

Oracle Database 10g Release 2 on Windows: Development and Deployment

An Oracle Technical White Paper

May 2006

Oracle Database 10g Release 2 on Windows: Development and Deployment

Executive Overview	3
Oracle on Windows.....	4
.NET Application Development.....	4
Oracle Developer Tools for Visual Studio .NET.....	5
Oracle Explorer.....	6
Designers and Wizards.....	6
Automatic .NET Code Generation	6
PL/SQL Editor.....	6
Fully Integrated PL/SQL Debugger.....	7
Stored Procedure Testing	7
Oracle Data Window.....	7
SQL Query Window.....	8
Integrated Online Help	8
Oracle Database Extensions for .NET.....	8
Oracle Data Provider for .NET	9
XML Features.....	9
Performance	10
Native Oracle Types.....	11
Other Major Features.....	12
New Features for Oracle Database 10g Release 2.....	13
Oracle Services for Microsoft Transaction Server	16
COM and ODBC Application Development.....	17
Oracle Objects for OLE	18
ADO and the Oracle Provider for OLE DB.....	18
Oracle ODBC Driver	18
Application Deployment	18
Oracle Fail Safe.....	19
Logical/Physical Standby and Multi-cluster Manageability.....	20
Security.....	20
Oracle Wallets in Windows Registry.....	21
Microsoft Certificate Store Integration.....	21
Directory.....	22
Native Authentication and Active Directory.....	22
Oracle Net Naming with Active Directory.....	23
Conclusion	23

Oracle Database 10g Release 2 on Windows: Development and Deployment

EXECUTIVE OVERVIEW

Oracle Database 10g Release 2 provides a comprehensive database solution to the Windows platform to make developing and deploying Oracle on Windows and .NET easier, faster, and more cost-effective.

The Oracle Developer Tools for Visual Studio .NET is a powerful “add-in” for Microsoft Visual Studio 2005 and Visual Studio .NET 2003 that makes it easier and faster for .NET developers to write Oracle applications. It is tightly integrated with Visual Studio and includes powerful features such as designers, automatic .NET code generation and a fully integrated PL/SQL editor and debugger.

With Oracle Database 10g Release 2 on Windows, Oracle now offers Oracle Database Extensions for .NET. This database option lets you deploy stored procedures and functions written in a .NET-managed language into Oracle Database on Windows. A .NET-stored procedure can be called from within .NET code, from a SQL statement or trigger, from another .NET, PL/SQL or Java stored procedure, or from anywhere else a stored procedure or function call is allowed.

.NET application development has been greatly improved with the introduction of new features in Oracle Data Provider for .NET (ODP.NET). ODP.NET is a native .NET data provider that provides the best performance and most Oracle database functionality available from a .NET environment. In its most recent releases, ODP.NET now supports database change notification, RAC connection pooling properties, and ADO.NET 2.0.

Developers have the flexibility to choose from among multiple Windows data access methods to best fit their needs, including ODP.NET, Oracle Objects for OLE (OO4O), Oracle OLE DB, and Oracle ODBC. They are fully integrated with the Microsoft Visual Studio and Microsoft Office product suites. Distributed transaction support is provided using the Oracle Services for Microsoft Transaction Server (MTS).

Oracle Fail Safe provides high availability solutions for Oracle software on one or more Windows clusters with an easy to use manageability console. Oracle also integrates with Windows native security and Active Directory, allowing Oracle to operate in heterogeneous operating system and directory environments.

ORACLE ON WINDOWS

By being the first with a relational database on Windows NT in 1993, Oracle had made a commitment to offer the best information management solutions available on the platform. Since that time, Oracle technology has adapted to the latest changes in Windows computing starting from the early client/server solutions to the Internet applications, and now to grid technology. Today, Oracle remains a leader on the Windows platform through its world-class database and its close integration with the operating system's underlying technologies. Oracle has over twenty-five years of data management expertise and is the leading vendor with the technology, experience, and services needed to deploy business-critical applications on Windows—whether on a corporate intranet or the Internet.

Oracle Database 10g Release 2 for Windows provides all the features needed for data management, whether it is used for a departmental or company-wide deployment. It allows users to take advantage of the cost-effectiveness and ease of use of Windows, while providing the scalability, reliability, and performance traditionally available from Oracle. This new Oracle database contains additional enhancements for application integration on the Windows platform.

Two areas where Oracle has focused its attention are in application development and deployment on Windows. Oracle Database 10g Release 2 has made significant strides to improving performance and the ease of use and manageability for both Windows developers and integrators.

One of the great benefits of using Oracle products is its support of multiple programming standards. By supporting Java, .NET, COM, and Win32 applications, Oracle ensures all developers can use Oracle's advanced database features, providing true flexibility for development organizations. Each of Oracle's data access drivers is designed to maximize performance and to have access to the latest database features.

.NET APPLICATION DEVELOPMENT

.NET developers writing code for Oracle applications must typically perform various database tasks, such as modifying table or view design, changing data in database tables, and editing and debugging PL/SQL stored procedures. The Oracle Developer Tools for Visual Studio .NET (ODT) makes these tasks easy. ODT is a tightly integrated “add-in” for Microsoft Visual Studio 2005 and Visual Studio .NET 2003 that includes powerful features, such as designers and wizards, automatic .NET code generation, and a fully integrated PL/SQL editor and debugger.

Oracle Database Extensions for .NET is new in Oracle Database 10g Release 2 on Windows. This database option lets you deploy stored procedures and functions written in a .NET-managed language, such as C# or VB.NET, into Oracle Database on Windows. A .NET stored procedure can be called from within .NET code, from a SQL statement or trigger, from another .NET, PL/SQL or Java

Oracle provides a number of products for .NET programming. Oracle Developer Tools for Visual Studio .NET is a plug-in for use at design time. Oracle Database Extensions provides support for .NET stored procedures. Oracle Data Provider for .NET enables data access to all of Oracle's key database features.

stored procedure, or from anywhere else a stored procedure or function call is allowed.

Oracle offers data access support with ODP.NET. ODP.NET is a native .NET data access provider with direct access to Oracle. It can be used from any .NET language, including C# .NET, Visual Basic .NET, and ASP.NET. ODP.NET offers the best performance and access to more Oracle database features than any other data provider. ODP.NET was designed specifically to allow .NET developers to maximize the Oracle database's capabilities.

ODP.NET 10.2 introduces many new features that both use new functionality available in Oracle Database 10g Release 2 as well as enhance existing functionality in the Oracle database. As such, developers will benefit using the latest ODP.NET version whether they are deploying new database applications or enhancing legacy applications.

Oracle Developer Tools for Visual Studio .NET

The Oracle Developer Tools for Visual Studio .NET makes it easier and faster for .NET developers to write Oracle applications. It is available as a free download at <http://otn.oracle.com/dotnet>.

The Oracle Developer Tools for Visual Studio .NET is a powerful new “add-in” for Visual Studio .NET that makes it easier and faster for .NET developers to write Oracle applications. Available now as a free download on the Oracle Technology Network, it is tightly integrated with Visual Studio 2005 and Visual Studio .NET 2003 and includes powerful features such as designers, automatic .NET code generation and a PL/SQL editor/debugger.

All of these features are designed with the goal of making developing for Oracle on Windows as intuitive and as easy as possible. Developers coming from a SQL Server background will find it to be familiar and intuitive, minimizing any learning curve. And for those who are new to Oracle development, this is an easy way to learn.

Major features include:

- Oracle Explorer – Browse and alter the Oracle schema via a tree control
- Designers and Wizards– e.g. Table Designer - makes database tasks easy
- Automatic Code Generation – Drag and drop to create working code.
- PL/SQL Editor and debugger – Edit and debug PL/SQL stored procedures, functions, packages and triggers
- Stored Procedure Testing – Run stored procedures and functions
- Oracle Data Window – View and edit your Oracle data
- SQL Query Window – Execute any ad-hoc SQL statement or script
- Integrated Help System - SQL, PL/SQL and Error Reference Manuals
- .NET Deployment Wizard - Deploy .NET stored Procedures easily

Oracle Explorer

The Oracle Explorer is a tree control that allows you to view the structure of the Oracle schema. All of the various Oracle schema types such as tables or stored procedures are included here. Metadata such as column datatypes or stored procedure parameter types are displayed in the Visual Studio .NET “properties pane” when a schema object is selected. A context menu for each schema object offers additional features: SQL Scripts can be generated for schema objects, filters are provided to limit what is shown, and designers and wizards can be spawned to alter the schema. For example, while viewing the structure of a table, the context menu can be used to spawn the “Oracle TableDesigner” to modify the design of the table.

Designers and Wizards

The Oracle Developer Tools offer a variety of designers and wizards to provide step by step assistance to create or alter database objects such as tables, views, stored procedures, stored functions, PL/SQL packages, sequences, indexes, constraints, triggers, synonyms, and more. For example, the table designer makes looking up datatypes or memorizing SQL syntax a thing of the past. In the case of the table designer, the user simply provides the table name and the names of the columns and chooses the appropriate datatype from a drop down box. Pressing the “save” button would then cause the table to be created or altered. A newly created schema object will then immediately appear in the Oracle explorer.

Any SQL that is generated by a designer to create or modify a schema object is displayed in both a “Preview” window (before a change is committed) as well as in the Oracle Database Output window (after the change is committed).

Automatic .NET Code Generation

Dragging and dropping a schema object from the Oracle Explorer onto an application’s form results in .NET code being automatically generated for SELECT, UPDATE, INSERT, and DELETE operations on that schema object.

Dragging and dropping a schema object from the Oracle Explorer onto an application’s form results in .NET code being automatically generated for SELECT, UPDATE, INSERT, and DELETE operations on that schema object. The code utilizes the OracleDataAdapter class which is provided by the Oracle Data Provider for .NET, Oracle’s robust .NET data provider. The DataAdapter can then be used to connect UI elements (such as a DataGrid) on the application’s form to the Oracle database, with a minimum of additional coding required. This use of a DataAdapter is standard and will be very familiar to anyone with programming experience with other databases.

The user is also given the option to generate code for a typed or untyped dataset based on an Oracle table or view. These datasets can then be bound as a data source to UI elements or can be used in other standard ways throughout Visual Studio .NET, such as with the Visual Studio XML Schema designer.

PL/SQL Editor

The PL/SQL editor provides a tightly integrated development environment for Oracle developers while freeing them from having to leave Visual Studio .NET

when stored procedure development is required. Standard Visual Studio features include syntax coloring to improve readability and collapsible regions to hide stored procedures or functions that are part of a very large and complex package - this makes it easier for attention to be focused on the procedure or function that is being worked on. Additionally, as SQL or PL/SQL statements are entered, a drop down list of tables or columns will pop up to provide autocompletion of the statement.

When the PL/SQL is compiled and errors are found, they are listed in the Visual Studio .NET task list. Clicking on them will take the developer to the line of code that failed. Pressing the context sensitive help key will open the error messages manual to the error code in question.

Fully Integrated PL/SQL Debugger

Now with ODT's integrated PL/SQL debugger, you can step through the PL/SQL code, view and modify variable values, look at the call stack, set break points, and more. You can step from your VB.NET or C# application code directly into PL/SQL code and back out again.

A new feature introduced in ODT 10.2.0.2 is a PL/SQL debugger that is fully integrated with Visual Studio. You can now debug the PL/SQL code inside of stored procedures, functions, or triggers in exactly the same way that you debug VB.NET or C# code. You can step through the PL/SQL code, view and modify variable values, look at the call stack, set break points, and more.

You can also step from your VB.NET or C# application code directly into PL/SQL code and back out again. For example, if your C# code is calling a PL/SQL procedure or function using the Oracle Data Provider for .NET, you can debug the C# code, and then step into the PL/SQL code to continue debugging. You can examine all of the parameter values passed into the stored procedure or function, including complex data types and arrays from within the PL/SQL debugger. When the PL/SQL code execution has completed, you are then returned to the C# code to continue debugging.

You can also set breakpoints and debug PL/SQL stored procedures, functions, or triggers called by any application located on any platform without requiring any modification or rebuilding of that application. For example, a PHP application running on Linux calls PL/SQL stored procedure FOO and passes in an array of values. After you set an environment variable in the environment of the PHP application, you can set a breakpoint in FOO within Visual Studio and begin debugging when FOO executes.

Stored Procedure Testing

Stored procedures and function may be tested quickly via the "Run" context menu on the Oracle Explorer. Input parameters are requested, and then the procedure is executed. Output parameters are provided in an easily readable format. Complex output types, such as REF CURSOR yield a link that when clicked will open up a grid containing the complex data.

Oracle Data Window

The Oracle Data Window displays table or view data in an easy to read grid and allows the developer to insert, update or delete data without having to leave Visual

Studio .NET development environment. For large tables, the user can provide a row number to jump ahead to a particular row.

SQL Query Window

Ad hoc SQL statements, such as those containing SELECT, CREATE, ALTER, etc may be executed from the SQL Query Window. If the output of the SQL statement is a table or view, the user is given a choice of text or grid output. Multiple statements may be highlighted and run as a group. The script operator “@” can be used to run SQL scripts.

Schema objects may also be dragged and dropped from the Oracle Explorer onto the SQL Query Window surface to automatically generate the appropriate SQL.

Integrated Online Help

The Oracle Developer Tools for Visual Studio .NET includes key Oracle documentation that has been converted into Visual Studio help format. The SQL Language Reference Guide, the PL/SQL User and Reference Guide, and the Error Messages and Codes manuals have each been included.

The chapters in these manuals may be read sequentially via the Visual Studio .NET help pane. In addition, this documentation has been integrated with the Visual Studio context sensitive help. For example, while writing a stored procedure, the developer can highlight a SQL or PL/SQL keyword, such as “SELECT”, press the context help key, and be taken automatically to the appropriate page in the SQL Language Reference Guide. In addition to these manuals, the Oracle Developer Tools documentation contains helpful getting started and “walkthrough” chapters as well as a reference guide.

Oracle Database Extensions for .NET

The Oracle Database Extensions for .NET is a new feature of Oracle Database 10g Release 2 on Windows that makes it easy to develop, deploy, and run stored procedures and functions written in a .NET managed language such as C# or VB.NET.

.NET Stored procedures or functions are developed using Visual Studio and then deployed using the tightly integrated .NET Deployment Wizard which is a new feature of the Oracle Developer Tools for Visual Studio .NET. The deployment wizard loads the newly created .NET assembly into Oracle and registers the procedure or function with the database.

Stored procedures that require data access can use the Oracle Data Provider for .NET. The .NET Framework 2.0 and ADO.NET 2.0 are both support in .NET stored procedures. The code used for the procedure or function is almost exactly the same as code written for a client side application, with the exception of an optional “context connection” in which the procedure uses the same connection as the caller. Code can also be written that can run both on the client side and in a stored procedure with no changes required.

The Oracle Database Extensions for .NET is a new feature of Oracle Database 10g Release 2 on Windows that makes it easy to develop, deploy, and run stored procedures and functions written in a .NET managed language such as C# or VB.NET.

After deploying, a .NET stored procedure can be called from within .NET code, from a SQL statement or trigger, from another .NET, PL/SQL or Java stored procedure, or from anywhere else a stored procedure or function call is allowed.

Oracle Data Provider for .NET

The Oracle Data Provider for .NET provides fast data access performance to Oracle databases. It supports the latest database functionality, including XML DB and native Oracle data types. New Features include database change notification, RAC connection pooling, and ADO.NET 2.0.

ODP.NET is a native .NET data access driver for the Oracle database. Because it was developed for Oracle-specific data sources in mind, ODP.NET is optimized for fast Oracle performance and to access the latest database functionality. As such, ODP.NET makes using Oracle databases more flexible, faster, and more stable than many other data access solutions for Oracle. Beginning with version 10.2.0.2, ODP.NET is compliant with both ADO.NET 1.x and 2.0.

Some data providers, such as OLE DB .NET and ODBC .NET are designed for maximum portability in accessing different data sources. What is lost when using these providers is the ability to take advantage of the database's inherent performance and data management features. For example, one of ODP.NET's unique features is support for native Oracle data types, such as REF Cursors and LOBs. These data types provide greater flexibility for data retrieval than their .NET data type counterparts. Developers can choose the data types that best fit their data usage and performance requirements. For example, REF Cursors will defer result set retrieval until the data is actually accessed, a key performance optimization often needed. With LOB data types, an application can choose to retrieve all LOB data in a result set in one database round trip, just a set data chunk per read, or defer LOB retrieval until the LOB data is read. ODP.NET provides greater control to performance tune and take advantage of these database features.

With the advent of ADO.NET 2.0, even code portability is no longer an issue. Developers can create a single set of code that can access multiple vendors' databases much more easily now with the ADO.NET data provider factory classes.

In Oracle Database 10g Release 2 and the subsequent ODP.NET 10.2.0.2 release, Oracle introduces new RAC connection pooling features, database change notification, and ADO.NET 2.0 support.

XML Features

ODP.NET provides support for Oracle XML DB and interoperating with System.XML. ODP.NET can work with schema and non-schema-based XML with support for native Oracle XML data types using OracleXMLType and OracleXMLStream.

With XML becoming a popular language for data integration and web services, many .NET programmers are using it in their applications. XML is key part of both the Oracle database and .NET Framework. ODP.NET allows developers to exploit these two technologies: Oracle XML DB and .NET System.XML services.

XML DB is Oracle's high-performance, native XML storage and retrieval technology available within the database server. It provides a unique ability to store and manage both structured and unstructured data under a standard W3C XML data model. XML DB provides complete transparency and interchangeability between the XML and SQL metaphors. ODP.NET makes available all of XML DB's functionality to .NET clients, allowing developers to share and make changes to XML between the database and .NET. This support extends to both schema

and non-schema-based XML to provide flexibility for differing application requirements. Moreover, ODP.NET contains two native XML data types, OracleXMLType and OracleXMLStream, for easier XML data management. XML can be shared easily between XML DB and Microsoft's System.XML services.

System.XML is a set of interfaces for manipulating XML data sets from .NET data providers. ODP.NET interoperates with the System.XML programming interfaces, feeding data via the ODP.NET DataAdapter interfaces. One of the main differences between XML DB and System.XML is that the former provides XML services where the data resides, on the database server; the latter manipulates XML on the client side. As such, ODP.NET provides greater choice for programmers to pick the XML technology that best fits their project requirements.

Starting with version 9.2.0.4, ODP.NET makes using XML easier with its integration of XML capabilities with Microsoft's XML framework and Oracle XML DB. Relational and object-relational data can be accessed as XML from an Oracle database instance into a Microsoft .NET environment. XML changes can be made and saved back to the server as XML data. In ODP.NET 10g and higher, support for schema-based OracleXMLType is now included in addition to non-schema-based XML.

Performance

One of ODP.NET's key differentiators over other providers is its out-of-the-box performance and numerous tuning options. Under the covers, a lot of optimizations have been done to ensure fast .NET access to Oracle data sources. Additionally, ODP.NET has many tunable parameters for specific data retrieval and data change scenarios. Many of these optimizations were developed for retrieving and manipulating Oracle native types, such as LOBs and REF Cursors.

One of the most widely used performance tuning options is connection pooling, which is critical for applications with large numbers of users that connect and disconnect from the database. ODP.NET creates a pool of connections with tunable settings that include connection lifetime and timeout, minimum and maximum pool size, and the number of connections to increment or decrement from the pool at one time. These parameters give developers greater control over how their application handles large user populations and the changes in those populations over time. This ultimately leads to better application response time and quality of service for end users.

For data retrieval, ODP.NET can specify a set amount of data to return for each database round trip. Many times, a developer may not need to retrieve all the data queried at once because the end user is consuming the data over a period of time. The data fetches for the query can be spaced in distinct chunks defined by the developer through two ODP.NET OracleCommand properties: FetchSize and RowSize. FetchSize tells ODP.NET how much data to retrieve per database roundtrip. RowSize indicates how large each row of data is. RowSize is a read-only property that gets set after a query is executed. If the developer wishes to

ODP.NET includes numerous performance tuning features to optimize data retrieval and data changes. Some of these tuning options include using connection pooling; intelligently setting a result fetch size to return per database round trip; using LOB data types; and employing PL/SQL associative arrays.

fetch ten rows of data per database roundtrip, all that is required is to set FetchSize equal to ten multiplied by RowSize. The wonderful thing about RowSize is that its value is determined at run-time. So if there is a schema change or a query change in the future, there is no need to modify the code to ensure ten rows of data are fetched per round trip.

A similar feature exists with LONG and LOB data types. For applications using these data types, performance is often a key issue because LOB and LONG data can be quite large. The chunk size of these types can be set to specify the amount of data to retrieve from each specific LONG or LOB per database round trip. These tuning options provide .NET developers the flexibility to tune their data retrieval for optimum large object retrieval.

LOB data types have additional performance capabilities. By default, executed queries defer the actual LOB data fetch until the user attempts to read the data. When retrieving many sets of large LOBs, this behavior is generally optimal to prevent overloading the network with LOB data being delivered to the client. However, for small LOBs, this behavior can slow be with more database round trips than is necessary. ODP.NET has introduced the InitialLOBFetchSize property in the 9.2.0.4 release on the OracleCommand and OracleDataReader classes to optimize small LOB retrieval performance. If InitialLOBFetchSize is set to a value greater than zero, the initial LOB data from all the LOBs queried is fetched in one round trip up to the number of characters or bytes that is specified in this property. For instance, if InitialLOBFetchSize were set to 10 KB, then the first 10 KB of all the LOBs selected would be retrieved to the client in one database round trip. This can significantly speed up applications using many small LOBs.

In ODP.NET 9.2.0.4 release and above, Oracle introduced the ability to pass arrays between the database and .NET. This made sharing large data sets between the database and client much easier. ODP.NET uses PL/SQL associative arrays in the database to pass data to .NET arrays. Associative arrays complement two other methods for passing large data sets between Oracle and .NET: anonymous PL/SQL and parameter array binding. These capabilities provide developers a number of choices to performance tune data management and reduce the amount of code they need to write for this task.

Native Oracle Types

Microsoft has introduced a set of unified data types among the different .NET programming languages. With ODP.NET, Oracle users have access to .NET data types as well as Oracle native types. Oracle types can be fully manipulated within a .NET application and interoperate with .NET data types. Oracle native types provide advanced functionality to store and manipulate data structures from the database, such as XML or Microsoft Word documents. Even with common types, such as OracleDecimal, the equivalent to the .NET decimal type, Oracle types provide additional functionality if needed. In the example of OracleDecimal, this

ODP.NET supports native Oracle database types, such as REF Cursors, within the .NET environment. Other major features include being able to use any type of PL/SQL; local and distributed transactions; and internationalized applications with Unicode.

data type provides a higher level of precision of 38 digits than the .NET decimal with a 28 digit precision.

ODP.NET supports the gamut of advanced Oracle types within the .NET environment, including REF Cursors, XMLType, LOBs (CLOBs, BLOBs, NCLOBs), BFILEs, LONGs, RAWs, LONG RAWs, and N-data types. One of the limitations of using other providers is that users may be limited in data type functionality. For example, in ODP.NET, multiple result sets returned from a stored procedure as REF Cursor output parameters can be accessed in an arbitrary way. One can retrieve the results of the second REF Cursor without retrieving the results of the first. With other .NET providers, data may have to be accessed in a linear manner. The first result set's data must be retrieved prior to accessing the second's. This would negatively affect performance.

Other Major Features

ODP.NET exposes many other Oracle database features, including PL/SQL, transactions, and Unicode support. ODP.NET users can fully execute PL/SQL stored procedures and functions in the database. PL/SQL can be packaged or non-packaged, or even exist as anonymous PL/SQL within .NET. Anonymous PL/SQL is generally employed to batch a set of SQL statements and execute the statements in one database round trip. Batching statements is a useful performance technique.

ODP.NET can participate in transactional applications with the Oracle database as the resource manager. ODP.NET employs Microsoft Enterprise Services as the transaction coordinator in a .NET environment. The Oracle Services for Microsoft Transaction Server (OraMTS) acts as a proxy among ODP.NET, Enterprise Services, and the Oracle database in order to coordinate these transactions. OraMTS provides a robust architecture for ODP.NET programmers to have their transactional applications maintain high availability and high scalability. In ADO.NET 2.0, ODP.NET supports distributed transactions via the System.Transactions namespace.

ODP.NET has full Unicode support so that .NET users can globalize their applications easily in multiple written languages. This globalization support allows developers to develop one set of code for multiple culture/language settings. ODP.NET globalization extracts the client computer's national language setting to display information in a locale-specific format. For example, a browser set to Japanese will have currency displayed in yen. Without additional coding, that same application can be deployed in Germany to show deutsche marks. This makes deploying applications in a multiple locales easier and faster since no additional coding is necessary.

New Features for Oracle Database 10g Release 2

Database Change Notification

A constant challenge for client side caches is keeping the data in synch with server data changes. Using database change notification, ODP.NET clients are alerted when data on the server is modified, allowing the client to refresh its data cache.

Database change notification, a feature of Oracle Database 10g Release 2, enables client applications to receive notifications when DML or DDL changes are made to a database object of interest. This feature can be used in both the .NET Framework 1.x and 2.0 releases.

To use database change notification, the application registers a query with the database. When a query has dependencies on underlying database objects and a change to an object is committed, the database publishes a change notification to the application. The notification only contains metadata about what data or objects changed; it does not contain the changed data. You can create a client event handler to reissue the registered query to obtain the changed data.

Database change notification is particularly useful to applications that use cached results. Caching is particularly effective at improving the application scalability by allowing rapid access to data without making expensive roundtrips to the database. But this scalability comes at the price of data becoming stale.

Database change notification solves the problem of stale data in a cache. Although database change notification is similar to a trigger in that both respond to a particular event, a trigger takes action immediately, whereas a database notification is just an alert, not an action. It is up to the application to determine what action, if any, to undertake and when. The application can immediately refresh the stale objects, postpone the refresh, or ignore the notification.

Web applications often handle a variety of data such that not all of it needs to be queried in real time. For example, a weather forecast is only updated periodically. End users don't need to query the database every time the web page is visited. Since many people will be requesting the same data, application performance and scalability are greatly enhanced by caching the results and retrieving the data from the cache. At some point, however, the weather forecast is updated and the cache must be refreshed. This may be done when the current weather forecast has significantly diverged from the cached forecast.

The process of executing queries to refresh the cache after receiving a change notification requires the database user to have the CHANGE NOTIFICATION privilege. Connecting as this user, applications subscribe to change notifications that are published when an event of interest occurs.

The change notification developer creates a client-side notification handler for the registered queries. The following ODP.NET classes are used when building change notification applications:

- OracleDependency – Used to create a dependency between an application and an Oracle database. It enables the application to receive a notification of a data change (for example, UPDATE statement) or schema change (for example, DROP TABLE). The OnChange event

handler in this class provides the client logic for what to do after the notification is received.

- OracleNotificationEventArgs – Provides the event data for a change notification.
- OracleNotificationRequest – Used to specify the characteristics of a notification request and its notification.

Oracle's database change notification has a number of features not available from SQL Server. Oracle supports all types of joins, whereas SQL Server does not support queries containing either outer joins or self-joins. SQL Server does not support notifications for statements using views, whereas Oracle database change notification supports views with the exceptions of fixed views (for example, V\$ tables) and materialized views. SQL Server notifications also require explicit column references, while Oracle Database notifications support both SELECT * and explicit column references.

SQL Server notifications are not persistent. When the SQL Server notification is published, the notification handler is removed. If the notification handler is still needed, a new notification handler must be created by the application. Oracle Database change notifications can persist after repeated changes when you set OracleNotificationRequest.IsNotifiedOnce to true.

Real Application Cluster (RAC) Support

RAC is a cluster database with a shared cache architecture that overcomes the limitations of traditional shared-nothing and shared-disk approaches to provide database solutions for any business application. The cluster database is hosted on multiple computing nodes permitting better scalability and higher availability than a single computer server by itself. Because no special hardware is needed, RAC clusters can be built with commodity hardware.

ODP.NET has always supported data access for RAC clusters transparently. To improve connection management based on real-time workload information, Oracle introduced two new connection pooling properties for the 10.2 release of RAC and ODP.NET. The first feature improves load balancing work across RAC instances, especially after nodes are added or removed from a cluster. The second feature automatically removes severed RAC connections from the connection pool.

With RAC, how ODP.NET connections are allocated is based on the database's load balancing advisory and service goal. Load balancing distributes work across all of the available RAC database instances.

In general, connections are created infrequently and exist for a long duration. Work comes into the system with high frequency, borrows these connections, and exists for a relatively short duration. The load balancing advisory provides advice about how to direct incoming work to the instances to provide the optimal quality

ODP.NET introduces two new connection pooling features for RAC databases: ODP.NET connections are automatically load balanced across existing nodes based on real-time cluster metrics and severed database connections are automatically removed from the connection pool.

of service for that work. This minimizes the need to relocate the work later to a different instance and ensures existing jobs can complete quickly.

The metric by which this work is distributed across instances is determined by the service goal. The database administrator sets the service goal for either service time or throughput. The service time metric is based on how quickly the database can complete tasks, essentially its response time. Service time is most useful for database applications where the load is constantly changing. The throughput metric gauges the amount of processor resources available to take on additional tasks. When a new operation comes in, it can be directed to the node with most processor time available. The throughput metric can be most useful when executing batch processes.

By using these service goals, feedback is built into the system. Work is routed to provide the best service times globally, and routing responds gracefully to changing system conditions. When a RAC node is added or removed from the cluster, this load balancing allows connections to more quickly distribute themselves evenly across all the nodes to account for the system change. End users will then experience less disruption or slowed service. In a steady state, the system approaches equilibrium with improved throughput across all of the RAC instances.

To use this feature in ODP.NET, the .NET application must use connection pooling and set the connection pool parameter, `Load Balancing`, to true.

The second feature added for RAC connection pooling allows ODP.NET to automatically free resources associated with severed connections that were caused by a downed RAC service, service member, or node. Without this feature, if a RAC node failed, the connection pool would retain connection resources that were no longer valid. End users may end up using these severed connections from the pool. Without a way to identify these connections, administrators would have to reset all the ODP.NET connections in the pool every time some part of the RAC cluster failed.

In 10.2 RAC and ODP.NET, these severed connections are now automatically cleaned up without requiring any administrator intervention at run time. To enable this feature in ODP.NET, use connection pooling and set the connection pool parameter, `HA Events`, to true.

ADO.NET 2.0

Beginning with ODP.NET 10.2.0.2, ODP.NET provides support for ADO.NET 2.0. ADO.NET 2.0 introduces a new level of abstraction for the Data Access Layer (DAL). Instead of using provider-specific classes that implement a generic interface, ADO.NET 2.0 offers the `DbCommon` classes, which inherit from the `System.Data.Common` namespace and allow developers to use factory classes. Database provider factory classes allow an easier way to create one set of generic data access code, as opposed to data source-specific access code, to any database supporting ADO.NET 2.0.

ODP.NET supports ADO.NET 2.0's new features, including provider factories, connection string builder, schema discovery APIs, and DataAdapter batch updates.

The OracleClientFactory class supports the creation of all of the classes in System.Data.Common. Because these concrete classes inherit from the DbCommon abstract base classes, you can write generic DAL code by using DbCommon base class object names. There are some areas of the DAL that remain data source-specific, including the connection string, SQL, and stored procedure calls.

Connection strings in ADO.NET are name-value pairs. The DbConnectionStringBuilder class has a dictionary that maps generic parameter names to provider specific parameter names. The OracleConnectionStringBuilder class inherits from and extends the generic DbConnectionStringBuilder class to expose Oracle-specific connection string properties. Programmers can use OracleConnectionStringBuilder dynamically to set connection string parameters at run-time and/or obtain connection string parameters from the app.config file.

The ADO.NET 2.0 schema discovery APIs provide a generic way to retrieve metadata from any data source. Developers can obtain Oracle metadata by using an OracleConnection.GetSchema method call. There are five types of common metadata that can be exposed: MetaDataCollections, Restrictions, DataSourceInformation, DataTypes, and ReservedWords. In addition, there is additional ODP.NET-specific data source information that can be retrieved.

When working with large DataSets, developers will want to minimize the number of database round trips required to perform updates. By using the OracleDataAdapter.UpdateBatchSize property, programmers can specify the number of rows to update per roundtrip. To update all the modified rows in one roundtrip, UpdateBatchSize can be set to zero.

Oracle Services for Microsoft Transaction Server

Many Oracle customers on Windows platforms use Microsoft's transaction server in the middle-tier. The transaction server is an application server for objects and transactions in distributed environments. At its core is the Distributed Transaction Coordinator (DTC), which coordinates transactions between distributed resource managers. In Windows 2000, the MTS executive is replaced by enhancements to COM+ services. With the introduction of .NET, it is now part of .NET Enterprise Services and System.Transactions.

The Oracle Services for Microsoft Transaction Server (MTS) integrate the DTC with Oracle databases. The services, which act as a proxy, allow customers to use Oracle databases as resource managers in DTC-coordinated transactions. Specifically, the services provide the following transaction operations:

- Enlistment - Context maintenance for global transactions
- Completion - Translation of two-phase commit calls between DTC and Oracle
- Recovery - Resolution of in-doubt DTC transactions

With Oracle Services for MTS, Oracle clients can participate in distributed transactions with Microsoft Distributed Transaction Coordinator. In ADO.NET 2.0, ODP.NET supports using System.Transactions in distributed transactions.

- Connection pooling - Caching of transactional database connection resources

The Oracle Services for MTS features an architecture designed for performance, scalability, and availability. Each MTS server maintains its own cache of resource manager proxies, so that any database may have numerous proxies associated with it. This cache of proxies is maintained by the Oracle Services for MTS integration layer, which runs directly inside the MTS server process. This integration layer handles transaction enlistment and completion among the application, DTC, and database.

In a multiple MTS deployment, this model leads to better availability because no proxy can become a single point of failure. Moreover, it leads to better overall performance as no single proxy can become a bottleneck to the database. Having the Oracle Services for MTS integration layer within the MTS process reduces extra-process communication, which is much slower than making intra-process calls.

Each MTS server runs a recovery daemon. Recovery is driven by the database, which requests the final outcome of in-doubt transactions through reenlistment. A PL/SQL package runs a periodic recovery job, which contacts the recovery daemon associated with the dubious transaction. The daemon requests the transaction result from DTC and informs the database of the result. A database administration account, named MTSSYS, is created upon installation for DTC recovery.

The Oracle Services for MTS allow developing applications with a variety of data access interfaces, including ODP.NET, OO4O, OCI, ADO/OLE DB, and ODBC. In general, ODP.NET, OO4O, and OCI provide better performance and compatibility with Oracle. ODP.NET, OO4O, and OCI use optimized connection pooling provided by Oracle's resource dispensers. The Oracle Provider for OLE DB and the Oracle ODBC driver employ Microsoft resource management for database connections.

Although the Oracle Services for MTS itself only runs on Windows, it can coordinate transactions with Oracle databases on any operating system. This makes it possible for customers using a Windows middle-tier to access a UNIX-based Oracle database, for example.

COM AND ODBC APPLICATION DEVELOPMENT

COM programming remains popular among Windows developers, especially when enhancing older applications. Oracle provides COM developers two programming interfaces for data access: Oracle Objects for OLE and Oracle Provider for OLE DB. More traditional ODBC data access is provided through the Oracle ODBC Driver. OLE DB and ODBC are often used through an automation layer, Microsoft ActiveX Data Objects (ADO).

Because it is an Oracle native driver, OO4O does not incur the overhead of ODBC and OLE DB drivers. OO4O provides very fast performance and exposes the latest Oracle server features.

Oracle Objects for OLE

Oracle Objects for OLE (OO4O) is a COM-based database connectivity tool that combines optimized access to Oracle databases with easy to use interfaces. OO4O can be employed in a variety of environments, ranging from web applications to client/server applications. It is accessible from virtually any programming or scripting language that supports COM, such as Visual Basic, Visual C++, Microsoft Office, Active Server Pages, and MTS/COM+.

Because it is an Oracle native driver, OO4O provides fast performance on Windows clients to Oracle databases—it does not incur the overhead of ODBC and OLE DB drivers. OO4O has been developed and has evolved specifically for use with Oracle database servers. It provides easy access to advanced Oracle features that are otherwise cumbersome or inaccessible to use from ODBC and OLE DB-based components, such as ActiveX Data Objects (ADO).

ADO and the Oracle Provider for OLE DB

The Oracle Provider for OLE DB gives ADO developers high performance and efficient access to Oracle databases. OLE DB is a Microsoft data access specification that utilizes a set of COM interfaces for accessing and manipulating data from various data sources.

The Oracle OLE DB provider is an optimized driver for Oracle databases, providing high performance and access to Oracle-specific database features, such as support for LOBs, PL/SQL, and REF Cursors. All these Oracle features and data access optimizations are generally not available in other third-party OLE DB providers for Oracle and the OLE DB-ODBC Bridge. By adhering closely to the Microsoft OLE DB specification, the Oracle provider allows ADO developers to take full advantage of OLE DB without having to sacrifice using advanced Oracle functionality.

Oracle ODBC Driver

The Oracle ODBC driver permits a more traditional way Windows applications can access the Oracle database server. Oracle ODBC is Version 3.51, Level 2 compatible. It provides a high performance method for accessing Oracle.

APPLICATION DEPLOYMENT

Deploying applications on Windows requires close integration with native Windows services and middle-tier servers to ensure proper application interoperability. Oracle Database 10g Release 2 provides this tight integration, allowing organizations to take advantage of core operating system and server functionality with Oracle's advanced feature set. Oracle's integration ensures transparent interoperability that is scalable, available, and secure. In this way, organizations can focus resources on building their application business logic, rather than correcting product incompatibilities or inefficiencies.

Oracle Database 10g Release 2 includes many sets of tools for ensuring optimized application deployment environments. When deploying mission-critical Oracle

The Oracle OLE DB provider is built to handle the high volume and high loads of the Internet application environment. It is generally used through ADO from any COM environment, such as Visual Basic.

Oracle provides many different products that integrate Oracle applications into Windows environments, including Oracle Fail Safe, Windows security integration, and Active Directory integration.

Through integration with the Microsoft Cluster Service, Oracle Fail Safe ensures Oracle software remains highly available on Windows. Fail Safe supports the Oracle database, application server, and E-Business Suite,

servers on Windows clusters, Oracle Fail Safe ensures that those servers remain available by providing robust fail over support through the Microsoft Cluster Service. Oracle integrates with native Windows security and directory tools that both improve database security and makes user management easier.

Oracle Fail Safe

Oracle Fail Safe is a core feature of Oracle Database 10g Release 2 that provides high-availability for mission-critical solutions deployed on Windows clusters. A cluster eliminates individual host systems as points of failure. Oracle Fail Safe works with Microsoft Cluster Service (MSCS) to ensure that if a failure occurs on one cluster node, then the Oracle databases and applications running on that node will fail over (move) automatically and quickly to a surviving node. Unlike RAC, Oracle Fail Safe is an active-passive cluster, whereby the standby machine does not take over the workload until the active node fails.

Oracle Fail Safe is optimized for Windows customers with database and application workloads that can be handled by a single system. Oracle Fail Safe solutions can be deployed on all commodity Windows clusters. It supports up to eight-node clusters on Windows Server 2003 Datacenter, the maximum node configuration available on Windows today. Supported products include:

- Oracle databases (Oracle10g, Oracle9i, Oracle8i, and Oracle8)
- Oracle Applications release 11i
- Oracle iAS, with the exception of cache products, including
 - Oracle Forms Server
 - Oracle Reports Server
 - Oracle HTTP Server
- Oracle Services for Microsoft Transaction Server
- Oracle Intelligent Agent
- Applications installed as Windows services

SAP, Baan, PeopleSoft, Lawson, J.D. Edwards, and other applications vendors also have validated their software solutions with Oracle Fail Safe.

Oracle Fail Safe includes two main components, a server and a manager. The server component, Oracle Services for MSCS, works with the cluster software to ensure fast automatic fail over during planned and unplanned outages. The manager, Oracle Fail Safe Manager, is an easy-to-use graphical interface that works with Oracle Fail Safe Server on one or more clusters to perform configuration, management, verification, and static load balancing. Together, these components provide a rich set of features and integrated troubleshooting tools that enable rapid deployment of highly available databases and applications—complete e-business solutions.

Oracle Fail Safe supports disaster protection by enabling Logical/Physical Standby support. For improved manageability, multiple clusters now can be managed from the same console.

Logical/Physical Standby and Multi-cluster Manageability

With Logical/Physical Standby integration, the Oracle Fail Safe database can be run on a primary site and any remote standby sites. Fail Safe will monitor the database, bring it online, and take it offline. This allows customers to have single instance fail over within a cluster with minimal downtime, along with added benefits of maintaining standby sites. Oracle Fail Safe is commonly used for planned failovers, such as software/hardware upgrades common to Windows environments. With Logical/Physical Standby, Fail Safe now provides true disaster protection offered by standby. Together, the two products complement each other to provide higher availability and disaster protection than with either product alone.

For customers with multiple cluster deployments, Oracle Fail Safe Manager permits the display of more than one cluster, allowing the user to manage all clusters from the same window, rather than from a separate window for each cluster. Administrators, for example, will be able to modify the database account credentials for databases across all their Fail Safe clusters all at once using a single wizard. Administrators will be able to avoid the tedious nature of comparing policies or modifying cluster display or alert preferences individually.

Security

Oracle Database 10g Release 2 provides tight integration with the native Windows security model to better safeguard application security. In a client/server environment, there are three options today for sites wishing to leverage Windows credentials or authentication services for signing on to the Oracle database.

The Oracle database supports a number of Windows security mechanisms, including single sign-on, Windows Kerberos, and Windows SSL. The ability to store Oracle wallets in the Windows registry eliminates the insecure operating system file storage. Oracle integrates with the Microsoft Certificate Store allowing Oracle PKI and Windows PKI applications to operate together seamlessly.

- The Oracle database includes the Windows Native Authentication Adapter, which installs automatically with Oracle Net on the server and client. This feature allows database users to have single sign-on capabilities by using Windows user credentials to support database authentication. To use this feature, Windows users must be defined as database external users. These users can have external roles assigned to them in Microsoft Active Directory Services that will be respected by the database.
- Oracle Advanced Security, a database option, supports database authentication using Kerberos tickets issued by the Microsoft Key Distribution Center (MSKDC). This capability allows users who have been issued a valid Kerberos ticket in the Windows environment to sign-on to their database accounts without having to provide a username/password.
- The Oracle Advanced Security option also supports database authentication over SSL using an X.509v3 certificate. The Microsoft Certificate Store (MCS) may issue this certificate. To use this feature, the certificate must be contained in an Oracle Wallet configured on the client.

The Oracle Wallet may be stored either in the user profile area of the Windows Registry or in an arbitrary file location on the client.

For web environments, integration with MCS is possible through Oracle Single Sign-on. In addition to supporting password-based authentication mechanisms, the Oracle Application Server through the Oracle Single Sign-on Server supports certificate-based authentication. This certificate can be generated by MCS. To use certificate-based authentication, a user certificate must be populated into the browser wallet and into the corresponding user entry in Oracle Internet Directory.

Oracle Wallets in Windows Registry

Oracle wallets can be stored in the Windows registry, providing increased security on Windows clients. Without this feature, Oracle wallets are often stored on the Windows file system. If the operating system file permissions are not secure, the Oracle wallets are also insecure. Insecure file permissions may come from improper administration or lack of operating system file security. On Windows systems, therefore, public key infrastructure (PKI) security is improved by storing Oracle wallets in the user profile area of the Windows registry. This registry area is accessible only to the properly logged-in user. Multiple Oracle wallets can be used and stored from the profile area.

Single sign-on for Oracle PKI applications is enabled through the Oracle Wallet Manager and Oracle Enterprise Login Assistant tools. The Wallet Manager creates encrypted Oracle wallets, while the Enterprise Login Assistant creates the decrypted wallet. The decrypted wallet is then used by Oracle PKI applications for SSL authentication. Both of these tools are enhanced to support the storage and location of Oracle wallets in the registry, as well as the default file system if required.

Microsoft Certificate Store Integration

Oracle's integration with the Microsoft Certificate Store allows Oracle PKI applications to interoperate with products that use Windows PKI. When this feature is enabled, Oracle PKI security uses Microsoft CryptoAPIs to access the Microsoft Certificate Store. The CryptoAPIs are used for operations, such as signing, encryption, decryption, verification, and validation. The Wallet Resource Locator (WRL) determines the PKI type and other relevant information.

With this feature, Oracle and non-Oracle applications can leverage the same set of PKI credentials (such as certificates, keys, revocation lists, and trustpoints) for public key security services, such as authentication and encryption.

Oracle's Directory Integration and Provisioning platform allows Oracle Internet Directory (OID) to centralize synchronization and provisioning with other directories, include Active Directory. Oracle makes Windows single sign-on possible and user management easier by allowing user mappings to be stored in Active Directory or in OID. Moreover, Net naming (tnsnames.ora) can be stored in either Active Directory or OID for centralized management.

Directory

Directory servers often make user and resource management easier by centralizing administration. Oracle Internet Directory (OID) allows enterprises to centralize their directory servers and leverage their existing directory servers, such as Active Directory. OID's Directory Integration and Provisioning platform consists of a set of services and interfaces built into OID. These facilitate the development of synchronization and provisioning solutions between the directory and other repositories. These may include other directories, such as Microsoft Active Directory Services, application user repositories, or database tables containing human resources information.

This platform includes two services: a provisioning service and a synchronization service. The provisioning service facilitates automatic execution of application-specific user provisioning activities when user or group entries in the directory are updated or deleted. These are executed as PL/SQL procedures that may add, delete or suspend privileges for a user in a connected system. Various Oracle product components use the provisioning engine to automatically create "account footprints" for users managed in the directory.

The synchronization service publishes changes made to data in Oracle Internet Directory to connected agents that subscribe to these specific sets of directory data. It also reads and applies changes made in connected systems. Together, these actions facilitate a "synchronization" of data contained in Oracle Internet Directory and the connected system, such as Microsoft Active Directory Services.

Oracle includes two additional native integration features that leverage both OID and Active Directory (AD) on Windows:

- Native authentication and AD
- Oracle Net naming with AD

Native Authentication and Active Directory

Oracle customers often require enterprise-scale security and schema management. This is especially true for businesses with large user populations. OID is designed to handle these enterprise environments and interoperate with departmental directory servers, such as Active Directory. Oracle security and administration are integrated with both these directory servers to provide seamless access and administration.

Oracle provides native authentication through Windows authentication mechanisms, allowing the operating system to perform user identification for Oracle databases. Oracle supports the core Kerberos security protocol on Windows. With native authentication enabled, users can leverage single sign-on to access Oracle simply by logging onto Windows, reducing end-user administration and eliminating redundant security credentials.

Oracle enterprise user mappings allow many operating system users to access multiple databases as a single global database user. In Windows-only

environments, these enterprise user mappings may be stored in Active Directory. For instance, an entire LDAP organizational unit (OU) in Active Directory can be mapped to one database user.

Oracle stores enterprise role mappings in LDAP. On Windows, this LDAP storage feature is certified on OID as well as with Active Directory. With enterprise roles, privileges for multiple databases can be managed at the domain level through directories. This is accomplished by assigning Windows users and groups to the Oracle enterprise roles registered in the LDAP store.

Oracle Net Naming with Active Directory

Oracle leverages LDAP technology through OID and Active Directory to improve end user database connectivity information management. Traditionally, end users reference database servers with Oracle Net-style names resolved through the TNSNAMES.ORA configuration file. This file must be administered on each client machine. To ease manageability, Oracle Net names can be stored and resolved through OID or Active Directory. Centralizing such information in a directory eliminates administrative overhead and relieves users from configuring their individual client machines.

Furthermore, Windows tools, such as Windows Explorer and Active Directory Users and Computers, can connect to databases and test database connectivity. Oracle tools can also participate. The Oracle Database Configuration Assistant automatically registers database objects in an LDAP directory. The Oracle Net Manager, meanwhile, registers net service objects with the directory. These enhancements further simplify administration.

CONCLUSION

As technology in Windows application development and deployment has evolved, so has Oracle. Oracle Database 10g Release 2 offers unprecedented flexibility for customers to choose the application tools and architecture that fit their requirements without sacrificing performance, scalability, ease of use, and security. Oracle's commitment to the Windows platforms includes new integration with the ADO.NET 2.0 infrastructure and Visual Studio 2005. By designing Oracle Database 10g Release 2 with Windows in mind, customer software applications can achieve seamless interoperability between the operating system and the database.

For more information about Oracle on Windows and .NET, visit

OTN Windows Technology Center: <http://otn.oracle.com/windows>

OTN .NET Developer Center: <http://otn.oracle.com/dotnet>

Oracle on Windows: <http://www.oracle.com/windows>



Oracle Database 10g Release 2 on Windows: Development and Deployment
May 2006

Author: Alex Keh

Contributing Author: Christian Shay

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 © 2006 Oracle Corporation
All rights reserved.