based on the individual user's identity or the group that the user belongs to.

Let us now consider the Enterprise Portal capabilities in Oracle9*i*AS Release 2 in greater detail.

Oracle9*i*AS Portal Capabilities

In Release 2, Oracle9*i*AS adds a number of new Enterprise Portal capabilities.

Self-service Content Publishing

ORCL00010758

Companies need a way to enable their employees to easily publish and manage content on the Web. They would like to centrally administer their portal data, produce a consistent look-and-feel to the site, and provide internal access to content, transaction, and business intelligence applications. Oracle9*i*AS Portal supports all of these requirements, using single sign-on, personalization, and portlet extensibility technologies to provide a simple, integrated view of external content, corporate information, and applications.

- *HTML Design Tools and Support for WebDAV Facilities* - In Oracle9*i*AS Release 2, support of the WebDav publishing standard gives users additional publishing flexibility and desktop integration. Improved enterprise searching capabilities and controls make content from external repositories more accessible.
- *Content Publishing* - The integrated page building allows content publishing users to directly specify and manage page content in a new editing dashboard that combines content location, attribution, security, and display characteristics. New customization and bulk action capabilities make it very easy for users to quickly build and personalize corporate portals. Content publishers can also take advantage of content approval and routing capabilities to specify one or more individuals for review/approval. A notification portlet displays items awaiting approval and a status portlet provides visibility into the approval process.

Wizard based Development

Oracle9*i*AS Portal makes it very easy to create and administer portal pages and portlets. A page template model to streamline rapid development and allow administrators to enforce consistency across pages compliments browser-based wizard driven development.

-  *Simplified GUI Development* - new usability related features allow portal end users to more easily interact with portal content and customize pages to suit their individual preferences.
-  *Integration of Servlets/JSPs* - In Oracle9*i*AS Release 2, the "Copy as a JSP feature" allows you to generate a page definition that can be edited/enhanced in your favorite JSP editing tool - when complete you simply upload the completed page.
-  *Enhanced Enterprise Search* - In Release 2, the Portal has much better enhanced enterprise search facilities through integration with Oracle Ultrasearch. Ultrasearch is an enterprise Intranet search facility that consists of - (i) A highly scalable, schedulable, parallelizable web crawler that can be targeted to search various Intranet and Internet sites; (ii) An efficient indexing facility that indexes the information generated by the crawler and stores the indices in a database; (iii) The search engine which looks up the generated and stored indexes to find appropriate information; and (iv) A thesaurus that can be used provide highly context-specific information about specific key words - for instance, the word "oracle" on Oracle Corporation's own Intranet means something very different from the same word on a Greek library's Intranet.

Extensible Portlet Framework

In addition to its core portal development and management features, Oracle9*i*AS Portal provides an extensible framework for integrating applications and business information. The framework is based on Oracle's unique portlet technology. Portlets are reusable interface components that provide access to Web-based resources. Any Web page, application, business intelligence report, syndicated content feed, hosted software service or other resource can be accessed through a portlet, allowing it to be personalized and managed as a service of Oracle9*i*AS Portal. The portal framework provides additional services including single sign-on, content classification, enterprise search, directory integration and security.

-  *New J2EE-based Portlet Interface* - Developers use the Portal Developer Kit (PDK) to create portlets using Java, and XML technologies. New in Release 2, the Java Portal Developer Kit (PDK-Java) provides a portal-independent user interface for creating and managing external

ORCL00010759

portlet providers. This gives developers a declarative way to create secure, custom portlets on existing J2EE applications.

-  *Pre-Built Portlets* - In addition to using the PDK to build your own custom portlets, Oracle9*i*AS Release 2 includes many pre-built portlets including People Search, Approval and Notification portlets, Folder portlets and, a number of extensive Search portlets You can also easily access your Business Intelligence data through the built in portlet providers: Oracle9*i*AS Discoverer - query results and workbook listings, Oracle9*i*AS Reports, as well and Oracle9*i* File System.
- *ISV Portlets* - You can also plug-in portlets from over a 100 partners including Webex, InfoSpace, Factiva, MapInfo and Inktomi.

Performance, Scalability, Availability, Mobility

To improve the performance, scalability and reliability of the portal, the deployment architecture has been completely integrated with Oracle9*i*AS infrastructure.

- *Page Assembly Servlets* - The Web Cache is has been fully integrated with the page servlet engine for portal page, page metadata and portlet content and with the existing file-system based cache.
-  *Web Cache Integration* - Portal page assembly logic utilizes Web Cache's invalidation-based caching features to pre-populate the cache, allowing many page requests to be served entirely from the cache and a significant reduction in calls to the portal database. To further enhance performance, Oracle9*i*AS Web Cache can be deployed in front of external applications accessed by the portal from a URL and called via an HTTP request. New functionality in Oracle9*i*AS Release 2 allows application providers defined within database procedures to be called in a similar fashion.

In addition, through integration with Oracle9*i*AS Wireless, any portlet can be wireless enabled.

Secure Hosted Portals

Customers who need to host multiple portals will be able to manage their environment more effectively by utilizing Oracle9*i* Virtual Private Database (VPD) features. VPD enables partitioned, deep data protection by attaching access control directly to the data. Using VPD, enterprises or ASPs hosting multiple companies' information can securely isolate the data at the row level, across all applications, within a single portal instance.

More detailed information on Oracle's Enterprise Portal Strategy and Oracle9*i*AS' capabilities is available in a separate technical whitepaper.

WIRELESS ENABLE INTERNET APPLICATIONS AND PORTALS

Wireless Internet technologies are growing rapidly driven by the combination of broad penetration of Wireless devices and the growing standardization of Internet Access from such devices. According to market research performed by the Meta group, "by 2003, over 50% of Internet access will be by non-PCs". Similarly, Gartner predicts "by the year 2005, one billion mobile devices will be used worldwide." Oracle believes that users will want to access all applications, from any wireless device, at any time, across any wireless network, and from anywhere in the world.

Oracle9*i*AS Wireless Strategy

Oracle9*i*AS provides a comprehensive wireless solution based on the following strategic elements:

ORCL00010760

- *Unified Development Model* - First, while Wireless Internet standards are emerging, there is still tremendous heterogeneity in the standards supported by wireless devices. Each device speaks a different wireless protocol and supports a variety of different Markup Languages. These different standards and the proliferation of different types of devices preclude an application developer from writing every application to individually support every single device available. Oracle9*i*AS offers a J2EE and XML standards compliant programming model that radically simplifies how developers create, deploy and manage Wireless accessible applications and portals.
- *Any Network, Any Device, Any Gateway* - Second, recognizing the heterogeneity of devices, network protocols, and other facilities required for Wireless Applications, Oracle9*i*AS certifies a very large number of devices, networks and gateways ensuring that any Application can be accessed across any network, any device, and any wireless protocol gateway.
- *Exploiting Wireless Internet* - Finally, wireless devices typically have small screens and limited data entry capability - rather than simply subset HTML information for small screens, Oracle9*i*AS allows users to exploit specific capabilities such as voice access, push, location-based services, wireless broadband, wireless PIM to ensure that wireless users have a rich user experience.

Let us now consider the product's capabilities in greater detail.

Oracle9*i*AS Wireless Capabilities

Oracle9*i*AS Wireless has several important facilities:

Unified J2EE, XML and HTTP Programming Model

Oracle9*i*AS Wireless provides application developers independence from the underlying wireless infrastructure - underlying networks, protocols, devices, gateways and other wireless complexities. A developer creates an application that outputs XML, and points Oracle9*i*AS Wireless to the application with a URL. When a request is received from a user to access this application, Oracle9*i*AS recognizes the device and network over which the request was received, and the URL that the user wants to access. It routes the request to the application and receives the results/content in an XML format (this process is called "content adaptation"). Oracle9*i*AS isolates content acquisition from content delivery through an intermediary format layer, Oracle9*i*AS Wireless XML, between the source format and the target format. It is a set of DTDs (Document Type Definitions) and XML document conventions used to define content and internal objects. Once the content has been received in this form, using XSL style sheets, Oracle9*i*AS Wireless transforms these XML documents to the mark up language and protocol format appropriate for the specific mobile device (this process is called "device or network adaptation"). There are several benefits to this model:

- *Wireless Infrastructure Independence* - First, and most importantly, is that Oracle allows users to develop applications with a single unified programming model based on J2EE, HTTP and XML standards without rewriting or customizing the application each time it needs to be accessed from a new device or new network.
- *Scalability* - Second, by keeping a single representation of the content in an XML format in memory, Oracle9*i*AS scales very efficiently as requests for the same content increases from multiple users with many different devices.  In Release 2, the XML Application Model (schema) has been significantly enhanced to capture richer information about various types of content and to address delivery requirements for high quality Voice, Color, Sound, Multimedia (2.5G-3G) content and persistent and multi-channel sessions.
- *MultiChannel Unification* - Finally, by using a unified Programming Model, Oracle9*i*AS also makes any application created to be easily accessed simultaneously from multiple channels -

ORCL00010761

e-mail, browser, push, voice, and any device's microbrowser - with consistent state across all these different access channels.

Oracle9*i*AS Wireless uses the Oracle9*i*AS Infrastructure repository to store persistent application objects. Runtime API's provide the functionality to manipulate the platform's persistent data objects stored in the repository. The Oracle9*i*AS Wireless APIs can customize the runtime behavior of the server, for example, providing a different authentication scheme or providing a customized device identification mechanism. Oracle9*i*AS Wireless also provides an extension framework, which allows for plug-in of additional logic, such as logging or system monitoring, that does not change the runtime behavior.

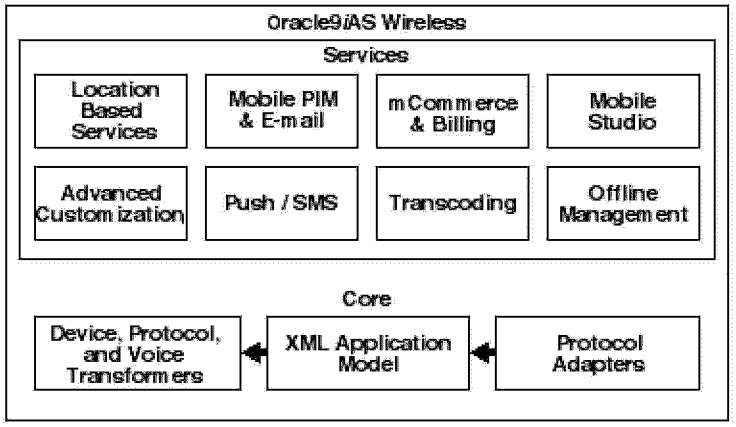


Figure 7 - Oracle9*i*AS Wireless

Wireless Product Capabilities

In Oracle9*i*AS Release 2, Oracle has significantly enhanced its wireless product capabilities including:

- *Productive Wireless Development Tools* - To make wireless developers highly productive, Oracle9*i*AS provides advanced Wireless Development Tools including - (i) The Service Designer is used by developers to design new wireless services, link services together, and to manage wireless applications; (ii) The Content Manager is used to manage the end user's view of the wireless services; (iii) and the User Manager controls the users, groups and the access control privileges they have to various system resources.
- *Protocol/Content Adapters* - Protocol or Content Adapters fetch the Application's content from various content sources in XML format. NEW In Release 2, Oracle9*i*AS Wireless includes a much broader set of Protocol Adapters - the two primary adapters being HTTP and OC4J (J2EE). (i) The HTTP Adapter provides the ability to act as a wireless proxy. Oracle9*i*AS Wireless will mobile enable any URL. It supports the HTTP and HTTPS protocol and is ideally suited for applications built completely in JSPs or Servlets. (ii) The OC4J (J2EE) adapter fetches content from the Java Servlet's/JSP's based applications running within the same J2EE container as the Oracle9*i*AS Wireless. (iii) Other Protocol Adapters include SQLAdapter for enabling database queries and PL/SQL; Web Integration Adapter allows existing HTML Web applications to be

ORCL00010762

modeled as XML documents; CustomAdapters, written in Java, can be created to integrate any application.

- *Device and Network Management* - Device/Network Adaptation automatically transforms and optimizes the application content to any device and network. NEW In Release 2, Oracle supports and certifies a much broader set of devices including 2-way pagers for asynchronous services (SMTP/SMS), all WAP devices, Voice access through regular phone lines, PDA devices. Device/Network Adaptation is enabled with Oracle9*i*AS Wireless Transformers, which utilize standard XSL Style Sheets. The Style Sheets are continually maintained and updated by Oracle.
- *Highly Scalable Wireless Push* - Oracle9*i*AS does not allow you to simply download information to wireless devices but can also allow applications to push information asynchronously to these devices over SMS or SMTP. NEW In Release 2, Oracle9*i*AS Wireless' asynchronous server is much more scalable through integration with Oracle's Advanced Queuing facility with alerts that can be set based on conditions and time.
- *Comprehensive Voice Support and Location-based Services* - NEW In Release 2, Oracle9*i*AS Wireless provides complete support for voice-based applications - wireless users authenticate themselves using a voice print; they can personalize the wireless services they want to access by speaking; and information can be delivered to wireless devices in voice form through integration with a number of leading VoiceXML gateways. Similarly, location-based services provide a framework for creating location-aware applications. NEW In Release 2 this has been enhanced with geo-coding, automatic and manual mobile positioning, routing and navigation and mapping.
- *Wireless Broadband Network and Multimode Device Support* - NEW In Release 2, Oracle provides the industry's most advanced support for wireless broadband networks (2G/2.5G/3G) - (i) Multimode Device certification allowing users to use popular multimode devices to access Internet Applications during the transition to wireless broadband networks; and (ii) Support for session management facilities that allow "continuously connected" wireless broadband devices to be supported in a highly scalable manner.
- *Advanced Wireless Customization and Services* - Finally, the Oracle9*i*AS Wireless Advanced Customization has added Presets, flexible Data Feed capabilities, and Multiple Profiles. User management is based on OID/LDAP and has support for Single-Sign-On.

Wireless Applications

NEW In Release 2, Oracle9*i*AS Wireless not only provides users with the ability to develop Wireless Internet Applications but also offers a number of facilities that can be used to very quickly Wireless enable existing applications. These include:

- *Mobile PIM (Personal Information Management)* - The Mobile PIM facilities offer (i) Mobile Access to Email - send and receive of e-mail to mobile devices including voice access to e-mail; (ii) Mobile LDAP Access - people finder access from mobile devices; (iii) Wireless Instant Messaging - send and receive instant messages to wireless devices; (iii) Mobile Calendaring and Scheduling - schedule meetings from mobile devices; and (v) the ability to connect to any IMAP/LDAP server.
- *Mobile Commerce* - For mCommerce, Oracle9*i*AS Wireless provides facilities for secure mobile transactions; secure storage of user identity in an mWallet (mobile Wallet); and support for the most common payment mechanisms.

Wireless Deployment

Oracle also provides three different forms of Wireless Application Deployment - (i) Hosted Deployment through OracleMobile; (ii) Customer Premise Deployment where customers deploy their own wireless infrastructure; and

ORCL00010763

(iii) Wireless ASP allowing service providers to use Oracle9*i*AS as their wireless infrastructure. For users choosing to host their applications with OracleMobile, the Mobile Studio offers Developers a portal environment where developer's can test and deploy wireless applications easily, and immediately. This is a cost efficient and scalable method to have developers deploy their wireless applications.

To summarize, Oracle9*i*AS Wireless is designed to use industry standards - J2EE, HTTP, and XML - to allow applications to be developed once, and to be accessed from any device, across any network, and any gateway. It also provides the performance, scalability, and availability required to support millions of wireless users accessing applications concurrently. More detailed information on Oracle's Wireless Strategy and Oracle9*i*AS' capabilities is available in a separate technical whitepaper.

INTEGRATE YOUR ENTERPRISE

As organizations move to capitalize on Internet enabled applications such as dynamic supply chains, customer relationship management, and emerging web services opportunities, they must first integrate enterprise applications and business processes in order to participate in the new connected environment. Achieving this integration can be very challenging. There are often many different applications using different technologies and architectures. Many of these systems are constantly evolving, thus complicating implementation and maintenance of the integration solution. Further, managing this complexity is difficult and can be very expensive. Worst of all the cost and schedules are often difficult if not impossible to predict without the appropriate process and tools.

Oracle9*i*AS Integration Strategy

Oracle9*i*AS addresses these challenges and provides the necessary infrastructure that enables you to effectively integrate a myriad of disparate technologies into a streamlined, efficient e-business operation. It provides a powerful business process-driven solution that enables organizations to implement a systematic and scalable infrastructure for integrating virtually any packaged or legacy applications, application components, trading partners and people. For Oracle E-Business Suite and Exchange customers, it provides the ability to quickly and easily link to 3rd party packaged applications such as SAP and PeopleSoft.

Oracle9*i*AS Integration provides a comprehensive enterprise-wide integration solution based on the following major strategic elements:

- *Single Standards-based Integration Architecture* - Oracle believes that organizations should not integrate systems in a piece-meal manner but should adopt a systematic architectural approach using a single architecture and a single platform to integrate legacy systems, packaged applications (A2A), business partners (B2B), and any systems that are wrapped as Web Services (Web Services). Further, in doing so, Oracle believes that companies need to shift away from using proprietary integration middleware to using an integration solution based on open standards such as J2EE, JMS, JCA, XML Messaging, and Web Services.
- *Single Comprehensive Integration Infrastructure* - To facilitate integration between legacy systems, packaged applications, business partners, and Web Services, Oracle provides a single comprehensive integration infrastructure consisting of (i) Modeling Tools that are used to model, generate, and monitor an end-to-end business process; (ii) Design Time and Runtime Repositories that capture metadata and state of the integration respectively; (iii) A Runtime or Integration Manager which orchestrates the integration and invokes various Services for Transformation, Translation, Events, Validation, Security and others; (iv) and a Broad Set of Connectors.

ORCL00010764

- *Single Productive Tool Set to Design Integration* - Further, Oracle's solution is designed to limit the amount of custom software code that is written to integrate different systems making the environment both maintainable and manageable. It does so by providing a comprehensive set of HTML Tools - (i) To Model or Design an end-to-end Integration; (ii) To Manage the systems infrastructure for the integration; (iii) To Monitor a Business Process that spans different systems; and (iv) To Analyze or capture Business Process Intelligence.
- *Single View of Business Process* - Oracle's integration solution allows customers to have a common view of the business process and to have a common view of information flowing between different systems. It allows users to go to a single console to monitor the current state of the business process (for instance, whether the purchase order been received in the SAP System); business events; data (for instance, whether the purchase order was for less than $1 Million), and other information without needing to query the many different systems that are being integrated with each other.
- *Maintainable and Manageable* - Systems being integrated using Oracle's solution can be gracefully evolved without significant impact on other systems. Integrations created with one set of systems and with one version of applications can be maintained when either the process itself changes or one or more of the systems being integrated changes.

Let us now consider the Enterprise Integration capabilities in Oracle9*i*AS Release 2 in greater detail.

Oracle9*i*AS Integration Capabilities

Oracle9*i*AS InterConnect (formerly Oracle Applications InterConnect or OAI) provides a comprehensive application integration framework that is designed to elevate the integration problem from a technical coding exercise to a functional modeling exercise, thereby reducing (eliminating, in the best case) the programming effort normally associated with integration. Its integration methodology promotes reuse and reduces the complexity and management issues that arise over the software lifecycle. The product's capabilities consists of the following key components:

Integration Design and Management Tools

Oracle9*i*AS Integration provides a single integrated set of browser-based HTML Tools that are aimed at enabling business users and developers to both very quickly and efficiently set up integrations, modify them, and monitor them. To make users productive and to improve time-to-market, the system is designed to limit the amount of coding required to set up and modify integrations. These tools include:

- *Integration Modeler* - Oracle9*i*AS offers a set of HTML-based tools to model all aspects of an enterprise-wide integration and business process including modeling the business process, mapping data transformations and translations, integrating specific end-points into the system, and creating and modifying trading partners and their agreements etc. All of these can be modeled with a set of HTML wizards and the information generated into a metadata repository.
- *Systems Monitoring and Management* - All the aspects of the integration infrastructure - the tools, repositories, runtimes, and connectors - are deployed on Oracle9*i*AS and are monitored and managed as a feature of the Application Server using Oracle Enterprise Manager.
- *Business Process Monitor* - To monitor the state of a business process or a specific integration, to determine whether a specific message (for instance a purchase order) has been delivered from one system to another, Oracle9*i*AS provides a comprehensive business process monitor that provides a single place for users to monitor, analyze, and drill down on the state of the business process.
- *Business Process Intelligence* - Finally, for administrators or business users to understand the overall flows of business processes within their organization (for instance to understand which systems

ORCL00010765

exchange the most information or which trading partners one does business with), Oracle9*i*AS also offers a set of business intelligence facilities to perform off-line analysis on the integration history.

Design Time and Runtime Repositories

As described earlier, Oracle9*i*AS Integration is a metadata-driven integration solution. To streamline the process of designing and maintaining enterprise-wide integrations, all of the behavior of the system is managed by configuration rather than through coding. The metadata and state information about executing integrations are captured in two repositories:

- *Design Time Repository* - When a user models an integration in the Business Process Modeler, the information that describes the integration is captured as metadata in the design-time repository. The design-time repository maintains the definitions of data types, business processes, transformations, and other information that specifies the behavior of the system and the services that need to be sequenced to carry out a business process.
- *Runtime Repository* - Once an integration or business process has been set up and deployed, the run-time repository maintains all instance data used by, or created by, the run-time execution of the system. This includes performance optimized design metadata, trading partner profiles and agreements, and system logs of various services that are being invoked. It also includes a comprehensive view of the state of the business process providing a single point where the Business Process Monitoring Tools can query to find this information.

Runtime and Integration Services

The run-time orchestrator reads the metadata information that describes the integration and executes the business process or integration.

- *Orchestrator* - The Orchestrator or Integration Manager is itself a J2EE standard compliant runtime that executes all integration processes and invokes a set of services required as data moves through the system. It is designed for performance, transaction integrity, scalability and high availability. The run-time also includes all of the trading partner management services used to provision and enforce the integrity of trading partner interactions.
- *Runtime Services* - As it executes integration, the orchestrator also invokes a set of services that include business process management, messaging, events, transformation, translation, validation, security, logging, monitoring, and transaction management.

Adapters or Connectors

Adapters or Connectors provide connectivity to external applications and technologies. There are three kinds of adapters that Oracle provides:

- *Custom Adapters or Adapter SDK* - When integrating a legacy system or business process, users may need to integrate a propriety or custom-built application into Oracle9*i*AS' Integration infrastructure. This can be achieved either by wrapping the systems interfaces via a standard J2EE Connector Architecture (JCA) API or as a Web Service. Oracle9*i*AS provides a set of tools and an adapter SDK to make this process easy.
- *Pre-Packaged Adapters* - Oracle9*i*AS also offers a large number of pre-packaged adapters to the most commonly identified systems including - (I) Legacy systems such as mainframes; (ii) Packaged Applications such as SAP, Peoplesoft, and Oracle's Applications; (iii) Oracle and non-Oracle Databases; (iv) and a Variety of Messaging Systems and Transport Infrastructures. Adapters insulate the core run-time system from the unique technology issues associated with the various end-points.

ORCL00010766

- *B2B Messaging Facilities/Adapters* – Oracle9*i*AS also offers B2B Messaging Facilities which are in effect a specialized form of adapter that provides connectivity to trading partners or to connect supply chain systems using a broad range of standard transport technologies while enforcing the security and collaboration requirements of B2B standards specifically RosettaNet 1.1 and 2.0, ebXML, and EDI. It includes - (i) **A Comprehensive Trading Partner Management System**[5] that speeds provisioning and streamlines administration of trading partners for B2B collaboration. The trading partner system maintains profiles - overall contact details for trading partner entities; parties - specific organizations within a profile that will participate in a collaboration (division, department etc.); and agreements - the specific collaborations, roles, and communication options that dictate how two or more Parties will interact. Oracle9*i*AS supports a variety of industry standards for import and export of profiles and agreements including ebXML's CPP/CPA and TPAML formats. (ii) **B2B Messaging Facilities -** Oracle9*i*AS provides support for B2B protocol standards[5] including ebXML, RosettaNet, and EDI (X.12, EDIFACT). It includes an extensible, layered B2B protocol engine that manages the following functions - Transport - HTTP/S, SMTP, FTP/S, messaging systems; Packaging and Signing - SMIME 3.0, SOAP 1.1, Digital Signatures; Document Exchange - XML, RNIF 1.1/2.0, ebXML, EDI; and Collaboration - RosettaNet PIPs, and ebXML Collaborations. The transport, packaging *i* signing, and document exchange layers are extensible via an open, Java-based Plug-in API. Custom collaborations may be created through a visual modeling tool.

To summarize, Oracle9*i*AS includes integration facilities to integrate legacy systems, enterprise applications, Oracle and non-Oracle Databases, and B2B trading partners all based on open standards. It can either be used either to tie specific systems together or as an enterprise-wide integration infrastructure through which all systems and applications communicate with each other.

More detailed information on Oracle's Enterprise Integration Strategy and Oracle9*i*AS' capabilities is available in a separate technical whitepaper.

Communicating with Stakeholders

Oracle9*i*AS makes it easier for users to collaborate with others using Oracle Internet File System (iFS). IFS consolidates all forms of corporate data into a single, unified and manageable repository. It allows you to store all your data, including documents, presentations, graphic images, HTML and XML Web pages, e-mails, and video in the same file system that is an extension to the Oracle9*i* Database. Users can access this content through familiar interfaces and protocols including the Windows file system, Web browsers, FTP, and e-mail clients. Internet File System consists of the following major components:

- *Repository* - All files are centrally stored and managed as Large Objects in a repository, which is held within an Internet File System schema in the Oracle Database. The files are stored and retrieved using standard Oracle JDBC calls to read and write Large Objects in the database.
- *iFS Server* - The Internet File System server houses the processes that manage the repository. It contains the Protocol servers and the Java API. Protocol servers handle communication between the clients and the Internet File System repository. Internet File System supports numerous protocols including HTTP, FTP, SMB, IMAP4, SMTP. The Java API is a class library that the developers can use to customize and extend the functionality of Internet File System, including creating custom parsers and additional protocol servers.

[5] The Trading Partner Management features and B2B support will be available early in 2002, in a version subsequent to the initial release of Oracle9*i*AS Release 2.

ORCL00010767

EXTRACT AND ANALYZE BUSINESS INTELLIGENCE

Oracle9*i*AS provides you with the ability to capture, analyze and extract business intelligence about how users are accessing your applications or Web Sites. Traditional information sources such as Enterprise Resource Planning (ERP) systems and CRM (Customer Relationship Management) systems can be correlated with information that comes directly from the usage patterns of the organization's Web site. Oracle9*i*AS Business Intelligence provides a comprehensive solution allowing you to put the right information, into the right users hands, at the right time, helping them to make informed decisions.

Oracle9*i*AS Business Intelligence Strategy

Oracle9*i*AS provides a comprehensive business intelligence solution that consists of the following elements:

- *Web Site Analysis* - Oracle9*i*AS Clickstream Intelligence provides users with a consistent view of aggregate patterns of usage or activity on their Web Sites or Internet Applications.
- *Web Site Personalization* - Oracle9*i*AS Personalization enables users to track the activity of a specific user and to personalize information or applications for that user based on his or her activity pattern in the past, behavior of users similar to that user, or his or her current activity.
- *Web-based Business Intelligence* - Oracle9*i*AS Reports Services and Oracle9*i*AS Discoverer respectively provide web-based enterprise reporting and decision support facilities - reports and workbooks can be accessed over the Internet via an Enterprise Portal.

Let us now examine each of these pieces in greater detail.

Oracle9*i*AS Clickstream Intelligence 

Oracle9*i*AS Clickstream Intelligence is a Web Analytics Tool that enables users to capture and analyze **aggregate information** about Web Site usage - Web site Throughput/performance, Visitor Traffic Patterns such as where users come to the Web Site from and where they go to from the site, the Effectiveness of specific Web pages or types of content, and Customer loyalty such as what percentage of users are repeat visitors vs. new visitors. It has several important capabilities:

- *Robust Data Mart Model* - Oracle9*i*AS Clickstream Intelligence consists of the following major components - (i) Log Analyzers that read Web Server Logs, parses cookies and query string data to identify unique visitors and sessions, and captures other critical application data. (ii) Extract, Transform, Load (ETL) facilities which extract the parsed data and loads it in a highly scalable manner into a Data Mart. (iii) The Clickstream Data Mart which provides a robust, extensible data model that delivers significant out-of-the-box value while providing the extensibility to meet a broad range of business requirements. Facilities include the ability to transform server log data into business facts such as page impressions and sessions; and to correlate Web traffic with business transaction data such as orders and marketing campaigns. (iv) Discoverer-based Reports that are used to generate a set of pre-built reports for measuring Web traffic and improving Web site effectiveness. Users can also build their own custom reports.
- *Scalable Data Flow* - Oracle9*i*AS Clickstream Intelligence provides excellent performance and scalability by leveraging Oracle9*i*'s advanced data warehousing features to scalably manage the flow of large data volumes from the Application Servers to the data mart.
- *Easy to Understand Reports* - Finally, Clickstream Intelligence provides a standard set of reports for rapid deployment and supports user-defined reporting. Users can create their own custom reports

ORCL00010768

and the system is extensible to correlate web site traffic patterns with user-defined data sources such as call centers, ad servers, etc.

Oracle9*i*AS Personalization

In contrast with Oracle Clickstream Intelligence that examined aggregate activity, Oracle9*i*AS Personalization provides real-time, 1:1 recommendations based on the specific behavior of individual web site users. It uses data mining technology to sift through the large volumes of data gathered from customers' clicks, transactions, and demographics, and provides users with accurate, personalized product recommendations leading to greater customer satisfaction. Web Site developers can improve site navigation based on the interests and profiles of their users. Oracle9*i*AS Personalization can also be integrated with e-business applications, such as Web Stores, online catalogs, Application Hosting environments, and Call Centers. It has several important capabilities:

- *Data Mining Driven Recommendation Engine* - Oracle9*i*AS Personalization consists of the following components - (i) REPAPI - The Recommendation Engine API (REAPI) allows developers to instrument elements of their Web Site that they want to personalize using a standard JSP or servlet API. (ii) Recommendation Engine - When users "click" on these instrumented elements, information about the user's behavior is sent to the recommendation engine eliminating the need to sift through large volumes of "noisy" clickstream data and providing the data necessary for personalization in a fraction of a second. (iii) Scoring Facility - A history of the user's behavior on the site during the current session or during past sessions can be correlated with information about other users' activities on the site in a data warehouse. SQL queries are used to "score" this information generating a sorted ordered list that is then used by the recommendation engine to serve real-time recommendations to individual users. Scoring can be executed in real-time (online) or batch mode (offline).
- *Sophisticated Predictive Customer Modeling* - The value of using data mining for personalization is that the algorithms find hidden patterns and customer profiles to make very accurate recommendations with a high probability of satisfying users. Oracle9*i*AS Personalization has a number of sophisticated predictive modeling algorithms that can be used including Collaborative Filtering, Transactional Naïve Bayes, and Predictive Associations. These predictive models may be rebuilt on a periodic basis.
- *Easy to Use with Automatic Customer Profiling and Modeling* - Oracle9*i*AS Personalization minimizes the effort needed to create highly accurate personalized recommendations. It automates the process of building a real-time profile for each customer or anonymous visitor. Further, it automatically selects the best offer for each point of contact based on what it knows about a particular customer. As individuals accept or decline offers, Oracle9*i*AS Personalization adjusts and incorporates that information into future offers. It also reduces routine maintenance efforts by allowing Web administrators to build, tailor, manage, and deploy many recommendation engines enterprise-wide from a single administrative interface. Web administrators can also set up schedules for primary events — such as model building, model deployment, and reporting — to occur automatically.
- *Performance and Scalability* - Oracle9*i*AS Personalization allows users to create "recommendation engine farms" that are comprised of many recommendation engines serving customized recommendations to the Web site. This architecture is extremely scalable for high-traffic sites. Oracle9*i*AS Personalization and Oracle9*i* Database store the predictive models in memory to handle the high traffic and speed requirements associated with e-commerce sites.

ORCL00010769

Oracle9*i*AS Reports Services

Oracle9*i*AS Reports Services is an Enterprise Reporting Service used by information systems departments to produce high quality, production reports. These reports dynamically retrieve, format, and distribute database information reports in an unlimited number of formats to an unlimited number of recipients and are easily, dynamically generated via the Web with a standard Web browser. Oracle9*i*AS Reports can create reports using any data to produce output in any format and deliver boardroom quality reports anywhere.

- *Scalable 9iAS J2EE-based Architecture* - Oracle9*i*AS Reports consists of two pieces (i) Oracle9*i*AS Reports Developer, the integrated development environment that ships with Oracle9*i* Developer Suite and (ii) Oracle9*i*AS Reports Services, the runtime component that executes report definitions, which is part of Oracle9*i*AS. New in Release 2, Oracle9*i*AS Reports has fully embraced the J2EE standard enabling seamless integration with servlets and JSPs. (iii) Oracle9*i*AS Reports Services leverages the load balancing and distributed facilities of Oracle9*i*AS to distribute requests across multiple report generation engines on the same CPU and across multiple CPUs. Oracle9*i*AS Reports Services also caches generated output for reuse on similar requests. (iv) Access control privileges for individual users and groups seeking to access reports are determined by validating user credentials against Oracle9*i*AS Single Sign-On and Oracle Internet Directory's (OID) LDAP repository.
- *J2EE Standards-based Architecture* -  New in Release 2, is the ability for (i) Reports developer's to create and deploy JSP-based Web reports by utilizing JSP custom tags - you can create your JSP report using the new JSP layout wizard or manually by adding your own custom tags. When a report is saved as a JSP file, the data model is embedded using XML tags. The entire report can now be defined using XML tags and saved as an XML file. (ii) Java Beans in the middle-tier may also be accessed within Reports to perform various data formatting tasks, such as generating a barcode. (iii) J2EE developers also now have the freedom to plug-in their own custom destinations to distribute reports using the open Java API for Pluggable Destinations and Notifications, if their destination is not supported out-of-the-box.
- *Unlimited Data Formatting* - Oracle9*i*AS Reports Services performs high quality data publishing in a number of different formats including Multibyte Support - Unicode, Adobe's Portable Document Format (PDF), HTML, HTML Cascading Style Sheets (HTML CSS), XML, Postscript, PCL, Delimited Text, Excel (CSV), and Rich Text Format (RTF). Oracle9*i*AS Reports Services allows developers to easily build an almost unlimited number of report formats without resorting to the complexity of nested report modules or complex procedural code. Pre-defined templates assist with the rapid creation of reports, and generated reports support advanced Web features such as embedded hyperlinks, bookmarks, and page-on-demand viewing. Reports can be customized via parameters input at runtime by the user, or by merging supplementary XML report definitions with the original for truly customized output.
- *Access to any data*  - In Release 2, Oracle9*i*AS Reports Services can combine data from multiple data sources into a single report including Oracle8i, 9*i* Databases, XML Feeds, Oracle Express and any JDBC-enabled Data Sources. Developers can even add custom data sources using the open Java API for Pluggable Data Sources. This means that report consumers have access to the right data in their report - regardless of where it came from.
- *Scheduled Execution, Report Bursting and Distribution* - Report execution can be initiated directly by a user from a web-browser, from the command line. New in Release 2, Reports can be kicked off from an Oracle database event such as a database trigger (when some data changes), or from a message in an Advanced Queue. Report output can be automatically burst based on repeating data and separately disseminated to appropriate users. It may also be distributed to the browser, printer, email, and Oracle9*i*AS Portal (either as a Portlet or as content for a Page Area) - either separately or simultaneously.

ORCL00010770

Oracle9*i*AS Discoverer

Oracle9*i*AS Discoverer is an intuitive ad-hoc query, reporting, and analysis tool that empowers people to make better decisions. Oracle9*i*AS Discoverer offers on-demand access to e-business data at Internet speeds, to all levels of the organization. Oracle9*i*AS Discoverer provides powerful query and analytical capabilities that can be used for example, to find out who are the top 5 customers or suppliers for a company, how their supply timelines have changed from the year prior, and how much money was spent with them. Oracle9*i*AS Discoverer uses a Web browser based interface that makes it easy for novice and experienced users to create queries, navigate through data, and publish report results. In Oracle9*i*AS Release 2, Discoverer has been enhanced with a number of important features:

- *Export to Oracle9iAS Report Services* - Oracle9*i*AS Discoverer provides "Export to Reports" capability in XML format. The SQL and the formatting will be transferred to Oracle9*i*AS Reports Services.
- *Enhanced Scheduling Facilities* -  Release 2 introduces enhanced scheduling capabilities that support scheduling workbooks/worksheets from a menu/wizard, scheduling workbooks when the query threshold is exceeded, a schedule 'repeat' interval, notification at login if scheduled workbooks are available and sharing of scheduled result sets whereby using database security, users are only permitted to view the shared data they are allowed to see.
- *Integration with* Oracle9*i*AS *Portal* -  Oracle9*i*AS Discoverer enables users to publish their favorite worksheets or list of workbooks to Oracle9*i*AS Portal. Release 2 delivers two distinct Discoverer portlets - a list of workbooks portlet and a worksheet portlet. The "list of workbooks portlet" includes the names of workbooks available from Discoverer. The "worksheet portlet" is an individual worksheet that contains only data (table or cross-tabular report), only a graph, or both.
- *Discoverer Viewer Support for Oracle Web Cache* -  Discoverer Viewer will support the Oracle9*i*AS Web Cache technology by caching the page content when a worksheet is opened in viewer. The major benefits include faster response times and a reduced number of sever hits, faster download by compression techniques, and load balancing by forwarding traffic to less loaded servers.
- *Support for Single Sign On-*  Oracle9*i*AS Discoverer supports Oracle's Single Sign On technology. Users will log in to Oracle9*i*AS Discoverer once, and be able to access multiple databases, end user layers, ,and other SSO compliant applications such as Oracle9*i*AS Portal – without logging in again during the same browser session.

Oracle9*i*AS provides you with the ability to capture, analyze and extract business intelligence about how users are accessing your applications or Web Sites.

ACCELERATE WEB SITE PERFORMANCE

Currently, Internet Applications are facing growing demands for complex e-business information from growing numbers of users. As Internet Applications are required to perform more computation under growing transactional loads, the performance of these applications declines sharply. Oracle9*i*AS includes key features to guarantee the performance and scalability of Internet Applications. Specifically, Oracle9*i*AS is designed to transparently improve the performance of J2EE Applications and Web Services. Further, Oracle9*i*AS also offers a

ORCL00010771

Web Cache to further improve performance and scalability by caching both static and dynamic content. Let us examine these pieces in greater detail.

Oracle9*i*AS Performance Strategy

Applications developed to J2EE and Web Services standards and deployed on Oracle9*i*AS run faster with **zero application changes.** They can support more users with less hardware either on a single CPU or on multiple CPUs. Oracle's Performance strategy is focused around the following objectives:

- *All Aspects of Performance* - To improve performance, Oracle focuses on all aspects of its Oracle9*i*AS runtime including its J2EE and Web Services code path; the network invocation costs (HTTP, RMI, RMI-over-IIOP, SOAP); to efficiently use in-process calls where appropriate for highly optimized local requests; to manipulate data; and for highly efficient transaction scheduling.
- *All Types of Hardware* - Since Application Servers are deployed on a variety of different kinds of hardware including 1-2 CPU commodity hardware and high end SMP clusters, Oracle designs its Application Server to provide fast performance both on low end hardware and high end clusters.
- *Application Transparent* - To make applications run efficiently, Oracle ensures that applications require **no modification,** and reduces the number of parameters that administrators need to configure to make applications run efficiently. We also design the Application Server in this fashion to ensure that static parameters which are optimized for a specific user and transaction load do not constantly need to be adjusted as loads on the system grows.
- *Caching to Further Improve Performance* - Finally, Oracle provides a number of different types of caching to improve application performance - (i) A Java Object Cache to cache frequently accessed Java objects and to cast SQL query results into Java objects to reduce type conversion overhead; and (ii) Web Caching to cache static and dynamic web content for performance.

Now let us look at these features in greater detail.

Oracle9*i*AS Performance Features

Oracle9*i*AS is highly optimized to make J2EE Applications and Web Services run extremely efficiently. Further, the various other services in Oracle9*i*AS that use this infrastructure - portals, wireless, integration, and business intelligence - transparently inherit these optimizations to run fast.

Some of the key features are:

- *J2EE and Web Services Optimizations* - Oracle9*i*AS Release 2 provides a number of optimizations including code path, network, in-process invocation, data management, automatic detection and caching of EJBs and SQL statements, optimized simple and complex database mappings, and JDBC database access to improve the performance of J2EE Applications and Web Services.
- *Highly Optimized Transaction Scheduling* - Oracle9*i*AS Release 2 also provides two important new transaction modes for highly optimized performance - (i) Serializable transactions which very efficiently delegates serializable transactions to the database for performance and eliminates latch contention between EJBs; (ii) Read-Only EJBs is a transaction mode that provides very efficient request processing for read only Entity Beans.
- *Java Object Caching* - Oracle9*i*AS Release 2 provides a Java Object Cache designed to reduce SQL-to-Java overhead (database access overhead) and to improve access performance for shared Java objects. The Java Cache is distributed and can be accessed In-Process, across process boundaries on a single box, and across processor/machine boundaries. To improve performance, Java Object Creation is distributed to avoid bottlenecks. The cache is configurable, allowing objects to

ORCL00010772

be grouped, pooled, pinned, and paged as necessary. Finally, the Java Cache is clusterable with updates that can be synchronized across clusters.

In this section, we provide a brief overview of some of these optimizations; more details are captured in the Oracle9*i*AS Release 2 J2EE Technical Whitepaper.

Oracle9*i*AS Web Cache

The Oracle9*i*AS Web Cache is designed to further improve performance and scalability of Internet Applications. Today's leading high-volume Web sites are faced with the challenge of serving thousands of users concurrently and providing them with accurate data in a dynamic environment where content changes rapidly. To address the performance and scalability demands of such Web sites, Oracle9*i*AS provides a Web caching solution with the unique capability of caching both static and dynamically generated Web content. .

Oracle9*i*AS Web Cache operates as a **caching reverse proxy server** that is situated in front of the Oracle HTTP Server. It improves performance of Web server instances by storing frequently accessed pages in memory, eliminating the need to repeatedly process requests for pages from the Web server, the applications, or the Oracle database. Unlike existing caching proxy servers, which are capable of handling only static content, Oracle9*i*AS Web Cache accelerates the delivery of both static and dynamic Web content. Using the administrative tools provided, Oracle9*i*AS Web Cache can be configured with custom rules used to determine when to invalidate and refresh cached content.

Oracle9iAS Web Caching Strategy

Oracle9*i*AS' Web Caching Strategy is based on the following objectives:

- *Improve Performance* - The Web Cache is primarily designed to provide a very cost effective solution to significantly improve the performance and scalability of heavily loaded Web sites. A single instance of Oracle9*i*AS Web Cache can sustain throughput levels equal to that of tens or even hundreds of Web servers serving a Web site. Oracle9*i*AS Web Cache can cache static HTML; dynamic web pages (generated with any page generation technology - PERL, JSP, ASP); web page fragments; GIFs, JPEGs, and other web page fragments; and not secure and secure content (HTTP-S content).
- *Improve Scalability* - Oracle9*i*AS Web Cache can also improve the scalability of Internet Applications by offloading application servers, requests being processed out of the cache. The Web Cache scales applications with less hardware and provides Surge Protection to allow predictable response even when user loads peak without needing extra hardware.
- *Improve High Availability* - The Web Cache also improves the perceived availability of Web Sites to users and in Release 2 provides a number of new high availability features including advanced Load Balancing, and Partitioned Cache Clusters.
- *Reduce Latency* - The Web Cache is integrated with Content Delivery Networks and in Release 2 also provides Geographically Distributed Caches to reduce network latency for users accessing remote Internet Applications.

Finally, in Oracle9*i*AS Release 2, the Web Cache is integrated with all aspects of Oracle9*i*AS including the JSP Translator, Servlet Engine, Web Services Environment, Enterprise Portal, Wireless and Business Intelligence features to improve performance of all these applications. Let us now consider the Web Cache enhancements in Oracle9*i*AS Release 2.

ORCL00010773

Oracle9iAS Release 2 Web Caching Features

In addition to providing unique caching capabilities to increase performance and scalability of Web sites, Oracle9*i*AS Web Cache also provides a number of other powerful features to ensure that Web sites using Oracle9*i*AS technology provide consistent and predictable responses:

-  *Page Fragment Caching and Dynamic Content Assembly* - New Edge Side Includes (ESI) functionality enables Oracle9*i*AS Web Cache to aggregate portions of Web pages and reassemble them on the fly for individual users. ESI is the result of a joint development effort between Oracle and Akamai, and has now been proposed as an open standard. ESI is a simple markup language that application developers use to identify content fragments for dynamic assembly in edge servers, such as Oracle9*i*AS Web Cache and third-party content delivery networks (CDNs). The "partial-page" caching functionality that ESI enables is especially useful for Web pages that contain targeted banner ads, individual account information, or other user-unique elements that should not be cached. With ESI, the edge server can store all of the common elements of a Web page and query the database or other content repositories only for any highly personalized objects. By uniquely identifying common elements (e.g., stock quotes, weather reports, news, graphics, etc.) that can be shared among different Web pages, only one copy of each element needs to be cached, invalidated and revalidated, thus saving valuable resources across all layers of infrastructure. In Release 2, additional ESI extensions provide the ability to perform and cache XSLT transformations on XML documents.
-  *Surge Protection* - Oracle9*i*AS Web Cache has a surge protection feature to prevent back-end Web servers from overloading when request volumes are high. You can configure the maximum number of concurrent requests that are served to a Web server. This feature limits the number of concurrent requests given to a Web server, preventing Web servers from being overloaded with requests and allows them to work at their peak efficiency. If the number of requests exceeds the limit set for a Web server, the incoming requests are placed in a queue.
-  *Content-Aware Web Server Load Balancing and Binding* - Oracle9*i*AS Web Cache provides content aware load balancing for distributing HTTP requests. In the case where Oracle9*i*AS Web Cache is unable to fulfill a request from its memory cache, it distributes the cache miss to the most available and highest performing Web server in the cluster. Further, Oracle9*i*AS Web Cache supports application affinity. Web sites that use session IDs or cookies to bind a specific user session to a particular Web server so that the state of the session can be maintained for a period of time are fully supported with Oracle9*i*AS Web Cache.
-  *Partitioned Web Cache Clusters* - Oracle9*i*AS Web Cache clustering enables multiple cache instances to work together as a single logical cache. Clustering Web Cache instances together increases the capacity of the cache and the number of requests that can be served concurrently. By automatically partitioning the sites content across multiple cache peers and replicating the most popular content among those peers, Oracle9*i*AS Web Cache can store more content and can support more browser connections. In addition, cache clustering extends application availability by supporting failure and recovery detection of cache instances. If a cache instance fails, other members of the cluster detect the failure and the cached content is automatically re-partitioned among the remaining cluster peers. Since the most popular content is always resident in each member's cache, the benefit of this failover mechanism is that system administrators can service the "downed" cache at their convenience, without disrupting service. Furthermore, management is simple, a cache cluster uses one set of cacheability rules and one set of invalidation rules for all cluster members.
-  *Transparent Failover* - Oracle9*i*AS Web Cache detects failures of back end Web servers and automatically routes requests around the failed server by distributing the request load amongst the

ORCL00010774

remaining Web servers in the cluster. Once a failed Web server has been identified, Oracle9*i*AS Web Cache will periodically check its status. When the failed server restarts, Oracle9*i*AS Web Cache automatically routes requests to it again. No intervention is required.

Oracle9*i*AS Web Cache can also accelerate delivery of content generated by third party application servers including Sun Netscape iPlanet, Microsoft IIS, BEA WebLogic, and IBM WebSphere. To summarize, Oracle9*i*AS includes key features to guarantee the performance and scalability of Internet Applications.

RELIABILITY, AVAILABILITY, AND SCALABILITY

Oracle9*i*AS not only provides the industry's most productive development environment for Internet Applications but also provides a number of new features that make it a mission critical platform for highly scaleable, highly available Internet Applications. Oracle9*i*AS provides a number of enhancements to address the three areas that determine how reliably Internet Applications can be deployed - Reliability, Availability, and Scalability (RAS). Let us examine these in more detail.

Scalability

Oracle9*i*AS provides the industry's most highly scalable Application Server. The scalability of a system refers to how well it can respond as user demands increase and it is limited by any bottleneck in the system. With application servers, the primary scalability bottlenecks are CPU-constraint (where the system becomes constrained because its processor becomes completely consumed) and memory-constraint (where the system becomes constrained because it is memory bound).

Oracle9iAS Scalability Strategy

Applications developed to J2EE and Web Services standards and deployed on Oracle9*i*AS simply scale better with **zero application changes** - they can support more users with less hardware either on a single CPU or on multiple CPUs. Users can start by deploying these applications:

- *On a single Java VM on a single CPU* - Oracle9*i*AS optimizes the use of scarce resources such as threads, memory, and database connections to improve scalability.
- *Clustered Java VMs on a single CPU* - As loads on the system increase, users can add more Java VMs and cluster them together with transparent load balancing, connection routing and state failover.
- *Horizontal Scalability with Multiple CPUs* - As loads increase still further, users can add more CPUs each with 1 more Java VMs with transparent load balancing, connection routing and state failover (called horizontal scalability)
- *Vertical Scalability with Clustered Hardware* - Alternately, users can move their Application Server and Applications from low-end 1-2 CPU systems to higher end SMP-clustered hardware (called vertical scalability). All of these different scalability strategies can be accomplished without changing the application.

Let us look at these different strategies in greater detail.

Single CPU Scalability

Oracle9*i*AS provides very efficient scalability supporting large numbers of users on a single CPU - supporting more users with on a single Java VM, and allowing clusters of Java VMs to be started to handler more users or greater loads. There are four important ways in which it provides such scalability:

ORCL00010775

- *Single JVM - Threading Model* - Oracle9*i*AS' threading model lends itself nicely to shared access to resources. Specifically, Oracle9*i*AS can maintain a hierarchy of J2EE Containers in the following manner: (i) A single hardware instance (box) can run multiple Oracle9*i*AS instances; (ii) A single Oracle9*i*AS instance can run multiple Java Virtual Machines (JDKs); (iii) A single Java Virtual Machine or JDK can start and maintain multiple threads; (iv) Each thread can run a single J2EE Application module (JSP, servlet or EJB). Additionally, Oracle9*i*AS' J2EE Container pre-starts and maintains a pool of threads to process requests. When a specific request is received from a J2EE client, an idle thread is selected and the specific J2EE module is loaded and executed on that thread. When the J2EE module has completed servicing that request, the thread is freed up and placed back in the pool.
- *Inbound Request Pooling (Connection Pooling)* - Oracle9*i*AS also maintains a pool of socket connections to handle requests coming from a J2EE client, a Web Server, a Load Balancer, or another J2EE Application module. If the request can be serviced in a request-response fashion (as in a stateless, HTTP 1.0 fashion), the socket is reused when the response is written back out on the socket. If the socket needs to be "kept-alive" (as in HTTP 1.1 or for clustering purposes), the J2EE container maintains the socket across multiple requests until it can be freed up.
- *Outbound Request Pooling (Database Connection Pooling):* To eliminate database access overhead, Oracle9*i*AS' J2EE Container pre-starts and maintains a pool of JDBC connections to a database. On specific requests, it selects/reuses an idle JDBC connection from the pool - when it is complete, it places the connection back into the pool to be reused by other clients. Connection pooling also improves mid-tier and database scalability since it allows multiple clients to be multiplexed onto fewer database connections.
-  *Multiple JVM and Clustering for Scalability* - When user or transaction loads grow beyond the volume that can be handled by a single JVM, multiple Java VMs can be started up each with one or more Oracle9*i*AS instances. These instances can be clustered together - requests are then transparently distributed to the instances in the cluster. The cluster, with the combined processing resources of all of the hosts, then appears as a single, highly available, logical container. Session State of both servlets and EJBs is replicated across all instances in the cluster providing failover in the case of instance failure. (More information on the clustering architecture is provided later in this document).

Multi-CPU Horizontal Scalability

When user or transaction loads grow beyond the volume that can be handled by a single CPU, multiple CPUs can be added. Multiple Java VMs can be started up on these CPUs, each JVM with one or more Oracle9*i*AS instances. These instances can be clustered together. Requests can then be transparently distributed to across these CPUs, JVMs, and Oracle9*i*AS instances by a hardware load balancer.  Session State replication and Application Failover is also supported for both servlets and Enterprise JavaBeans.

Vertical or Hardware Scalability

When faced with scalability bottlenecks, administrators can choose not to add more CPUs but to migrate the application server to a higher end hardware configuration (if a system is CPU-bound) or add more memory (if a system is memory-bound). To support such vertical scalability, Oracle9*i*AS provides a number of features, specifically:

- *Availability on Broad Range of OS/Hardware Platforms* - Oracle9*i*AS is available on a broad range of hardware platforms scaling from low-powered desktop system, low-end uniprocessor machines to high-end SMP clusters and on all major Operating Systems including Solaris, HP-UX, AIX,

ORCL00010776

Tru64, Windows (2000, NT, and XP underway) and LinUX. This allows system administrators to upgrade hardware and operating systems without rewriting their applications.

- *Single Node or Multi-Node Clusters* - Oracle9*i*AS' J2EE Container also provides "cluster" support independently of the hardware platform or OS being used (i.e. its cluster capability does not require a specific SMP hardware configuration). This allows J2EE Developers and Administrators to be able to leverage these features without being tied to any OS/hardware platform.

Load Balancing

Load distribution or load balancing essentially means how requests from clients can be distributed across multiple Oracle9*i*AS instances on a single CPU or on multiple CPUs. Oracle9*i*AS Release 2 provides a number of advanced capabilities including:

- *Load Balancing at HTTP Server* - The web server uses a simple but efficient mechanism to load balance between HTTP server processes within a single instance of the service. The master HTTP server process does not service client requests itself but spawns and monitors a group of child processes which take turns accepting HTTP requests from a shared socket by using a mutex. Once a child receives a request but before it begins servicing it, it releases the mutex which can be acquired by another child - as a result, access to the socket is serialized but children may service requests in parallel. Additionally, Oracle HTTP servers can be run on multiple nodes where client requests can be load balanced over the separate host instances using a variety of techniques including DNS round robin or a dedicated hardware load balancer (see below).
-  *Load Balancing at Oracle9iAS J2EE Container* - Servlet and EJB container instances load balance requests either across instances on a single node or across multiple nodes using a variety of load balancing algorithms. There are several load balancing enhancements in Release 2 including those for RMI, RMI-over-IIOP, and SOAP requests.
-  *Integration with Third Party Load Balancing Products* - Finally, Oracle9*i*AS Release 2 has been certified with load balancing Appliances - Cisco Local Director, BigIP, and Alteon - for stateless and stateful load balancing.
- *Connection Re-direction and Node Affinity* - Finally, for stateful applications, Oracle9*i*AS supports standard facilities such as cookies and dynamic URL-rewriting to bind and redirect clients to an existing session on a specific instance.

Offloading Application Servers with Web Cache

Finally, Oracle9*i*AS also provides the Web Cache to further improve the scalability of Internet Applications while offloading the Web Servers and Application Servers. The Web Cache intercepts requests directed to Web servers and serves cached pages or partial pages directly from memory, thereby greatly decreasing the load on Web Servers and Application Servers. When the Web cache is used, Web servers only receive requests for new content and for content that is not cacheable.

To summarize, Oracle9*i*AS provides a number of scalability strategies to improve the scalability of applications on a single CPU or on multiple CPUs without requiring application changes.

High Availability and Reliability

Availability of an overall system or of an individual system component is defined as the percentage of time that it works normally or alternatively the mean-time-to-failure for the system. Oracle9*i*AS Release 2 is designed to provide a highly available and highly reliable infrastructure to deploy applications.

ORCL00010777

Oracle9iAS High Availability Strategy

In a similar manner to the Oracle Database, Oracle's High Availability strategy for the Application Server focuses on two primary elements:

- *Zero Unplanned Downtime* - Zero Unplanned Downtime describes a set of facilities that are aimed at ensuring that the application does not have any downtime even if one or more instance of the application server on which the application is deployed fails.
- *Zero Planned Downtime* - Zero Planned Downtime describes a set of facilities that are aimed at eliminating the need to take down an application in order to perform planned maintenance operations such as patching, upgrade, and maintenance.

Let us look at each of these elements in greater detail.

Zero Unplanned Downtime

Oracle9*i*AS Release 2 provides a number of features for Zero Unplanned Downtime, specifically:

- *No Single Point of Failure* - Oracle9*i*AS can be deployed in an architecture that exposes no single point of failure. This means that despite the failure of any node in the system, Oracle9*i*AS will continue to function and service client requests. NEW In Release 2, the J2EE containers are all homed on either local or remote Oracle HTTP Servers as load balancing mount points ensuring that the system has complete redundancy in the event of a failure of any HTTP Server or J2EE Container.
- *Automatic Connection Rerouting* - When any specific Oracle9*i*AS instance fails, the configurable load balancing facilities at every tier described earlier ensures that client requests can be transparently re-routed to an alternate instance. For stateful applications, Oracle9*i*AS can re-route connections from a "stateful" client to an existing session on a specific Oracle9*i*AS instance or in the event of a failure another instance belonging to the same cluster (see the section on clustering below).
- NEW *Automatic Death Detection - Fast Restart Architecture* - With Release 2, Oracle9*i*AS has integrated fault monitoring facilities to both detect the death or failure of a specific instance and restart it in order to minimize Mean-Time-to-Recovery for that instance. Oracle Process Manager and Notification Service (OPMN) provide this facility. The process monitoring architecture consists of three facilities - (i) **Fault Monitors**, written in Java or C, to periodically ping various processes associated with an Oracle9*i*AS instance; (ii) The **Notification Service** which channels events from different instances via a publish-subscribe mechanism to other services that are interested in receiving these event notifications. The events are all communicated over HTTP between different processes. (iii) **Instance Recovery** - Specific instances can be recovered either by administrative intervention from Enterprise Manager which receives event notifications; or by an automated job which can be used to restart the instance. (iv) **Dynamic Registration and Transparent Load Balancing** - When the instance is recovered, the instance dynamically registers itself with the appropriate load balancing mount point and requests are automatically routed to it. Specifically, the Apache mod_OC4J, which is the new Apache module that routes requests from the Oracle HTTP Server (OHS) to J2EE Containers, works in conjunction with the OPMN and DCM Services to keep its routing table updated so that it load balances across only live OC4J processes in an OC4J instance.
- NEW *Enterprise Manager and Distributed Instance Restart and Reconfiguration* - When Oracle Enterprise Manager receives an event notification via the OPMN Notification Service, an administrator can restart the instance. Enterprise Manager in turns invokes a service called the Distributed Configuration Management (DCM) Service. DCM performs two tasks - first, it identifies the

ORCL00010778

configuration information needs to be restarted with by looking up a configuration repository which can either be a local XML file or the infrastructure repository where cluster-wide configuration information is maintained. DCM then generates and propagates a set of events to OPMN - together these two services ensure that the Oracle9*i*AS instance gets the correct startup/shutdown commands and is appropriately configured when started. When the instance registers itself dynamically with the appropriate cluster and load balancing point, DCM and OPMN synchronize this information maintaining an up-to-date topology of the deployed architecture. OPMN and DCM are illustrated conceptually in the figure below.

- *Transparent Application Failover (TAF) and Database State Management* - Finally, every Oracle9*i*AS middle-tier communicates with the Oracle Database using a connection pool. Every Application writes its long-lived state persistently to the Oracle Database. With Oracle8*i* and 9*i* Databases including RAC, when a specific node in the Database fails, all state maintained in the Database is transparently "failed-over" to another node. When Oracle9*i*AS is used in combination with the Oracle Database, the middle-tier servers have the ability to transparently re-route JDBC connections and database requests to the failed-over node. Depending on the degree of connection redundancy, Oracle9*i*AS can provide various degrees of TAF with the database - (i) Cold Failover - A user is connected to the failed over database node but needs to re-establish the session and restart queries; (ii) Warm Failover - A user is connected to the failed over database node, the session is automatically re-established but the user needs to restart queries; and (iii) Hot failover - A user is re-connected, the session is re-established, and the user can simply continue to execute queries right from the point where the cursor was prior to failover.

ORCL00010779

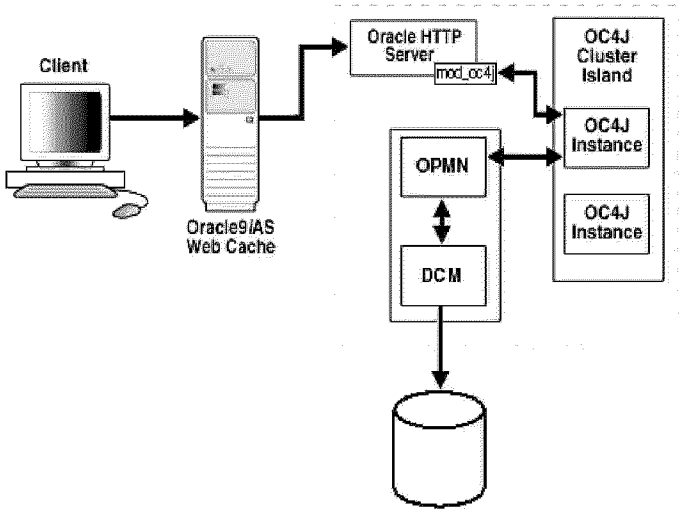


Figure 8 - Process Manager and Distributed Configuration Management

To summarize, these different services in Oracle9*i*AS Release 2 provides two important facilities for Applications - (i) *Zero Application Downtime - Longer Mean-Time-to-Failure* - Prevent applications from having any downtime even if any specific Oracle9*i*AS instance has a failure. This significantly increases the Mean-Time-to-Failure for Applications. (ii) *Faster Failure Recovery - Shorter Mean-Time-to-Recovery* - They also ensure that in the event of a failure, the Oracle9*i*AS instance can be recovered much more quickly and efficiently decreasing the Mean Time-to-Recovery from faults.

Zero Planned Downtime

To reduce the need to take down an application deployed on Oracle9*i*AS in order to perform planned maintenance operations such as patching, upgrade, and maintenance on the Application Server itself, Release 2 provides a number of new facilities:

- *Hot Deployment of Applications* - First, Oracle9*i*AS allows for hot deployment of J2EE Applications - JSPs, Servlets, EJBs, and Web Services - to an instance of the Application Server that is already up and running. For instance, when an application developer changes an EJB module that has already been deployed, the developer does not need to either re-deploy the EJB or restart the server. The user needs to simply edit the server.xml configuration file, the server reads the file, and automatically picks up the changes. Further, Oracle9*i*AS has class loading facilities that enables it to distinguish between different versions of J2EE artifacts that are deployed to it allowing sessions executing with the earlier version of the EJB Module in the example above to complete gracefully when the updated EJB is deployed.
- *Dynamic Reconfiguration* - Second, Oracle9*i*AS also supports the ability to dynamically reconfigure J2EE Container instances including clusters. All configuration information including clustering information is specified in XML files and Application Server instances do not need to be re-bounced when configuration changes are made. Further, note that when multiple systems are

ORCL00010780

configured in a cluster, cluster-wide configuration information is managed by DCM which can also synchronize this information with a repository and taked.

- *Rolling Upgrade, Cycling of Instances* - Third, when an Application is deployed to a cluster of Oracle9*i*AS instances, any specific instance in the cluster can be taken off-line, modified, upgraded, and rejoined to the cluster without any application downtime - we call this "rolling upgrade of instances." Further, users can also use Enterprise Manager's Job facilities to kick off scheduled jobs to perform these planned maintenance activities.

Oracle9*i*AS Clustering

Clustering with Oracle9*i*AS essentially means the use of a group of Oracle9*i*AS Servers that coordinate their actions in order to provide scalable, highly available services in a transparent manner. Clusters enforce homogeneity between member instances so that a cluster of application server instances appears to function as a single virtual instance. With Application Servers, clustering provides two important benefits - (i) Load Balancing where the distribution of a workload is distributed in the most optimal fashion across a set of instances for efficient processing; and (ii) Failover where requests can be handled by other instances in a cluster when a specific instance fails without interrupting the application.

Oracle9*i*AS Release 2 provides a broad set of clustering facilities with several new capabilities:

- *Oracle9iAS Clusters* - In Release 2, the clustering facilities have been broadened to include not just the J2EE Container but the Web Cache, HTTP Server, J2EE Container (including all its facilities - JSP, Servlet, EJBs, JNDI, JMS), and Web Services facilities. Release 2 introduces two key new concepts - application server instances and clusters. An Oracle9*i*AS instance represents the collection of Oracle9*i*AS component processes and the configuration data required to start those processes. A simple way to think of an Oracle9*i*AS instance is to equate it with a runtime instantiation of a single Oracle9*i*AS environment. A cluster is a collection of Oracle9*i*AS instances with identical cluster-wide configuration and cluster-wide application deployment. When you configure an Oracle9*i*AS instance you can choose to configure it as part of at most one cluster. Instances belonging to a cluster can be deployed on a single CPU or on multiple CPUs.

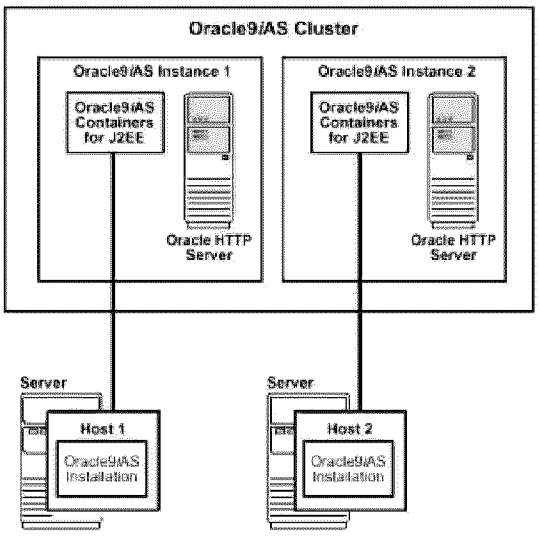


Figure 9 - Oracle9*i*AS Cluster

ORCL00010781

- *Web Cache, HTTP Server, and Stateless J2EE Clustering:* With Oracle9*i*AS Release 2, Web Cache, HTTP Server, and J2EE Container (Servlet Container and EJB Container) instances can be clustered together to service stateless requests. Each tier of the application server can load balance requests to the next - (i) With Web Cache or HTTP Server instances, either one Web Cache or HTTP Server instance can load balance requests to the other instances in the cluster, or an external load balancer can direct the requests. (ii) With the J2EE Container, the Apache mod_OC4J load balances requests across the instances. This model is supported for stateless JSP, Servlets, and stateless Session EJBs. (iii) Should any of these instances fail, as described earlier, the load balancing facilities automatically re-route requests to the other instances in the cluster.
- *Stateful Servlet, EJB Clustering*: Oracle9*i*AS Release 2 also addresses the far more difficult problem of clustering Application Servers together when the requests are stateful. In this case, the clustering facilities not only provide load balancing but also failover of session state between instances of the cluster. If Oracle9*i*AS' J2EE container has been configured to support clustering, it is registered along with a group of other instances as a cluster with a common cluster ID. In the case of an instance failure, the load balancer recognizes the instance failure, identifies other servers with the same cluster ID, and routes the connection to one of those servers using the load-balancing algorithm. The session state from the failed J2EE container has already been replicated via IP-Multicast across all instances in the cluster, and, as a result, the client is able to access the session again. With Oracle9*i*AS Release 2, this model is supported for both stateful JSPs and servlets, stateful session EJBs, and Entity Beans.
- *Complete Cluster Management* - Finally, Oracle Enterprise Manager provides a comprehensive set of GUI-based facilities to manage Oracle9*i*AS clusters in Release 2. These facilities include (i) **Cluster Creation** - The ability to Create a cluster; to Destroy it; to Add or Remove instances from the cluster; to Start, Stop, or Restart a cluster as a single operation. (ii) **Cluster Configuration -** The ability to configure and re-configure a cluster of instances homogeneously as a single operation without configuring each component; and to add an instance to a cluster and to have it inherit the clusters configuration. (iii) **Monitoring -** The ability to carry out cluster-wide performance and fault monitoring. (iv) **Application Deployment -** The ability to deploy a J2EE Application to a cluster without deploying the application to each instance in turn.

Based on the description above, Oracle9*i*AS Release 2 provides a number of high availability strategies to improve the availability of applications reducing both planned and unplanned downtime. Further, in summary, Oracle9*i*AS Release 2 provides a complete infrastructure to deploy scalable, highly available and reliable applications.

MANAGE AND SECURE YOUR WEB INFRASTRUCTURE

Finally, Oracle9*i*AS Release 2 includes a comprehensive management infrastructure designed to streamline and lower the cost associated with the management of Internet systems infrastructure, applications, and users.

Oracle9*i*AS Management Strategy

Oracle has a comprehensive management strategy based on three specific objectives:

- *Lower Cost of Ownership by Centralizing Management* - Oracle9*i*AS provides administrators with a single browser-based management tool to install/configure, monitor, manage, and troubleshoot all Oracle9*i*AS facilities, on one or more Oracle9*i*AS instances, on one or more CPUs.
- *Provide Comprehensive User Management* - With the Internet, the volume of users accessing applications and hence needing to be managed increases sharply. To lower the costs associated with managing users and their security privileges, Oracle9*i*AS provides a single security and directory framework for all Oracle9*i*AS facilities and for all applications deployed on Oracle9*i*AS.

ORCL00010782

- *Provide Comprehensive Application Management* - Finally, Oracle9*i*AS also provides a comprehensive set of facilities to manage applications deployed on the Application Server.

Let us examine each of these elements in greater detail.

Management, Administration and Monitoring

Oracle Enterprise Manager is a system management tool for centrally managing your Oracle platform. Combining a graphical console, Oracle Management Servers, Oracle Intelligent Agents, common services, and administrative tools, Oracle Enterprise Manager provides a comprehensive systems management platform for managing both Oracle9*i*AS and Oracle9*i* Database. Oracle9*i*AS Release 2 introduces major new enhancements to provide a fully integrated monitoring, management and diagnostics environment specifically for Oracle9*i*AS. The brand new web-based administration interface is used to manage all Oracle9*i*AS components and features and supports:

- *Comprehensive Monitoring* - Oracle Enterprise Manager provides a comprehensive set of facilities to monitor the status (up/down) of Oracle9*i*AS facilities; to measure their resource utilization; to diagnose faults; and to monitor the performance/throughput of Oracle9*i*AS facilities.
- *Systems Configuration and Management* - Based on information gleaned from monitoring the systems, Oracle Enterprise Manager provides administrators with facilities to centrally control the configuration (and re-configuration) and administration of one or more Oracle9*i*AS instances.
- *Application Management* - Finally, Oracle Enterprise Manager provides facilities to deploy, track, and administer J2EE Applications and Web Services.

By making the management facilities accessible from a HTML browser, administrators can mange Oracle9*i*AS either locally or remotely (for instance when out-of-the-office), including through firewalls. Let us look at each of these elements in greater detail.

Comprehensive Monitoring

Accurate, timely and relevant performance and troubleshooting data is absolutely essential in maintaining and monitoring the health of your production web sites and applications. Oracle9*i*AS Release 2 provides a host of critical performance data that is essential to tune your application server, identify resource availability issues, locate bottlenecks or help tune your application server to achieve the maximum throughput and the minimum response time possible.

- *Dynamic Monitoring Service* - Oracle9*i*AS has been fully instrumented with a service called the Dynamic Monitoring Service (DMS) to collect the performance data discussed above. DMS provides a comprehensive set of built-in performance metrics to automatically measure runtime performance statistics and resource consumption such as CPU and Memory being utilized, I/O, Response Time, Requests/Second, Transaction Throughput and several other metrics. DMS allows you to monitor both the duration of important phases of request processing as well as status information such as the number of requests being handled at any given moment. In addition, you can easily add monitoring to your own applications via the DMS library and API. The performance metrics are measured automatically and continuously using efficient performance instrumentation hooks - no extra configuration is required. In Oracle9*i*AS Release 2 users can view DMS metrics through a browser as well as through the OEM console.
- *Fault Monitoring* - Oracle Enterprise Manager also uses the underlying fault monitoring, "fast restart" architecture described above - (i) To monitor whether one or more Oracle9*i*AS instances are up and running; (ii) To diagnose faults on a specific instance; and (iii) To allow administrators to intervene and restart instances in the event of failures.

ORCL00010783

- *Oracle Enterprise Manager* - The new administration interface provides a comprehensive aggregate view of the current and historical status and well being of your Oracle9*i*AS environments. It allows you monitor status, usage and performance data for your entire Oracle9*i*AS environment including: (i) Instance and component availability and uptime statistics; (ii) Resource utilization including CPU and memory; (iii) Server load and responsiveness including user volumes, active requests, request throughput and processing time, error rates

The Oracle9*i*AS Home Page within Oracle Enterprise Manager provides an aggregate view of all information for your Oracle9*i*AS instance - administrators can easily drill down to lower levels for more specific information on each component.

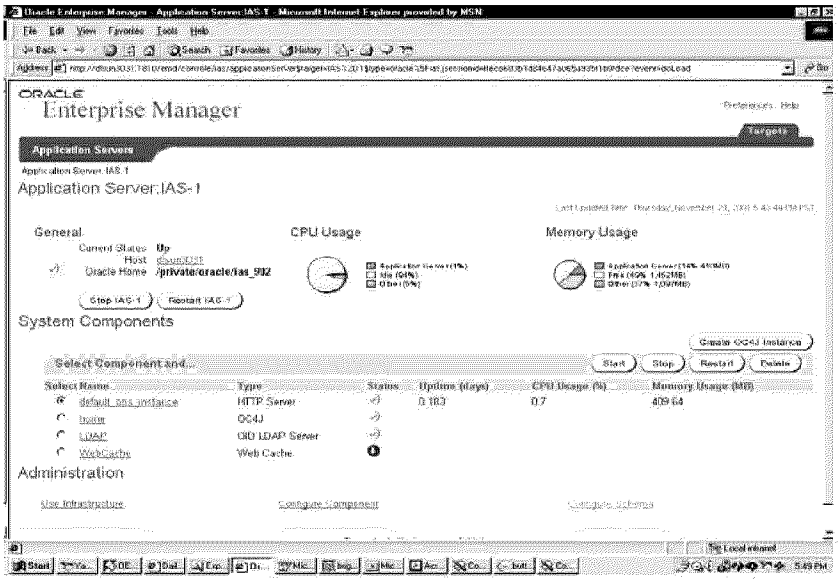


Figure 10- Oracle9*i*AS Management Home Page

A Single Point for Distributed Configuration and Management

The brand new administration interface provides a central point of control for configuring and administering a distributed Oracle9*i*AS environment. It provides a very intuitive, high performance Servlet-based HTML interface that allows administrators to effectively manage the Oracle9*i*AS environment including:

- *Discovery* – It automatically discovers all Oracle9*i*AS instances and their associated components on a target node for management by the Oracle9*i*AS administration tools.
- *Management* - It provides the ability to start and stop all Oracle9*i*AS components
- *Cluster Management* – It provides a single point to create and manage Oracle9*i*AS farms and clusters. Clusters provide a central point for managing multiple Oracle9*i*AS instances that host a common set of J2EE applications.
- *Configuration* – It allows users to perform common administrative operations for all Oracle9*i*AS components, including configuring ports and log files.
- *Centralized Configuration Information - Infrastructure Repository* - All configuration and deployment information is kept external to any Oracle9*i*AS instance or any cluster of instances in a

ORCL00010784

centralized "Infrastructure Repository". The DCM Service described earlier reads and updates information from the infrastructure repository and can also translate this information into the configuration files specific for individual components. Administrators can reconfigure one or more instances, or a cluster of instances, using Oracle Enterprise Manager that in turn reads and writes information from the Infrastructure Repository.

The new administration interface and infrastructure repository simplifies configuration and administration of the Oracle9*i*AS infrastructure and provides administrators with all the tools they need to really focus on monitoring key day to day usage and performance patterns.

Application Deployment and Administration 

Oracle9*i*AS Release 2 provides a brand new comprehensive HTML interface for easily administering Oracle9*i*AS' J2EE Container, J2EE Applications, and Web Services. The new interface supports:

- Configuring Oracle9*i*AS' J2EE Container through declarative property sheets, as an alternative to manually editing XML files
- Configuring application services and resources including web sites (IP and URL paths for web modules), JDBC data sources and security settings including users groups and roles.
- Deploying J2EE applications via a wizard that allows deployment o a single Oracle9*i*AS' J2EE container or to a Cluster; and Automates mapping URL paths, data sources, selecting a user manager and mapping security roles
- Browsing and Maintaining Applications including installing application updates or patches, and modifying application configuration
- Monitoring the Availability, usage and performance of all applications to determine request volume, longest response time; and additionally, for Servlets and JSPs load time, service time and overhead.

Oracle9*i*AS Release 2 and Oracle Enterprise Manager streamline and lower the cost associated with the management of Internet Systems Infrastructure and Internet Applications.

Security Services

Oracle9*i*AS Release 2 introduces a comprehensive integrated security framework supporting all Oracle9*i*AS facilities, as well as third party and custom applications deployed on Oracle9*i*AS. The framework is based on three specific elements -

- *Single Sign-on For Users* - With the rapid growth of the Internet, users need to remember many different passwords to access many different Internet Applications. This causes users inconvenience typically resulting in poor security practices (such as yellow post-it notes on pin-boards) and increased administrative costs. Single Sign-On solves this problem by ensuring that the user enters the user name and password only once to access all the authorized applications. From the user's perspective, authentication to each application happens transparently. This has four important benefits: (i) Greater Convenience - Users need to remember only one password; (ii) Increased security - Fewer user name and password combinations lower the risk of someone stealing them and gaining access to a user's restricted information. (iii) Ease of administration - Administrators also need to manage fewer passwords. (iv) Centralizing the authentication process also makes it possible to support additional authentication mechanisms in a localized manner. For example, you can implement an LDAP-based authentication, or digital certificate-based authentication, and the change would be localized to the SSO Server.
- *Centralized Security Administration for Administrators* - Oracle9*i*AS also lowers the cost and improves the convenience of security administration by centralizing provisioning of user accounts

ORCL00010785

and their access control privileges in an LDAP Directory Service. Administrators can easily and centrally create new user accounts, set up their access control privileges, and revoke user privileges.

- *Fine Grained Security for Applications* - As discussed earlier, Oracle9*i*AS offers a complete implementation of Java Authentication and Authorization Service (JAAS) to provide fine-grained security for J2EE Applications and Web Services.

Oracle9iAS Single Sign-On New

Oracle9*i*AS Single Sign-On (SSO) addresses the problem of 'too many passwords'. Oracle9*i*AS Single Sign-On resolves this problem by enabling authentication to all appropriate applications in an enterprise by entering a user name and password only once and by providing centralized administration of user name and password combinations for all users in an enterprise. With Oracle9*i*AS Single Sign-On, users typically sign on to a centrally administered SSO Server through a central Web portal. Once it authenticates a specific user, the SSO Server displays links to all the applications for that user. Let us first look at Oracle9*i*AS' Single Sign-On architecture and then examine how external applications can be made part of the single sign-on framework.

- *Single Sign-On Architecture* - The SSO Architecture includes the following key pieces: (i) Directory: Repository for all user related information. Includes identity, group membership, passwords, and other user-related information. Accessible through LDAP. New Oracle9*i*AS Release 2 fully supports Oracle Internet Directory (OID). By default, users and passwords are managed in OID. Users and their passwords are now provisioned for SSO via OID, using the OID Delegated Administration Services (DAS) Self Service Console (SSC). (ii) SSO Server: Service available on the network that authenticates users and stores their identity securely in a cookie. (iii) New mod_osso: Mod_OSSO is a new module for the Oracle HTTP Server, which allows the HTTP Server to become an SSO partner application. Applications running underneath the Oracle HTTP Server will be able to obtain a user's identity, as validated by SSO, from the HTTP Server. These applications can therefore enjoy the benefits of SSO without being re-written to be SSO partner applications. (iv) The Oracle9*i*AS J2EE Container, that accepts identities from HTTP Server for security. (v) New Third Party Authentication API - This API allows the SSO server to be configured to obtain validated user identity from a trusted third party authentication mechanism. It allows Oracle9*i*AS to support third party authentication products, such as Netegrity Siteminder(R), so that a user who is signed in via a third party product does not have to re-authenticate to access Oracle9*i*AS, or resources protected by Oracle9*i*AS SSO. (vi) New Public Key Infrastructure (PKI) Authentication - A Public Key Infrastructure (PKI) provides the means to bind public keys to their owners and helps in the distribution of reliable public keys in large heterogeneous networks. Public keys are bound to their owners by public key certificates. These certificates contain information such as the owner's name and the associated public key and are issued by a reliable Certification Authority (CA). Through PKI Authentication, user who has an X.509v3 certificate installed in their client will be able to authenticate to Oracle9*i*AS via SSL. In Oracle9*i*AS Release 2, the SSO Server can be configured to obtain a user's identity from the SSL module without requiring the user to submit an explicit SSO username/password for authentication.
- *Partner Applications i External Applications* - There are two kinds of applications to which Oracle9*i*AS Single Sign-On provides access: Partner Applications and External Applications (i) Partner applications are integrated with the SSO Server. They contain an Oracle9*i*AS Single Sign-On API that enables them to accept a user's identity as validated by the SSO Server. (ii) External applications are web-based applications that retain their authentication logic. They do not delegate authentication to the SSO Server and, as such, require a user name and password to provide access. Currently, these applications are limited to those that employ an HTML form for

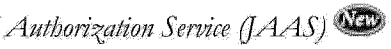
ORCL00010786

accepting the user name and password. The user name may be different from the SSO user name, and the SSO Server provides the necessary mapping.

The SSO offering in Oracle9*i*AS Release 2 is a critical differentiator for customers looking for a robust, fully integrated SSO architecture.

Centralized Administration of Security

Oracle9*i*AS lowers the cost and improves the convenience of security administration by centralizing provisioning of user accounts and access control privileges in an LDAP Directory Service. This is discussed in greater detail in the section on directory services below.

Java Authentication and Authorization Service (JAAS) 

Oracle9*i*AS Release 2 provides a complete JAAS implementation that provides a number of important capabilities:

- *Enforce Application Security* - The JAAS Provider serves as a standard User Manager plug-in to enforce security constraints for Servlets, JSPs, EJBs, and Web Services.
- *Authenticate, Access Control, Authorize Users* - JAAS provides facilities to authenticate users (identify users), authorize them with access control privileges (limit what they can do), delegate privileges (enabling code to run securely, with privileges of other users).
- *Fine-Grained Security Privilege Management* - JAAS provides principal and code-based policies to allow Java developers to specify security privileges for users in a fine-grained manner.
- *Run-As Privileges* - A J2EE Application may be configured to run with the permissions associated with the current client, or as a specified user (e.g. run as "DBAdmin"). This allows developers to enforce "least privilege" in their applications, allowing users only those privileges needed to perform a function, because users can only exercise privileges in the context of a well-formed business rule (e.g. an enterprise bean).
- *Single Sign-On and Centralized Security Administration* - JAAS is integrated with Oracle's Single Sign-on Server for single sign-on across any Oracle9*i*AS Application. It is also integrated with Oracle's LDAP Directory for centralized security administration.

As discussed in this section, Oracle9*i*AS Release 2 includes a comprehensive management infrastructure designed to streamline and lower the cost associated with the management of Internet systems infrastructure, applications, and users.

Directory Services

Oracle Internet Directory (OID) is a critical component of Oracle9*i*AS management and security infrastructure. It provides facilities to centrally manage user accounts, their access control privileges, and groups. Once users have been created centrally in OID, they can be shared across all components in Oracle9*i*AS. When users log in, they are authenticated once by Oracle9*i*AS Single Sign-On against their OID credential, and can thereby access multiple applications seamlessly. There are several important facilities that OID provides:

- *Key Directory Features* - OID provides the following key directory features: (i) Native LDAP v3 server supporting all LDAP2000-compliant RFCs, including LDAP v2 and v3 RFCs; (ii) Supports the X.500 information, naming, and storage model; (iii) Extensible directory schema for online modifications with no downtime; and (iv) LDAP developer APIs in Java, C, and PL/SQL to assist with application development.
- *Self-Service Console* - OID includes a Self-Service Console, an easy-to-use, web-based interface that allows end-users and application administrators to search for and manage data in the directory. Combined with the new Delegated Administration Service, this console provides Oracle9*i*AS

ORCL00010787

administrators with a means of provisioning end-users in the Oracle9*i*AS environment. OID also enables components of Oracle9*i*AS to synchronize data about user and group events, so that those components can update any user information stored in their local application instances.

- *Password Management* - OID offers sophisticated password policy management capabilities (e.g., control over expiration times and password length) and the ability to store passwords using a variety of hashing schemes. These features allow administrators to define consistent security policies across applications and easily share passwords with other systems.
- *Powerful Directory Search Capabilities* - OID provides users with very fast directory searches through sophisticated server-side caching capabilities. OID also provides two key features that ensure administrators can deliver seamless directory services to all users: (i) Alias De-referencing - When a user or an application searches on an alias, OID automatically de-references the alias and returns the entry to which it refers. This feature enables administrators to change the names of objects in ways that are transparent to users and applications. (ii) Enhanced Proxy Capabilities - Administrators can safely establish performant, auditable middle-tier application access to the directory "on behalf of" end user communities.
- *Synchronization with 3rd Party LDAP Servers* - The Oracle Directory Integration Platform enables customers to synchronize data between various directories and OID. The Oracle Directory Integration Platform is a set of services and an interface that makes it possible to develop synchronization solutions with third party metadirectories and other enterprise repositories, such as iPlanet. With Oracle9*i*AS Release 2, OID includes an agent for out-of-the-box synchronization with Oracle Human Resources and an agent for synchronizing information with selected third-party LDAP servers. OID also provides a plug-in framework for applications that require customized functionality, such as referential integrity of data. The plug-in framework is delivered with a programmable interface, allowing user-defined operations to be invoked by the directory server before or after LDAP commands.

To summarize, Oracle9*i*AS Release 2 includes a comprehensive management infrastructure designed to streamline and lower the cost associated with the management of Internet systems infrastructure, applications, and users.

PARTNER SOLUTIONS

In addition to the breadth of functionality provided by Oracle9*i*AS, Oracle9*i*AS is rapidly gaining momentum in the market place and there is a growing number of 3rd party partners providing a range of functionality that is integrated and certified with Oracle9*i*AS. Our partners deliver solutions in the areas of Content Management, Development Tools, Integration, Portal, Security, Systems Management, Web Services, and Wireless.

Content Management

The combination of Oracle9*i*AS and the powerful capabilities of industry-leading content management applications provide customers with a complete solution to create, manage and deploy high-performance, content-rich Web sites. Key Oracle9*i*AS content management partners include: Documentum, Fatwire, Interwoven, Vignette, Obtree and Serena.

Developer Tools

Developer tool partners have helped expand the variety of tools available for Oracle9*i*AS including rapid application development, application testing and debugging, memory leak detection and removal, performance and scalability bottleneck identification, component assembly, and application deployment. Key partners include

ORCL00010788

Borland Software, Computer Associates' Cool Joe, Macromedia, NetIQ, Neuvis, Rational Software, Sitraka, Sun Microsystems' Forte for Java, TogetherSoft, VMGear, and WebGain.

System Integrators

Oracle's key integration partners include: Answerthink, Aris Corporation, Canopy International, Cap Gemini Ernst *i* Young, KPMG Consulting, and Xoriant.

Enterprise Portal

Partners in the Oracle Portal Partner Initiative (OPPI) support a variety of functionality for the portal framework built into Oracle9*i*AS, including:

- **Developing custom Oracle9*i* portal solutions** . Key consulting partners include: Answerthink, Cap Gemini Ernst and Young, and Diversified Computer Consultants.
- **Providing news and information portlets.** Content aggregator partners include companies such as Factiva, Northern Light, and YellowBrix.
- **Developing new application modules to enhance portal functionality.** Key technology partners include: Autonomy for search, OnePage for portlet tool builders, Open Text for content management, SiteScape for discussion forums, and Verity for search and knowledge management.

Security

Together, Oracle9*i*AS and leading security partners such as Access360, Baltimore, Business Layers, Calendra, Check Point, Entrust, Netegrity, RSA, Siemens, and VeriSign, give customers an extra measure of confidence by delivering enhanced Internet security services that provide electronic certification, cookie verification, firewall privacy and other security management capabilities.

Systems Management

With the support of a broad range of industry-leading management products, IT administrators have expanded access to products they can use with Oracle9*i*AS for performance monitoring, load testing, infrastructure maintenance, upgrades and service level management. Key partners include: Applant, BMC Software, Computer Associates, Compuware, Precise, HP OpenView, Empirix, Mercury Interactive and Quest Software.

Web Services

With support from industry-leading Web services tools and framework partners, customers have increased flexibility in how they develop Web services for deployment on Oracle9*i*AS. Key Web services partners include: Borland Software, Bowstreet, Cacheon, Cape Clear Software, Infravio, SilverStream Software and Sonic Software.

Wireless

Oracle's mobile partners offer technologies such as personalization, locality, and content management that form a powerful infrastructure for delivering any type of content or application to any wired or wireless device. Key wireless technology and integration partners include: Cap Gemini Ernst *i* Young, Captaris, Compaq, Intrexa, KPMG, Kada Systems, MapInfo, MobileWay, Motorola, PWC, Siemens, SmartTrust, Symbol Technologies, Trintech, VoiceGenie, Wingcast, and Webraska.

SUMMARY

ORCL00010789

The Internet has played a vital role in transforming the way companies do business - enabling them to reach new customers and retain existing ones, implement new self-service business processes, streamline their supply chains - ultimately making them more efficient and lowering their costs. As companies have sought to make themselves more efficient by deploying their Enterprise Applications and Business Processes on the Internet, they face three fundamental systems challenges - (i) *Application Development Complexity* - The set of tools, standards, and technologies to develop business applications based on Internet technologies continues to evolve rapidly; (ii) *Application Deployment Complexity* - The infrastructure required to deploy and manage business applications and processes on the Internet needs to mature to meet mission-critical needs for performance, reliability, availability and security; (iii) *Middleware Cost and Complexity* - The number of pieces of application infrastructure, called "middleware," required to maintain new applications on the Internet and to web-enable existing applications continues to grow in both cost and complexity.

To resolve these challenges, Oracle recommends that Organizations design all Applications to Internet standards so that they can be deployed on corporate Intranets, Extranets or Internet. In doing so, Oracle recommends that organizations consolidate applications off desktop computers and multiple disparate pieces of middleware onto Application Servers - Applications will run more reliably at lower cost.

Oracle offers Oracle9*i* Application Server (Oracle9*i*AS) Release 2 - the second major release of its 100% standards-compliant Application Server to simplify how Internet Applications are developed; to make Internet Applications fast, reliable, manageable, and secure; and to substantially lower the Total Cost of Ownership associated with developing, deploying and operating Internet Applications. Oracle9*i*AS Release 2 has a number of new capabilities in three primary areas -

- *Most Productive to Develop Internet Applications* - Oracle9*i*AS Release 2 provides the most productive Application Server environment for developers to develop and deploy Internet Applications. Oracle9*i* Application Server is a 100% standards-based application server that provides a complete, and fully integrated platform for developing static and dynamic Web sites J2EE applications, and Web Services. Oracle9*i*AS provides support for open standards application development with full support for the Java2 Enterprise Edition platform, XML and emerging Web Services standards. You can simplify information access for your customers and trading partners by delivering enterprise portals they can easily customize and access from networked browsers or wireless devices. Through a comprehensive integration framework, modeling tools, pre-built adapters and Web Services, you can redefine your business processes, and integrate your applications and data with all of the key players in your virtual enterprise. You can deliver tailored 1:1 customer experiences via real-time personalization, assess and correlate web site traffic patterns using clickstream analysis and satisfy demands for up-to-the minute business information using Oracle9*i*AS integrated business intelligence services. Further, Oracle9*i*AS Release 2 is 100% Standards Compliant supporting standards such as J2EE 1.3, Web Services, SOAP, WSDL, UDDI, RosettaNet 1.1/2.0, ebXML, WebDAV, LDAP v3, SSL v3, and several XML standards.
- *Most Reliable to Deploy and Manage Internet Applications* - Oracle9*i*AS Release 2 provides the most reliable platform to deploy Internet Applications. This includes the industry's most optimized J2EE and Web Services environment; excellent scalability on single CPUs and multiple CPUs; several new high availability features including fault tolerance, load balancing, clustering, fast restart and distributed configuration management, online upgrade and maintenance; static and dynamic content caching to speed up the performance and scalability of Internet Applications; and integrated systems and security management for Internet applications and users.
- *Lowest Total Cost of Ownership* - Finally, Oracle9*i*AS Release 2 is a single integrated product and designed to lower the total cost of ownership to develop, deploy, and operate Applications. Oracle9*i*AS Release 2 is a single product, not a single brand with multiple poorly integrated products offering users lower software costs, lower hardware costs, lower integration costs, lower support costs; and lower operating costs.

ORCL00010790

By speeding up how you develop and deploy Internet Applications, by making these Applications run more reliably, and by integrating the infrastructure you need to run your Internet Applications; Oracle9*i*AS offers you the lowest total cost of ownership. Oracle9*i*AS is a complete, fully integrated e-Business platform that enables you to focus on your core competencies and innovate to achieve competitive advantage.

ORCL00010791



Oracle9*i* Application Server
September 2002

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

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com

Oracle Corporation provides the software
that powers the Internet.

Oracle is a registered trademark of Oracle Corporation. Various product and service names referenced herein may be trademarks of Oracle Corporation. All other product and service names mentioned may be trademarks of their respective owners.

Copyright © 2002 Oracle Corporation
All rights reserved.
