



## **Improving .NET Application Performance and Reliability with Managed Database Data Connectivity**

*The Role of ADO.NET in Application Performance,  
Scalability, Security and Deployment*

**Stephen Wynkoop**

*Microsoft SQL Server MVP  
SSWUG.ORG Founder*

## Table of Contents

<b>TABLE OF CONTENTS</b> .....	<b>2</b>
<b>INTRODUCTION</b> .....	<b>3</b>
OVERVIEW OF THE ENVIRONMENT.....	3
ADO.NET PROVIDER SELECTION.....	4
BENEFITS OF CHOOSING THE RIGHT ADO.NET PROVIDER.....	4
<b>WHAT IS MANAGED CODE?</b> .....	<b>4</b>
WHAT HAPPENS WITHOUT MANAGED CODE?.....	5
AREN'T ALL PROVIDERS MANAGED CODE?.....	5
<b>THE DATADIRECT ARCHITECTURE</b> .....	<b>6</b>
VERSIONING CONSIDERATIONS.....	7
PERFORMANCE TUNING .....	8
DEVELOPMENT PLATFORM SUPPORT.....	8
INTEROPERABILITY OPTIONS .....	9
PLATFORM SUPPORT .....	10
TABLE 1 - CRITICAL SELECTION FACTORS.....	10
<b>OTHER CONNECTIVITY PRODUCTS</b> .....	<b>12</b>
<b>MORE INFORMATION AVAILABLE</b> .....	<b>12</b>
<b>FREE TRIAL VERSIONS</b> .....	<b>12</b>
<b>ABOUT DATADIRECT</b> .....	<b>12</b>
<b>ABOUT STEPHEN WYNKOOP</b> .....	<b>13</b>

## Introduction

Many factors can impact application performance, scalability, and security in the .NET environment. Developers and system architects frequently look first to the database to improve critical attributes of their applications. However, the key to unlocking the best performance from your .NET application may lie not in the database, but rather in the data connectivity layer that connects your application to your database. This is especially true for .NET applications connecting to Oracle, Sybase or DB2.

Building applications in .NET gives you a couple of different options when it comes to how you implement the database access layer. First, you have the non-managed code providers from many of the major database vendors, and second, you have fully-managed providers, like those available from third party vendors, like DataDirect Technologies (<http://www.datadirect.com>).

The difference between managed and non-managed providers can make a significant difference in many areas, as we'll show in this paper. Application performance, scalability, installation support and other areas are all impacted by the decisions on what provider layer you implement. Informed decisions at design-time can pay huge dividends as your application moves into development, deployment, and ongoing management.

This paper will focus on ADO.NET and how you can best implement it in your environment, what things there are to consider as you select your data connectivity middleware, and how your choices impact the development, deployment, and production-use of your application.

## Overview of the Environment

Building your application makes it necessary to take into account not only the application architecture for the development process, but also the steps to deploy your application, supporting it in the field after it goes into production and how well your application works with other applications that it may be co-existing with on systems where it is installed.

ADO.NET is part of the .NET Framework. Using ADO.NET as your access library means you can use a common programming model from within Visual Basic, C# or J#. There are providers and services as part of ADO.NET – providers are the specific access components for different data stores. For example, if you're working with Oracle, DB2, Sybase or SQL Server, you'll have a choice of providers for each of those. Working through ADO.NET, you'll be able to access those systems using the common library.

Once you have the providers you need for your particular database, you'll reference these in your application project and will be able to connect to, query and work against the database platform of choice.

## ADO.NET Provider Selection

Selecting the right model when setting up your project can be a daunting task, but the payoff can be significant. Not all providers are architected the same way, and choosing wisely at design time can save you time, energy and headaches later.

When you're considering the components to use, you should consider several factors including the database platform you'll be working with, since this will dictate your provider options. You'll also need to consider carefully whether you have requirements that move beyond the basic data access and include things like distributed transactions, multiple provider installations on a common client, and multiple platform interoperability – each of these plays a role in your selection of the best provider for your installation. Be sure to consider carefully the different factors that will be pertinent to your own installations based on the requirements of your systems. The selection table in this whitepaper will help you determine whether your specific situation has requirements that will dictate one approach over another.

## Benefits of Choosing the Right ADO.NET Provider

By using the right ADO.NET providers with your database, you can alleviate the requirement to know specifics of talking to your database and pay more attention to your application. By using key tools and providers with ADO.NET you can further make sure your environment supports your critical requirements (see Table 1).

In addition, careful selection of your provider can mean easier migration between platforms, improved performance tuning abilities and increased reliability and security needs for your application. Further, by using a managed code solution, you can assure the highest level of performance, reliability and security for your applications when working with that provider.

## What is Managed Code?

Managed Code refers to program code that is built to execute on your system and pay attention to the rules of execution. The system is able to make sure appropriate housekeeping is instigated, exception handling is covered, data types and array bounds are honored, etc. In essence, managed code “plays by the rules” and allows for closer integration with the operating environment. Managed code runs completely within the Common Language Runtime (“CLR”) environment, a standardized, execution environment that is part of the .NET framework.

The CLR allows components to more completely and transparently work together, creating a very tight integration between components.

Think of a managed code solution as one that knows what is expected of it, how to work tightly in the environment, and one that can take advantage of these tighter integrations in terms of providing better performance and fewer loose ends that can cause issues for integration, security and the like.

With Oracle, DB2, Sybase and SQL Server 2005 combined with the .NET Framework-based environments, you have the ability to run managed code as part of your application, and, as it relates to this paper, as part of the provider used to work with the database. This is critical

because it affords additional security, performance and tight integration with the systems you're executing against on your system.

There are other advantages as well to this tight integration. Versioning becomes possible, moving away from the issues of DLLs and applications ("DLL Hell"), better memory management and better management of thread pooling.

## What Happens Without Managed Code?

When an unmanaged application runs, the operating system is forced to run it and hope for the best. An image (DLL, exe, etc.) is loaded into memory and execution is passed to it. Without the hooks afforded by managed code (the ability to know what's happening, what's expected and the provision of some visibility into the expected behavior of the application), the operating system is put into a wait and see mode. If something happens, for example out of bounds arrays, memory issues, performance issues, the operating system doesn't really have any recourse or controls to apply.

To address this, the OS will be running the process at what amounts to being arms-length – putting some protections around the executing item. This negotiation of putting things in a box can cause performance issues as the application executes across these boundaries. As demands on the application grow, these performance hits can turn into scalability nightmares.

Finally, as many are familiar, versioning with an executable can be a real challenge. If you've ever looked to swap out a DLL that's been in use, you quickly find that it can present real challenges both with the DLL and the associated registrations that may be needed.

In short, unmanaged code runs "in the wild" and requires a fair bit of running on faith, whereas managed code is able to work tightly with your systems, both at the operating systems level and the database engine level. In addition, deployments can be more complex with unmanaged code, since you cannot take advantage of ClickOnce deployments.

*ClickOnce deployment, formerly No-Touch deployment, is a feature of the Microsoft .NET Framework. This feature allows you to package and install applications for easy distribution to the end-user environment. Applications are installed, then, when the new client accesses the application, the system knows to check the server for any updates. If updates are available, they are downloaded and refreshed for the application's immediate use.*

## Aren't All Providers Managed Code?

While this gets confusing, the short answer is "absolutely not." There is a difference between managed code and 100% managed code. This difference can be the cause for performance/scalability degradation and possible security holes in the application layers. With providers that are not 100% managed, the transition between the provider layer and the database can be an issue.

By starting with a providers that is 100% managed across environments, you can remove the variables and make sure your application environment is fully-optimized and 100% Managed Code, regardless of the target database today and in the future.

## The DataDirect Architecture

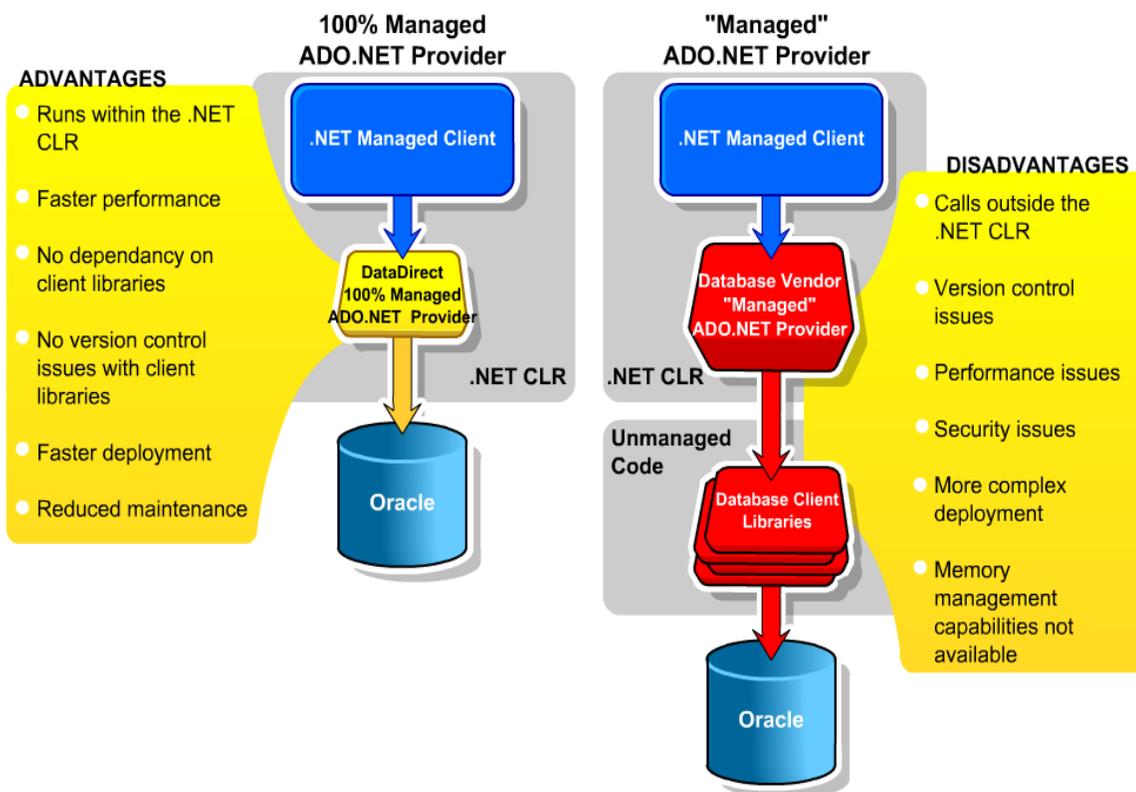
DataDirect's providers run within the .NET CLR and are 100% Managed Code. They are able to work directly with the database at the wire protocol level, assuring best performance, security and all of the benefits of managed code, from versioning to faster deployments and lower maintenance overhead.

By avoiding the switch between a .NET Managed Code Client, to the .NET provider from the database vendor and then to the database client libraries, finally to the database, DataDirect is able to jump, all within Managed Code, from the Client to the database engine.

DataDirect works on a wire protocol level with the database. Essentially this is like to talking to someone in their native language, rather than speaking through an interpreter. By using a wire protocol approach to working with the database, DataDirect providers are able to work directly with the database's native API. This means your applications will be working directly with the database, while at the same time, taking full advantage of the 100% managed code environment. This approach results in faster access and the removal of an entire layer of work that must be done by other provider libraries.

**Wire Protocol** – *this type of API is a very low-level, highly-optimized TCP-based API provided by each database vendor.*

*This is the same API that database vendors native database clients use to communicate to the database. DataDirect has worked with and has the permission of database vendors to use this same wire protocol in the DataDirect providers.*



The DataDirect providers handle the database and make it possible for your application to interoperate across databases and database versions. This “SQL Leveling” functionality removes the database-specific dependencies by providing a consistent interface and set of properties and methods to use in working with the database.

With the wire protocol drivers, you no longer need the client-side drivers for database access. Rolling out versions becomes a non-issue as the providers exist side-by-side, so your users get increased performance while you’re able to maintain an easily deployable application architecture.

### Versioning Considerations

When you install DataDirect providers that work at this low level against the database, you gain the advantage of having well-defined paths to the database. These paths and the wire protocol level access, provide a significant advantage in versioning.

Each DataDirect provider runtime instance operates and controls their connection to the database independently from other DataDirect provider instances. It is possible to host and deploy multiple versions of DataDirect providers permitting “side-by-side” deployment. Thus solutions can be either tightly coupled or loosely coupled to their data connectivity components.

The issues with traditional providers arises from the fact that the different pieces of database access, from the client tools to the database to the ADO.NET providers, are all essentially hard-wired to one-another. This means that, to change one, you must change all, regardless of the impact on other applications that may rely on those same providers. You alleviate this requirement by moving to a 100% managed environment because each application can be using a different, specific provider for access to the database.

This provides you with the ultimate in flexibility not only for deployment, but for support of future updates, application of provider-layer changes on an application-by-application basis and the ability to update or change one application without taking down other applications or requiring wholesale updates and rebuilds of those unrelated, but dependent, applications.

## Performance Tuning

Performance and scalability are greatly improved by using a managed provider as you're not required to jump between managed and unmanaged code as will be the case with other providers as they are forced to make calls outside the CLR.

In addition, DataDirect includes a performance tuning wizard to help establish the parameters that govern your provider's access to the database. The tuning wizard will give you the connection string you'll need to optimize your connection based on query sizes, connection habits and more. The wizard is standard with the DataDirect providers.

*For more information on performance, be sure to check out information about the DataDirect Performance Wizards:*

*<http://www.datadirect.com/products/net>*

DataDirect has a significant experience base working with providers and building out uncompromising solutions for connectivity and performance. DataDirect has a heritage of ODBC and JDBC products, even to the point of helping develop specifications for libraries for major database providers. The majority of large independent software vendors embed DataDirect products in their application. These efforts bring a unique advantage to the DataDirect providers – you gain the leverage that comes from insight into the inner workings of the database and the long-standing experience building highly-scalable, secure and performance tuned solutions.

Finally, by using the managed code environment and providing for the ultimate in flexibility for your deployments by allowing multiple versions, ease of support and updates, and configuration flexibility on the desktop, you assure supportability in the future, while preserving your options for security and performance along the way.

## Development Platform Support

The DataDirect managed providers seamlessly integrate with Visual Studio 2005 so developers do not have to leave their familiar working environment. This provides you with more leverage from within the Visual Studio environment as you build out your applications and gives you the

cross-platform speed, security and scalability tools you need for your application development process.

The data providers have built-in tracing to work with debugging and structured development cycles. You can trace input arguments as well as outputs and returns to/from method calls.

*More information on performance monitor counters to reference is available here:*

*<http://www.datadirect.com/products/net>*

Here's a brief summary of the DataDirect Provider developer-specific features:

The developer support features are always standard with DataDirect providers. **64-bit .NET Framework Support** – Applications that need the performance and scalability provided by 64-bit platforms can deploy DataDirect Connect for ADO.NET.

**Conformance to the ADO.NET 2.0 specification** – Support for the latest version of the data access API included in the Microsoft .NET Framework 2.0 makes it easier to support a heterogeneous database environment.

**Side by Side Deployment** – Ability to deploy applications side by side with different versions of the provider.

**Visual Studio 2005 Developer Aids** -- Seamless integration with Visual Studio 2005 using visual development aids reduces development costs.

**ClickOnce Deployments** –Support for ClickOnce deployment, introduced in the .NET Framework 2.0. This means that DataDirect's provider can be deployed alongside your application code, reducing deployment costs.

**Database Vendor Parameter Binding & Syntax** – In addition to standardized parameter markers, DataDirect now supports database vendor syntax. This capability allows organizations to realize the benefits of DataDirect ADO.NET connectivity without having to modify their provider-specific SQL code.

**DataAdapter Batch Update Support** – The DataDirect providers now support the ability to update DataSets using configurable batches. Efficiencies gained by batch update support results in better performance and through-put.

## Interoperability Options

Building applications that work between database systems requires being able to easily and quickly move between platforms without sacrificing speed, security and accessibility to your application. With DataDirect providers, you gain access to a standardized library of calls, platform-specific methods and properties and all provided in a common interface so you'll be able to move around quickly and easily within the different platforms you support.

DataDirect provides for SQL Leveling, a technology built to make it easier to work against different platforms without having to know the specific and particular syntax for any given platform. You'll be able to use the common interface in Visual Studio to access the properties and methods in the same manner – all exposed in the Visual Studio environments tree-based architecture.

SQL Leveling lets you use a common command interface to access multiple platforms, without having to know the different command structures and syntaxes for working with the specific platforms. Using this SQL Leveling capability, your application code can be written and executed against all of the major database vendors, without recoding and without regard to specific SQL capabilities implemented by the database vendor.

## Platform Support

DataDirect provider support most popular enterprise-class databases on the market. These include IBM DB2, Oracle, Microsoft SQL Server and Sybase. In addition, any single DataDirect provider will work across all versions of a given database (e.g.: Oracle 8, 9 and 10). Please check the DataDirect site (<http://www.DataDirect.com/products/net/matrix/dotnetmatrix.htm>) for information on specific versions and other platforms as they become available.

### Table 1 - Critical Selection Factors

ADO.NET Provider Feature	DataDirect Connect for ADO.NET	Competing .NET Data Providers
<b>100% Managed Code</b>	DataDirect Providers are 100% managed. DataDirect ensures that all data processing is done within the CLR, providing better performance, scalability, security and reliability	<p>No other .NET data provider is 100% managed. Alternatives that are not 100% managed require calls outside the CLR; degrading performance and creating instability and compromised performance due to memory leaks.</p> <p>The result is a heightened risk of security breaches as well as data connectivity failure.</p>

**No dependency on native database client**

DataDirect's ADO.NET data providers communicate directly with the database through wire level protocol, eliminating the need for the native database client.

As a result, deployment costs and installation upgrades are effectively eliminated, as are conflicts between client versions.

Competing .NET data providers require native database client installations, adding substantial cost to deployment and maintenance, and increasing the likelihood of client version conflicts.

Additionally, this approach adds substantial cost and complexity to testing and Quality Assurance efforts each time clients are upgraded.

**Comprehensive interoperability between database vendors and versions**

DataDirect's ADO.NET data providers offer SQL leveling capability that ensures commonality across database vendors (Oracle, DB2, Sybase and SQL Server) and between vendor database versions (Oracle 8i, 9i, and 10g, for example).

As a result, development, deployment and Quality Assurance efforts are streamlined and application flexibility is greatly increased.

Competing .NET data providers offer fewer features and functionality.

Distributed transaction support, load balancing, connection retry, failover and connection pooling, for example, are not supported by other .NET data providers.

**Superior performance and scalability**

DataDirect Connect *for* ADO.NET offers architectural advantages that lead to superior performance and improved scalability.

DataDirect's providers have been proven under stringent testing and benchmarking procedures that have shown significantly improved performance (50% to 350%) under load, while demanding fewer server and network resources.

Competing .NET data providers lack the performance and scalability capabilities to perform under load intensive, mission critical data connectivity environments.

As a result, organizations are forced to compensate for driver shortcomings by installing additional servers dedicated to hosting applications, increasing overall cost and complexity of deployment.

## Other Connectivity Products

**DataDirect Connect for ODBC** – DataDirect Connect for ODBC is the fastest, most comprehensive suite of ODBC drivers for all major databases. The unique wire protocol design of DataDirect's ODBC driver eliminates the need for database client software and libraries, which greatly simplifies installation and administration and dramatically improves performance.

**DataDirect Connect for JDBC** – DataDirect Connect for JDBC is the fastest and most comprehensive suite of Type 4 JDBC drivers for all major databases. DataDirect Connect for JDBC is the SPECjAppServer/ECPerf performance and scalability leader, and supports advanced functionality such as distributed transactions, connection pooling, updating BLOB/CLOB data types, and Windows Authentication for Microsoft SQL Server and Oracle. DataDirect Connect for JDBC consistently supports the latest database features and is fully compliant with J2SE 5.0 and JDBC 3.0.

**DataDirect for SequeLink** – DataDirect SequeLink is a complete end-to-end middleware solution that provides a universal client for ODBC, JDBC, and ADO.NET. SequeLink is the best choice if your application needs access to three or more data stores.

**DataDirect OpenAccess SDK** – DataDirect OpenAccess SDK can be used to quickly build custom ODBC, JDBC, ADO.NET and OLEDB Drivers for specialized data stores.

## More Information Available

For additional details on different areas in this whitepaper and the specifics of implementation, deeper technical information and other references, please visit <http://www.datadirect.com/products/net>.

From here, you'll be able to quickly jump to additional technical references and deeper technical detail on many of the points outlined here.

## Free Trial Versions

A 15-day free evaluation copy of DataDirect Connection *for* ADO.NET is available at <http://www.datadirect.com/downloads/registration/connect/net>.

## About DataDirect

DataDirect Technologies is the most comprehensive data connectivity vendor. We have components for every major database, platform and data access API. We are fully committed to database independence with products that are optimized for interoperability. We focus 100% of our resources on data connectivity. Many of our customers like the fact that they can use one vendor for all their needs, leaving the core competency of data access to us so they can focus on the application or product they sell.

Get more information, or download your trial for DataDirect providers on the web at <http://www.datadirect.com/products/net>.

## About Stephen Wynkoop

Stephen Wynkoop is the founder of SSWUG.ORG (<http://www.sswug.org>) and a Microsoft SQL Server MVP. He writes a daily column on the database industry for SSWUG and is also host of a weekly SSWUGtv show on the site. He's also a frequent presenter at conferences and produces webcasts several times monthly for SSWUG. He can be reached at [swynk@sswug.org](mailto:swynk@sswug.org).