WHITE PAPER

Oracle Database and Microsoft Windows and .NET Interoperability: Packing Much More than Meets the Eye

Sponsored by: Oracle

Al Hilwa

April 2009

Global Headquarters: 5 Speen Street Framingham, MA 01701 USA P.508.872.8200 F.508.935.4015 www.idc.com

IDC OPINION

Though often coming without much fanfare, the interoperability initiatives between Oracle and Microsoft technologies around Oracle database support on Windows and with .NET application development have been substantial and have seen several iterations and improvements. Today, interoperability has reached a significant level of depth and maturity such that Oracle database customers developing .NET applications can truly leverage the best that these two vendors have to offer in their respective areas of excellence. In particular, the interoperability tools enable the following specific scenarios:

- Oracle databases that can run on the Windows platform allowing Windows users to enjoy the full complement of differentiating features that Oracle offers, such as Real Application Clusters (RAC), Automatic Storage Management (ASM), Real Application Testing, Database Vault, and Virtual Private Database (VPD)

- Application developers who utilize the Visual Studio integrated development environment (IDE) to author and debug applications that access Oracle databases, running on any platform, with a significantly higher level of productivity than is possible without them

- High-performance .NET applications that are written to the Oracle database and that utilize its unique and differentiating capabilities (e.g., Oracle Spatial Option, SQL Analytics, User-Defined Types [UDTs], RAC, and VPD) in a high-performance manner

- ASP.NET-based Web applications that utilize an Oracle database for storing and retrieving a variety of Web site information for ASP system-of-record needs

- Complex .NET logic that can operate with the data access advantages of Oracle database stored procedures running on the database tier

TABLE OF CONTENTS

P

LIST OF FIGURES

P

IN THIS WHITE PAPER

In this white paper we explore two key areas of interoperability between the Oracle and the Microsoft worlds. First, we assess Oracle's strategy for supporting the Windows platform; then we assess the overall development situation for .NET developers wishing to write applications to Oracle, the motivation for the end-user organizations and for Oracle to support this combination, and the tools that Oracle provides to support it.

SITUATION OVERVIEW

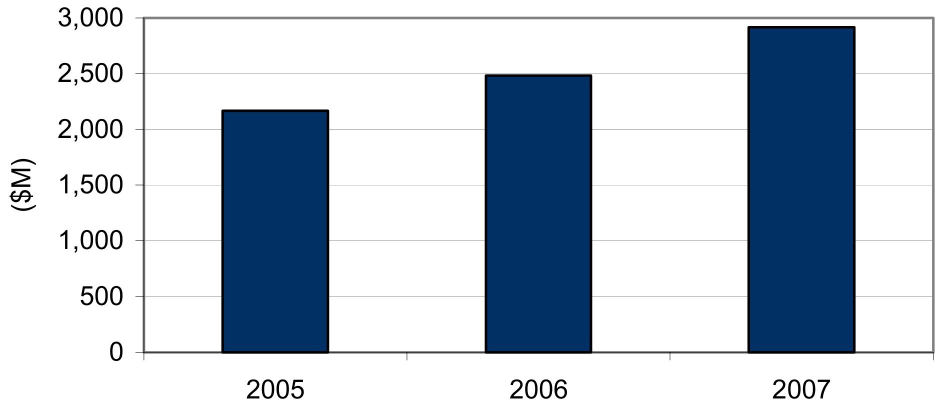
As Oracle has retained its lead in the database market, one of its largest database platforms by revenue is Windows. Additionally, a great deal of application development today takes place in the Visual Studio IDE, essentially to write applications in Microsoft languages and to deploy them on Microsoft's .NET Framework as the container application server. Thus, in addition to the large number of deployments that Oracle enjoys on the Windows platform, there is a large contingency of developers who build and maintain .NET applications that run against Oracle Database back ends on a variety of platforms, including various versions of Linux, Unix, and Windows. As Microsoft has pushed its database offering higher up toward the enterprise, touting integration with the overall Microsoft stack (especially the Visual Studio IDE) as a key differentiator, we have also seen Oracle reach into the Microsoft developer base to ensure that .NET applications can be developed productively and can run efficiently against Oracle databases.

THE ORACLE DATABASE ON WINDOWS

Key areas of collaboration between Microsoft and Oracle include running the Oracle database on the Windows operating system, an effort which began in the 1990s when a Microsoft offering in the server space began to take shape. In 1993, Oracle shipped a database for Windows NT before any independent database vendor, and today, according to revenue data tracked by IDC, Windows constitutes the largest single platform for Oracle. This is not surprising because Windows remains the largest platform category for relational database serving across the relational database market, according to IDC data, with over 46% of the RDBMS market by dollar volume being sold to run on Windows Server. Additionally, as illustrated by IDC data estimates in Figure 1, Oracle's database revenue on the Windows platform has experienced significant growth over the past three years. Today, Oracle maintains a significant share of the Windows database market overall and a sizable community of customers that run Oracle on the Windows Server platform. Thus, despite overt competitiveness with Microsoft in many market spaces, Oracle invests a significant amount of engineering effort in ensuring that the Oracle database works well on Windows. Overall, the fruits of this investment have paid off handsomely with strong benchmarks and Oracle's most compelling features such as RAC, Active Data Guard, ASM, Real Application Testing, Database Vault, and VPD, which are all available on Windows.

FIGURE 1

Oracle RDBMS Revenue on the Windows Platform, 2005–2007



Source: IDC, 2009

Oracle's Platform Strategy for Windows

To support Windows successfully, Oracle engineered its engine in the mid-1990s for Windows' thread-oriented concurrency architecture. While Unix and Linux systems tend to have a lightweight process model, the analogous process features in Windows are considerably higher in latency and are considered too heavyweight for high-performance concurrency work. Microsoft provided the threading capability instead for lightweight concurrency purposes, but threads did not provide the same level of code isolation provided by processes. This meant that software ported from Unix architectures on a process-to-process basis rarely provided sufficient performance. Oracle was aware of this and thus invested considerable porting effort into a thread-based implementation that provided both the needed performance and stability to be competitive. The result of this port, which was brought to market in 1993, was early adoption, and for a few years (until the 2000–2001 time frame), Oracle was the leading market share holder in Windows database revenue, according to historical IDC data, selling more databases on a dollar volume basis than any other vendor, including Microsoft. Windows as a database serving option was widely adopted and established in the marketplace by the early 2000s, and Oracle continued to stay ahead of the game in terms of features and capabilities by bringing the full complement of its family jewels to the Windows platform. This has allowed Oracle to be a top player on Windows platforms. While Oracle's high-profile support for Linux has also been successful and has tied up the public airwaves in terms of PR bandwidth, customers of Oracle on Windows continue to report a high level of support and satisfaction, and a considerable amount of Oracle's energy continues to be invested in the Windows platform. IDC expects the level of Windows support from Oracle to continue. Evidence of Oracle's continued commitment came recently in the form of relatively rapid support for new Windows releases. For example, Oracle announced during Oracle OpenWorld 2008 that it would release a port of the Oracle

Database 11g for Windows Server 2008 in the first quarter of 2009. Both the 32-bit and 64-bit (x64) versions of Windows Server 2008 are now supported by Oracle Database 10g R2 and 11g.

Oracle DBMS Integration Touch Points with Windows

In addition to adaptation to the specific Windows thread architecture, over the years Oracle has worked to bring its key technologies to Windows. Many of these technologies, such as Oracle RAC and Fail Safe, have deep platform dependencies and thus have been the recipients of significant engineering investment. Noteworthy areas of integration are as follows:

- **Full support for the Windows core operating system.** In addition to supporting both 32-bit and 64-bit versions of Windows Server, Oracle Database includes performance optimization features such as support for large pages, NUMA (Non-Uniform Memory Access) support, VLM (Very Large Memory) support, and asynchronous I/O capability in addition to the thread-based architecture discussed earlier.

- **Integration with Microsoft Transaction Server (MTS).** Oracle Services for MTS allow Oracle databases to be used as resource managers in Microsoft Distributed Transaction Coordinator (MSDTC)–coordinated transactions and support Enterprise Transaction Services (or Microsoft Transaction Manager).

- **Integration with Microsoft Volume Shadow Copy Service (VSS).** Oracle VSS Writer is transparently integrated with Windows VSS. It supports automatic online point-in-time copy of Oracle Database using a VSS requestor, which can be used to offload backup and reporting to another server using transportable snapshots. For Oracle DBMS 11g, the new writer supports copies at the file level and at the volume level, allowing implementations of various scales to take advantage of the new capability.

- **Integration with Microsoft Cluster Service (MSCS).** Oracle Fail Safe works with MSCS and provides high availability for solutions deployed on Windows Clusters. For Oracle RAC, Oracle has developed the Cluster File System, which is an Oracle-written file system on Windows for shared disks to allow one software image to be used by all nodes in a cluster.

- **Windows security and Active Directory integration.** Oracle Database supports Windows security and Active Directory through single sign-on, integrated user provisioning, database registration, and name resolution. Oracle supports single sign-on capabilities for Windows-only environments with Windows Native Authentication adapter, which uses Kerberos and allows Active Directory users and groups to be mapped to database users and roles. Oracle also supports single sign-on with SSL and Kerberos authentications for heterogeneous environments. Oracle configuration tools can be used to register Oracle databases and Oracle Net service names in Active Directory. End users can then discover and use these names to connect to the service, thereby eliminating administrative overhead and the requirement for a client system configuration. Finally, Oracle also provides a comprehensive Identity Management solution that supports virtual directory and services for synchronization and provisioning with Active Directory and can work in a heterogeneous environment. Customers have the choice of storing directory information for many services in either Active Directory or Oracle Internet Directory (OID) — the corresponding Oracle offering. Customers in a Windows-only environment who choose not to use OID or Oracle Identity Management (OIM) — Oracle's end-to-end identity management offering — can integrate Active Directory directly with the Oracle database to provide single sign-on and Oracle Net naming management. A specific example of the new capabilities in 11g is the ability to disallow anonymous access to a directory's database service information, a new 11g Kerberos implementation that now makes use of more secure encryption algorithms such as 3DES and AES in place of DES. The level of integration with Active Directory and Windows security features presents Oracle-on-Windows users with a considerably higher level of manageable interoperability than ever before.

- **Administration support.** Oracle Database administration is enhanced through integration with the Windows Event Log, Performance Monitor, and Service Manager. Additional administrative features in Oracle DBMS 11g include the Oracle Cluster Manager, which offers management for multiple clusters from a single console.

ORACLE'S DEVELOPER STRATEGY

The collection of components that Oracle calls Oracle Data Access Components (ODAC) includes sophisticated IDE extensions, high-performance data access, and the ability to program .NET application logic on the back-end Oracle database. These tools provide a significant improvement in developer productivity and permit high-performance applications to be written with .NET that rival any application written in any other language in complexity.

Historical Commitment to the Visual Studio Developer

To specifically address the developer angle, Oracle engaged with the Visual Studio team early in the .NET days. Specifically, Oracle joined the Visual Studio Industry Partner (VSIP) program, which permits partners to gain deep insight into how to integrate extensions into the Visual Studio IDE. Engagement in the VSIP program has allowed Oracle to work collaboratively with Microsoft to build, test, and refine a set of tools that is increasingly proving essential for the .NET-on-Oracle developer. The effort has resulted in the development of the ODAC set of tools. The following sections highlight the four most important tools in the ODAC toolset.

Oracle Developer Tools for Visual Studio

Oracle's most comprehensive offering for the .NET developer is the Visual Studio plug-in known as Oracle Developer Tools for Visual Studio, or ODT for short. The first release of this product dates back to the mid-2005 time frame when Oracle first introduced this capability. The early release introduced an Explorer window that displays the Oracle database schema via a tree control and a number of other features, including a set of Designers and Wizards and a Data Window interface for editing Oracle data. ODT was an important strategic and marketing response to a generally improving Microsoft SQL Server database, which is often positioned by Microsoft as attractive to developers and integrated with Visual Studio. However, more than a marketing device, ODT was well-received by the significant number of developers and development shops that built custom applications against an Oracle database back end running on any operating system platform. ODT brought the state of Visual Studio development against Oracle Database many steps forward, and the improvements in the 11g release are expected to generate interest and adoption from an even larger community of developers.

Architecturally, ODT is an add-in to Visual Studio and comprises multiple tools, each serving a specific function that addresses a specific area of developer productivity. The following list highlights the capabilities of the latest 11g generation of the tools:

- Support for Visual Studio 2008, Visual Studio 2005, and Visual Studio .NET 2003

- Integration with Visual Studio through Microsoft Server Explorer, which is a tree control that is built into Visual Studio and is used for browsing system resources. Developers can now use it to browse Oracle schemas, and this explorer includes context menu items for launching appropriate Oracle Designers and Wizards to create and alter schema objects. Microsoft's Query Designer, which makes it easy to visually design queries, can now be used to design queries against Oracle database. Visual Studio's ASP.NET Web application development environment is also fully integrated to make rapid generation of Web applications based on Oracle easy and fast.

- Designers and Wizards are provided to make various tasks simpler and faster (e.g., the Table Designer, which creates database tables).

- Oracle Data Window allows easy table viewing and navigation to specific rows as well as the ability to update and modify the data.

- Automatic Code Generation permits "crud"-style (create/retrieve/update/delete) code to be generated automatically when schema objects are dropped on Visual Studio forms. This code generation is provided for developing client/server applications as well as ASP.NET Web applications and Microsoft Office applications.

- PL/SQL Editor permits the editing of PL/SQL stored procedures, functions, packages, and triggers. Auto–syntax completion is provided with popups, and collapsible code regions are supported. Additionally, integration with the Visual Studio task list to handle generated compile errors leads to productive debugging.

- PL/SQL Debugger brings debugging of PL/SQL code to a level of parity with C# code, allowing the stepping back and forth from .NET to PL/SQL code.

- Oracle Database Project with Source Control Integration makes it possible to generate SQL scripts for Oracle schema objects that a .NET application uses, manage them in an Oracle Database Project, and check them into source control.

- An Oracle SQL Script editor is integrated into Visual Studio that honors Oracle SQL syntax. The scripts can be executed with an underlying SQL*Plus execution engine (which ensures that the scripts execute in the same way they would with other Oracle scripting tools).

- Integrated Help System now permits the developer to navigate the SQL, PL/SQL documentation along with the Error Reference Manuals on par with the navigation of Visual Studio .NET help and documentation.

- Oracle Database UDTs are supported throughout Visual Studio, providing the ability to create UDTs in Oracle with multiple new designers. A powerful UDT Custom Class code generation wizard makes using UDTs from .NET code easy and fast. An application can be rapidly created that utilizes UDTs with almost no coding required.

- A Grant/Revoke Privileges Wizard gives developers the ability to assign database privileges to schema objects and other users.

- .NET Deployment Wizard allows .NET code to be developed and run in the same manner as stored procedures "inside" the Oracle database (this capability is covered later in this white paper).

- Import Table Wizard makes it easy to move tables and their data from external data sources such as Microsoft SQL Server, Microsoft Access, and Microsoft Excel spreadsheets into Oracle Database.

- SQL Query Window provides a more integrated alternative to the SQL*Plus interface for writing ad hoc SQL and submitting it to the database.

- Stored Procedure Testing allows the testing of these components with the manual provision of parameters and the handling of complex result types.

Finally, it should be noted that 11g version of the tool can be used against older versions of the Oracle database running on any supported Oracle platform. Overall, Oracle Developer Tools, especially with the new 11g release, provide productivity improvements that truly enhance the experience of the .NET developer community that works with Oracle databases.

Oracle Data Provider for .NET

The improvements in the development tools discussed earlier are enabled and made more useful with the data access components provided, and now considerably enhanced for 11g, by Oracle. Known as Oracle Data Provider for .NET (ODP.NET), this feature-rich middleware component makes it possible to develop and deploy applications against back-end Oracle databases in a reliable, secure, and high-performance manner. More specifically, ODP.NET allows the key differentiating features of the Oracle database to be surfaced (e.g., RAC, Active Data Guard, VPD, UDTs) and utilized for maximum use of hardware resources, high performance and throughput, and high availability. The new release of the driver features a list of additive improvements that build on a legacy of nine releases since the provider was first released in its current form with the Oracle 9i R2 database, including support for the latest additions of .NET, namely .NET 3.0 and 3.5. The following are some of the most important capabilities of ODP.NET as of the 11g release:

- **ODP.NET load balancing.** ODP.NET is fully integrated with the runtime load balancing feature of Real Application Clusters. When an application requests a connection from the pool, ODP.NET automatically dispenses the connection to the least loaded database node so that work is evenly distributed. It provides real-time balance of database workloads across machines based on current server machine activity, preventing any one machine from becoming overburdened compared with other cluster machines.

- **Connection pooling improvements.** The connection pooling engine is highly configurable with parameters that permit the tailoring of the connection pool to the size of the user population and application requirement. New performance counters allow better monitoring of the connection pool behavior. Additionally, Windows users can now authenticate with connection pooling.

- **ODP.NET Fast Connection Failover (FCF).** This feature enables fast detection of failed instances and automatic removal of ODP.NET connections to a downed RAC member or Data Guard instance from the connection pool to ensure high availability for .NET applications.

- **Support of SecureFiles for unstructured data management.** SecureFiles provides file system–like performance, deduplication, compression, and encryption for reducing storage space and improving security and performance. Large Objects, or LOBs, which are used to store images, videos, and other binary data, can be troublesome for client/server applications because fetching them is time-consuming and can lead to performance issues. The 11g release of ODP.NET implements round-trip optimizations that benefit all applications using LOBs. This is in addition to the sophisticated support provided by prior releases of ODP.NET for LOB fetching, allowing LOBs to be accessed at random points within the object and supporting a high degree of developer configuration and control of LOBs that are fetched.

- **Caching.** The caching capabilities built into ODP.NET are probably one of the most important improvements in the provider for the .NET database developer. Caching features including statement caching, metadata caching, and client result cache. Statement caching eliminates the need to reparse statements, thereby saving valuable time for statements that are executed often. Without requiring changes to .NET code, the client result cache operates automatically and utilizes a database notification feature to remain consistent. Finally, ODP.NET provides control over data fetch sizes by correlating them to the fetched data row sizes, dynamically allowing the developer to tune data retrieval performance.

- **Support for PL/SQL stored procedures and result sets.** ODP.NET provides full support of PL/SQL data types, which can be mapped to .NET data types on the client side. It supports REF Cursor to defer data fetching and retrieving to only when data is needed. It also supports associative arrays to pass large amounts of .NET data of the same data type. ODP.NET enables batching with anonymous PL/SQL to execute multiple commands in a single round trip to the database.

- **Leveraging Oracle security features.** ODP.NET client identifier allows application context tracking and can be used in conjunction with VPD and Label Security. ODP.NET proxy authentication allows multiple unique users to share the same connection pool user and improve performance and scalability.

- **User-Defined Types.** ODP.NET allows the developer to map Oracle objects, collections, and REFs to .NET custom types. ODP.NET has provided full support for the gamut of native Oracle data types from within .NET, such as REF Cursors. In many cases, these types of advantages over the .NET types (e.g., 38-digit decimals instead of 28-digit decimals in .NET) make it possible to implement applications that take full advantage of the Oracle database.

Oracle Providers for ASP.NET

Oracle Providers for ASP.NET is a collection of ASP.NET 2.0 providers that follow the ASP.NET 2.0 provider model and use the Oracle database as the data source. Microsoft ASP.NET 2.0 includes a number of services and providers that store application state in databases and other storage media. Oracle Providers for ASP.NET ensure that .NET developers, already familiar with ASP.NET providers, can easily learn and use these providers. Also, existing ASP.NET applications, using ASP.NET providers, can easily leverage Oracle Database by choosing Oracle Providers at deployment time.

Several new providers have been added in the 11g release that generally permit the storing and retrieval of the objects they manage in the Oracle database. The new providers are as follows:

- **Membership Provider** manages registered user information, providing the capability to verify log-in credentials and change passwords, among other tasks, as well as create/delete functionality.

- **Role Provider** manages Web site user role information.

- **Site Map Provider** manages site map information by allowing the building of an upside-down tree of site map objects as well as providing methods for retrieving nodes from the site map stored in the Oracle database.

- **Session State Provider** manages session state information through the ASP.NET session state service.

- **Profile Provider** manages user profile information.

- **Web Event Provider** manages ASP.NET health monitoring event information and provides efficient buffering and flushing capabilities.
- **Web Parts Personalization Provider** manages personalization data.
- **Cache Dependency Provider** automatically invalidates ASP.NET cached data based on changes made to Oracle database data.

Oracle Database Extensions for .NET

An interesting capability first introduced in Oracle Database 10g is the ability to run .NET code as Oracle stored procedures. It is common knowledge that much enterprise application code continues to be developed and maintained in database stored procedures. Stored procedures provide a declarative (SQL) and thus highly compact and productive approach to development with the added benefit of a much lower impedance mismatch factor between database objects and types and application code objects and types. Thus, much of the business logic of many important applications was developed, mostly during the era of preapplication server client/server architectures when the network inefficiencies of putting code on the client were prohibitive. The emergence of standardized SQL in the late 1980s and early 1990s provided an easy and productive alternative, but writing application code in SQL is a skill that seems to be increasingly lost to the more procedurally oriented world of application developers. What's more, the set of tools and capabilities (e.g., version, configuration, and project management) were largely absent for this type of code, at least in anything resembling an integrated manner. Thus, to retain the benefits of code that is tightly coupled to the database and provide languages more familiar to the typical application developer, a desire emerged to support stored procedures in multiple languages. Oracle was early to the game in the mid-1990s with Java stored procedures, and then with SQL Server 2005, Microsoft began to integrate the .NET Common Language Runtime (CLR) with the SQL Server stored procedure engine. Not to be outdone, Oracle invested in similar engineering skills to additionally support CLR stored procedures in the Oracle database. Today this capability is available in the form of Oracle Database Extensions for .NET. While the jury is out on whether such capabilities, for any database, will become mainstream approaches to application development, there are applications that are simultaneously database and algorithmically intensive in an intertwined fashion that can benefit greatly from them. Thus Oracle has continued to maintain this capability and further supports it with the deployment mechanism in the Oracle Developer Tools profiled earlier.

The Oracle Database Extensions for .NET provide great flexibility. With the exception of an optional "context connection," which allows a stored procedure (or function) to use the same connection as the caller, the stored procedure code is indistinguishable from client code and can also be written such that it can run on either without change. This ability allows for enhanced code reuse and deferred decision making on the exact tier a code is run in, enhancing performance fine-tuning flexibility.

CHALLENGES/OPPORTUNITIES

The capabilities highlighted in this white paper have demonstrated the levelheadedness of business decision making behind the two software giants as they have collaborated to support their joint users. Nevertheless, we ponder in this section some specific challenges and opportunities facing this joint scenario:

- Can Oracle and Microsoft turn up the marketing on these capabilities, highlighting the mutual benefit to their joint customers? IDC has found that more awareness is needed in this area and that customers overall perceive the two environments are not so interoperable. The reality is this has begun to change, but both companies need to drive home the message. Better interoperability is a win-win situation that is likely to benefit both companies.
- While the level of support provided so far for .NET developers on Oracle databases is significant, the .NET and Visual Studio world continues to evolve. In particular, many enterprise shops with existing .NET development practices have begun to adopt Visual Studio Team System (VSTS) as they evolve toward a life-cycle approach to the application development process. As Microsoft makes Visual Studio Team System 2010 extensible to enable support of other databases, it is an opportunity for Oracle to provide deeper integration with VSTS as this user base grows to gain the same productivity gains for testers, architects, and team managers that standard application and database developers have attained with the current iteration of ODT.
- Another opportunity available to Oracle is to take advantage of its unique position as a broad provider of database and applications technology to bridge the worlds of Java and .NET. There are many independently evolving technologies in the Java and .NET worlds that can benefit from standardized layers that provide cross-targeting. An example is the world of object-relational mapping where there are many standards, including Oracle's TopLink. In the .NET world, the Microsoft Entity Framework is likely to take the payload for object-relational functionality. With Oracle's leadership in data management, IDC expects that an opportunity exists to provide integration across these two worlds.

CONCLUSION

While Oracle and Microsoft are fierce competitors in many areas, both have realized that they have a significant customer base in common and that it is in their collective interest to support the interoperability needs of those customers. The reality is that Oracle offers compelling database technology and Microsoft offers compelling operating system and development tools technology. Additionally, the Oracle-on-Windows and .NET-on-Oracle scenarios are both prevalent in their respective user bases. Therefore, both have found it important to collaborate in these areas, often without hype or fanfare. The collaboration has intensified as Oracle has sought to establish itself as a top 2 application software player through an acquisition strategy that has produced an even more complex and variegated installed base and as Microsoft has found it necessary to reach customers unwilling to run an entirely homogeneous Microsoft stack. The collaboration between Oracle and Microsoft, at least from a platform and tools perspective, has been one that has put users first and has kept them above the fray.

CASE STUDY

In the process of researching this paper and as part of our day-to-day research activities, IDC conducts in-depth discussions with IT organizations. We have found a significant user base for both the Oracle-on-Windows and the .NET-on-Oracle scenarios. The various interoperability components and the many touch points of Oracle and Microsoft technologies appear to be well-exercised in the general user community of both companies. As we indicated earlier in the IDC platform revenue breakdown for the Oracle database, the number of deployments of Oracle on the Windows platform may well exceed those of any other platform running the Oracle database. For this reason, we chose to profile the newer .NET capabilities with the following case study.

Lipper, a Thomson Reuters Company

Lipper, a Thomson Reuters company, provides independent insight on collective investments, including mutual funds, retirement funds, hedge funds, and fees and expenses to the asset management and media communities, covering over 193,000 funds globally. Lipper delivers premium data, fund ratings, analytics, and commentary through specialized products. Additional information on Lipper is available at **www.lipperweb.com**. A core offering of Lipper is its new premium desktop, Lipper for Investment Management (LIM), which provides Lipper's clients a rich environment for exploring fund intelligence and analytics. LIM places heavy demands on user interactivity, as it was developed as a rich-client application using the .NET Framework. At the same time, the base of fund information utilized by LIM makes heavy demands of the database management system hosting it; thus, Lipper has long since selected the Oracle database platform as LIM's back end. This combination is quite common in the industry, allowing Lipper to leverage the best development and database skills available from the broadest talent pools available for these skills, namely .NET developers and Oracle DBAs.

Today Lipper is running on the Oracle 10g system and leverages Oracle database for supporting current and historical analytics. The Lipper premium desktop application, thus, for all its user interface richness, is primarily a platform or shell for receiving and displaying the time-series data stored in the database. The application is architected around a layer of Web services that reach out to the Oracle database APIs to provide a rich set of capabilities that include screening operations, data segmentation, and flexible analytics.

A Case for ODP.NET

Because of this strong dependency on data in the architecture of the application, much rests on the data access technology embedded in the application. The availability of Oracle Data Provider for .NET resolved an important issue of how to achieve high-performance interoperability between the .NET and Oracle worlds in a well-supported manner that harnesses the features and capabilities of both worlds.

"ODP.NET enjoys specific differentiation from other providers in its ability to allow flexible fetch sizing and batch sizing by configuring the default fetch size," says Bill Evjen, Global Head of Platform Architecture at Lipper. In fact, the application enjoys the use of many unique Oracle database features such as the VPD capabilities, which support the sophisticated data segmentation requirements of LIM. VPD permits a much higher level of granularity to be applied on items stored in the database while permitting otherwise like data to be stored in the same table, considerably reducing management overhead. LIM also takes advantage of Oracle's UDT features to support the many fund types available.

"While the VPD capabilities in the Oracle database are crucial for our needs, we also take advantage of other capabilities in the provider such as statement caching," Evjen says. In fact, the Lipper developers, being heavy Visual Studio users, are now also heavy users of most of the technology Oracle provides for .NET developers. For example, the debugging features and the SQL scripting capabilities in the Oracle Developer Tools for Visual Studio profiled earlier in this paper have now become part of the essential development workflow.

The Lipper problem of providing high-performance access to Oracle data in the context of a .NET application was resolved with a highly competent ODP.NET implementation. The value of differentiated features such as Oracle VPD is allowed to shine through.

The Lipper story provides an excellent illustration of why Oracle must continue to invest in its .NET efforts and why these tools can avail their users of a unique combination of features not otherwise possible if they were to utilize an integrated stack of a single vendor.

Copyright Notice