Building Data-Driven Applications Fast using Oracle Developer Tools for Visual Studio

Introduction

Enterprise development has always involved working with the Oracle database management system. According to a recent survey conducted by developer community portal TheServerSide, a full fifty percent of enterprise developers use Oracle on the back end of their applications. This attests to the popularity of the Oracle database in applications where performance and reliability are critical.

.NET developers using Visual Studio can be intimidated by targeting an enterprise Oracle database. As an enterprise-quality database can look complex to a developer who may find a large number of features that could be useful in building data-driven applications, but don't have a good idea of how to use those features.

Further, .NET developers are likely to have a greater familiarity with Microsoft SQL Server, because of its availability to Visual Studio users and its feature support from within Visual Studio. .NET developers who know how to work against a SQL Server database discover that switching the database to Oracle is easy and straightforward. There is no learning curve because the development environment is the same with Server Explorer in Visual Studio and the same ADO.NET data access API. With Oracle and ODT, .NET developers can use the latest versions of Visual Studio 2008 and .NET Framework 3.5.

However, through the use of Oracle Developer Tools for Visual Studio (ODT), developers can easily use a set of tools that are integrated with Visual Studio to build data-driven applications, edit data, and create and debug .NET stored procedures. ODT can be freely downloaded from the Oracle Technology Network (OTN), making it a must for every Visual Studio developer working with Oracle databases.

.NET developers can use ODT to quickly build data-driven .NET applications. ODT helps developers generate code, write, debug, and deploy PL/SQL and stored procedures, create user-defined types, and perform other activities that result in the rapid development of high-quality applications. ODT provides ease of use and unique features for working with the Oracle database, with full integration into Visual Studio.

In addition, there are other components that make it possible to more easily work with Oracle from Visual Studio. For example, Oracle Data Providers (ODP) for .NET makes it easy to have

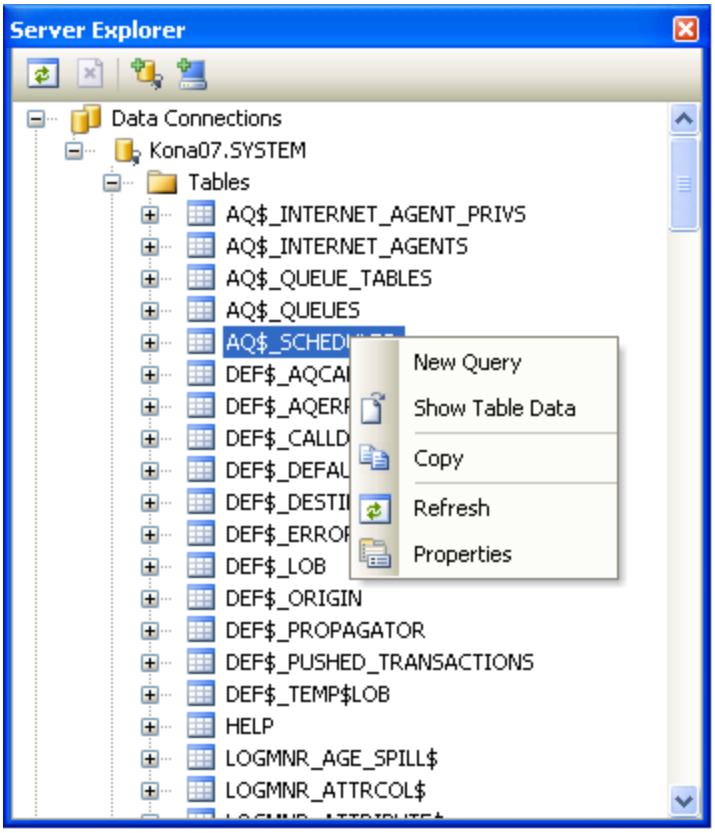
The essentials of the problem – .NET developers often lack familiarity on working with the Oracle DBMS, and may feel intimidated by the extensive feature set – make it challenging to overcome. However, ODT bridges the gap between Visual Studio and the Oracle DBMS, enabling .NET developers to use the environment they are familiar with to get the most out of the database.

An Overview of ODT

ODT and other Oracle products can be freely downloaded for development purposes from the Oracle Technology Network at otn.oracle.com. The download itself can be found at http://www.oracle.com/technology/software/tech/windows/odpnet/index.html. Installing is a matter of unpacking the zip file and running the installation routine. ODT takes only a few minutes to install and integrate with Visual Studio. The next time Visual Studio is launched, ODT features become available in from within Visual Studio solutions and projects.

Further, developers can also freely download a copy of the Oracle database Express Edition at http://www.oracle.com/technology/software/products/database/xe/htdocs/102xewinsoft.htm . The Oracle Express Edition can be installed largely automatically in just a few minutes. Once installed, you can administer the database through a Web-based administrator's console. Alternatively, many of the data administration features are also available from within Visual Studio using ODT.

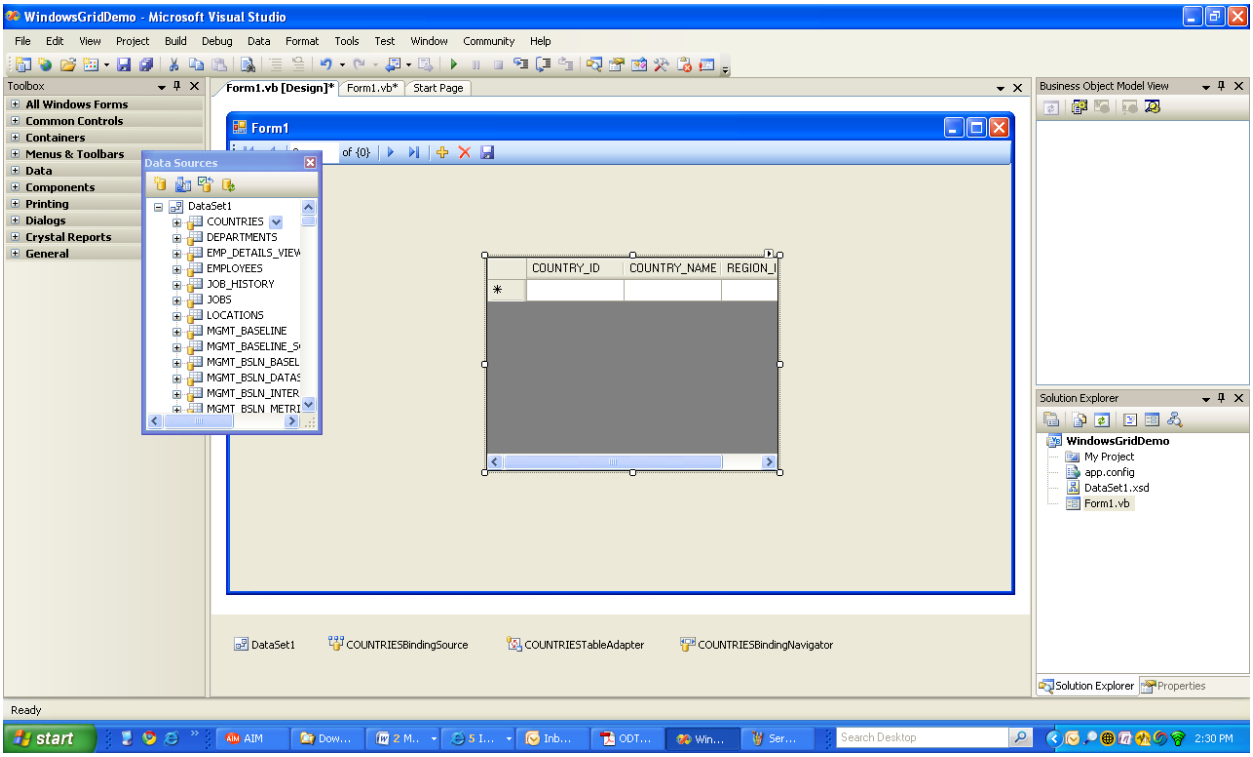
Visual Studio integration starts in the Microsoft Server Explorer window. The Server Explorer, which enables Visual Studio users to browse data sources, goes much further with the use of ODT. In addition to being able to browse data sources, ODT enables the use of the Server Explorer to browse database schema objects which are represented as nodes in the Server Explorer tree. Each object node in the schema offers a menu which allows operations on that node.



ODT enables the Visual Studio Server Explorer to browse Oracle database tables.

For example, you can select Retrieve Data, which opens up the Oracle Data Window for viewing data. You can also use this menu to get access to the Oracle Table Designer for fast creation or modification of an Oracle table. It saves developers from having to work through a DBA for database changes, or using the database administrative tools themselves. For quick database creation or changes to test specific parts of an application, the Oracle Data Window provides a direct way to set up and use a sample database.

The Visual Studio Data Sources window also provides easy access to data in an Oracle database and turn it into code. Oracle data sources can then be dragged and dropped onto the design canvas to immediately generate both the control .NET code which accesses an Oracle Database. That action places the data source in the panel beneath the canvas and a default data grid onto the canvas itself. You can then get access to the data in the Oracle database through the control.



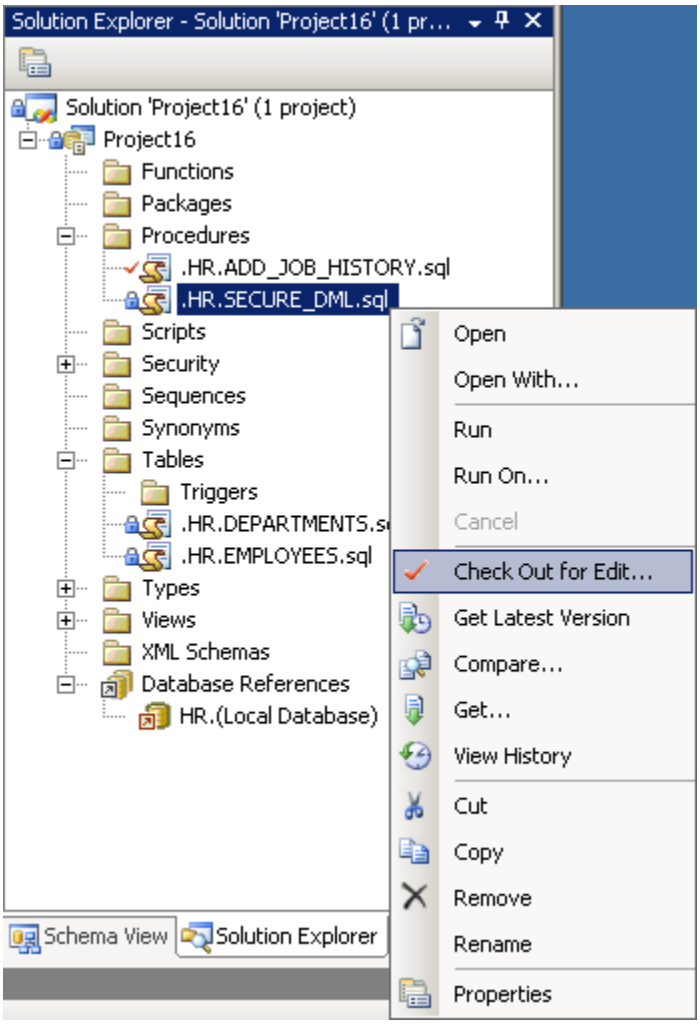
ODT enables developers to work from within the Visual Studio Data Sources Window to generate code for direct database access.

Developers working with Microsoft Office using Visual Studio Tools for Office (VSTO) can likewise rapidly generate Office-based applications in a similar manner – by dragging and dropping an Oracle data source onto an Excel spreadsheet or Microsoft Word application. A similar approach can be used with ASP.NET applications. Dragging a data-driven widget onto the Visual Studio Web design canvas launches the Visual Studio Data Source Wizard. This

wizard automatically generates the required code for data access and then connects this code to the GridView control.

The Oracle ASP.NET Providers make it straightforward for Web application developers using Visual Studio to take advantage of Visual Studio Web controls in establishing database connectivity. Among the advanced controls available are the Login control or the Sitemap control, which enable developers to build in features that might otherwise take extensive coding in ASP.NET.

For new projects, ODT provides an Oracle Database project type that immediately establishes the database connection and creates repositories for the various types of database files that might be created as a part of the project. While this project is most appropriate for rich client applications, it helps developers get started by providing some of the foundation for beginning the development process. It also provides additional features. For example, double clicking on any script file in the Oracle Database project opens the script in the Oracle SQL Editor, enabling developers to easily edit those scripts.



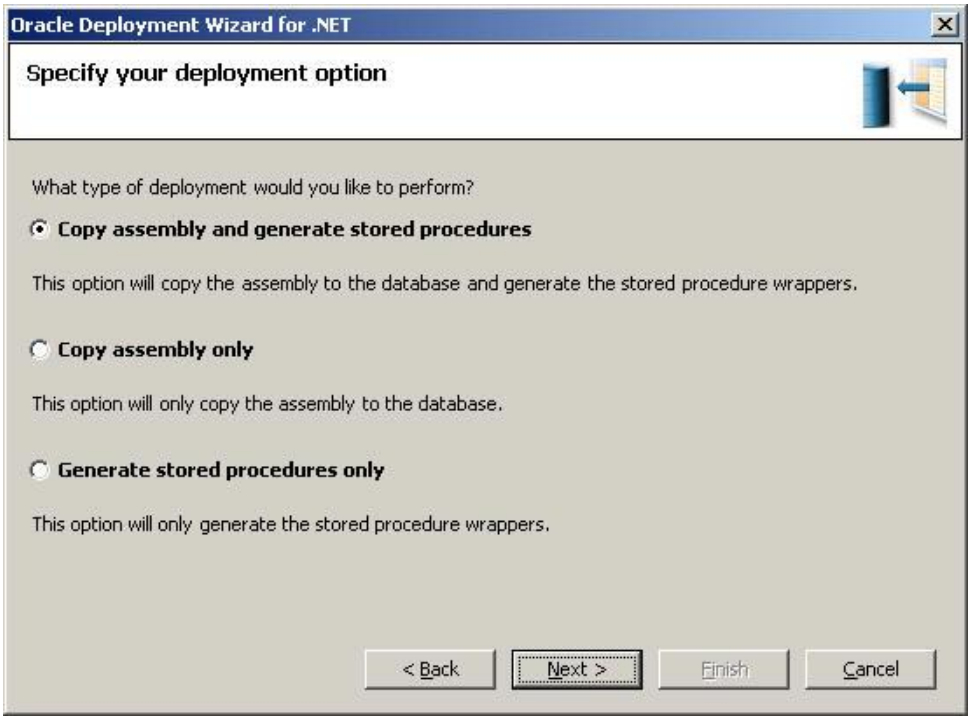
ODT provides for an Oracle database project type that creates a project structure supporting many of the files that might be used in a data-driven application.

Whatever the type of application is under development – rich client, Web, or Office – the ODT plug-in for Visual Studio provides features for quickly getting started in building that application. In particular, it enables Visual Studio developers unfamiliar with Oracle and perhaps a little unsure of how to leverage its features to quickly get started with building data-driven applications.

ODT and Stored Procedures

Experienced Oracle developers are used to working with PL/SQL stored procedures, which are stored inside Oracle and run in the same process as the database. .NET developers, on the other hand, also have the option of writing managed code as a type of stored procedure. A .NET stored procedure, unlike PL/SQL code, runs in an external process and the .NET code is compiled into a .NET assembly. The assembly is managed code compiled into the .NET intermediate language (IL) and JIT-compiled into machine code and executed when called at runtime.

Visual Studio developers write .NET managed stored procedures just as they would any other method in code. Once written, in order to deploy the stored procedure, select Build |Deploy Solution from the Visual Studio .NET 2003 menu. This will start the Oracle stored procedure deployment wizard, which provides for automatic deployment of these components. The wizard allows you to select the deployment option you wish to use. If you have not yet deployed this procedure, accept the default option to deploy the assembly and create stored procedure wrappers in the database.



As a part of the stored procedure deployment process, developers can specify what they want to deploy.

Because the stored procedure is a normal .NET assembly, it can be debugged just as you would any .NET assembly. It only requires that the assembly be built in debug mode and the resulting PDB file placed in the location of the assembly itself.

Oracle Data Providers for .NET

In almost every case with databases, the best performance is achieved by using a provider optimized for that database platform. The standard data providers are available with Visual Studio, and they work well with Oracle. However, data providers optimized for Oracle will perform significantly better. The Oracle Data Provider for .NET (ODP.NET) provides standard ADO.NET data access, while exposing Oracle database-specific features, such as XML DB, data access performance optimizations, and Real Application Clusters (RAC) connection pooling.

Oracle's unique .NET features include data access performance and tuning capabilities, including client result cache, configurable result set data retrieval, array parameters, Ref cursors, configurable LOB and SecureFiles retrieval, and more robust database change notification (DCN). In .NET, developers can use these database capabilities to better tune their data retrieval and updates.

Because the Visual Studio project must connect to an Oracle database, it is necessary to add a reference to the dynamic link library containing the Oracle data provider. Within the Solution Explorer, the developer can select the References node to add that reference. Once the data

provide is installed and referenced, developers can take advantage of features opened up by ODR.NET.

ODP.NET has two properties, RowSize and FetchSize, that can be set per command. When a statement is executed, the RowSize value is retrieved, which tells ODP.NET how large each row is in the query. The importance here is that this value is set at run-time. If the query changes or the schema changes, the developer does not have to re-optimize this code. Once you have the RowSize and you know that your end user reads ten rows of data at a time, you set FetchSize = 10 * RowSize. FetchSize can be of any length, you don't have to use RowSize, but most customers do.

The result is that .NET developers using ODP.NET can easily take advantage of the Oracle database using a high performance data provider. The data provider also enables developers to take advantage of database features that make accessing data easier and faster.

Summary and Conclusions

There should be little reason why Visual Studio developers would have significant difficult in employing the Oracle database as the backend to .NET applications, whether they be rich client, Web, or Office applications. The tools contained within ODT enable even developers who have little experience with the Oracle database to use the Visual Studio environment to easily build data-driven applications with Oracle on the back end.

Further, the ability of .NET developers to make use of native data provides with ODP.NET improves performance and enables them to take advantage of database features such as XML DB and RAC connection pooling, as well as specific features in the database such as Ref cursors, configurable result set data retrieval, and array parameters.

Last, ODT.NET provides the ability to easily deploy and debug .NET stored procedures. This eases the ability to test, diagnose, and fix those .NET stored procedures, and to make high-performing stored procedures available for production databases. Anyone using stored procedures with Oracle would make use of this feature to build .NET applications that can execute complex database operations using .NET code.

Overall, the widespread use of the Oracle database in many different types of organizations means that many .NET developers are writing data-driven applications to that database. Because of the potential skills mismatch between .NET developers and Oracle, it may seem like a technical challenge. However, Oracle Development Tools for Visual Studio make it easy to meet that challenge and quickly and easily write those applications. Further, ODT enables database access to achieve the high performance necessary for mission-critical applications. This combination offers the power and features of the Oracle Database within the Visual Studio environment where the developer is most productive.