

Why developers should use ODBC instead of native proprietary database interfaces

The financial and technical basis for using ODBC with wire protocol drivers instead of native database interfaces in both new and existing applications

Introduction

When writing applications in C, C++, VB, PowerBuilder, Delphi, Cold Fusion, etc., programmers and architects have two options for handling the interface to data. They can either develop unique, non-portable code to talk to the native proprietary interface of each database that their program requires, or they can write one set of code to the ODBC standard interface and use ODBC drivers to connect from the interface to each database.

The ODBC interface has always been understood to bring great advantages in the areas of interoperability, developer productivity, and support. It has become a widely adopted standard with thousands of commercial software products and tens of thousands of custom corporate applications. Yet despite its clear benefits, many developers and architects don't use ODBC, primarily because they perceive that its benefits come at the expense of application performance. The ODBC performance penalty is a myth that was dispelled a long time ago, yet people continue to believe it.

Wire protocol technology actually makes ODBC faster than native

DataDirect Technologies was the first commercial vendor to produce ODBC drivers and we continue to be the industry leader. DataDirect Technologies has recently revolutionized the ODBC driver design with the introduction of a suite of drivers based on wire protocol technology. Wire protocol drivers do not need database client libraries to work. They speak to the database directly via its wire protocol. Not requiring the client library is a major advantage. It makes applications much easier to develop and deploy, and reduces management demands on clients. It also dramatically improves application performance, so much so *that applications using ODBC with wire protocol drivers have actually been proven to perform faster than applications written to the database native interface.*

This paper lays out the argument for why using ODBC with wire protocol drivers is a better option than programming in each database's native language for new applications, and why removing native interfaces in existing applications in favor of ODBC and wire protocol drivers is a good investment of time and resources.

Writing new applications to an ODBC interface with wire protocol drivers significantly reduces development time and leads to better performance

ODBC's flexibility makes it a better programming choice in almost every situation

The choice is up to you. ODBC has clear benefits. Developers do not need to learn the intimate details of each database they work with. This means they can be more productive and you can avoid having specialized skills sets.

When people tell us they don't want to use ODBC, the performance myth is the usual reason stated, but it is not the only answer we get. It appears there are other myths floating around about ODBC that need to be dispelled. Here are some key points about ODBC that you should be aware of.

- ODBC is still an actively supported standard and is not "old technology." In fact, Microsoft (ODBC's promoter) will be coming out with a new revision to the standard at the same time they introduce the next version of Microsoft SQL Server.
- ODBC can take advantage of database-specific functions by use of ODBC pass-through mode or ODBC escape sequences.
- ODBC works well on all databases, including DB2, Oracle, Sybase, Progress, Informix and more, not just Microsoft SQL Server.
- ODBC is not a Windows-only API. It can be used on a variety of platforms. DataDirect Technologies produces ODBC drivers for Windows, HP-UX, Linux, AIX, Solaris, Mac, Tru64, and IRIX. (Click [here](#) for a full listing of all the databases and platforms we support.)

In fact, the only "limitation" ODBC has is that it is a relational API and can only work with data types that can be expressed in rectangular or two-dimensional format. (For example, it will not work with data types like Oracle's spatial data type.) If your data isn't relational then you can't as easily use ODBC.

Wire protocol drivers eliminate the need for database client libraries

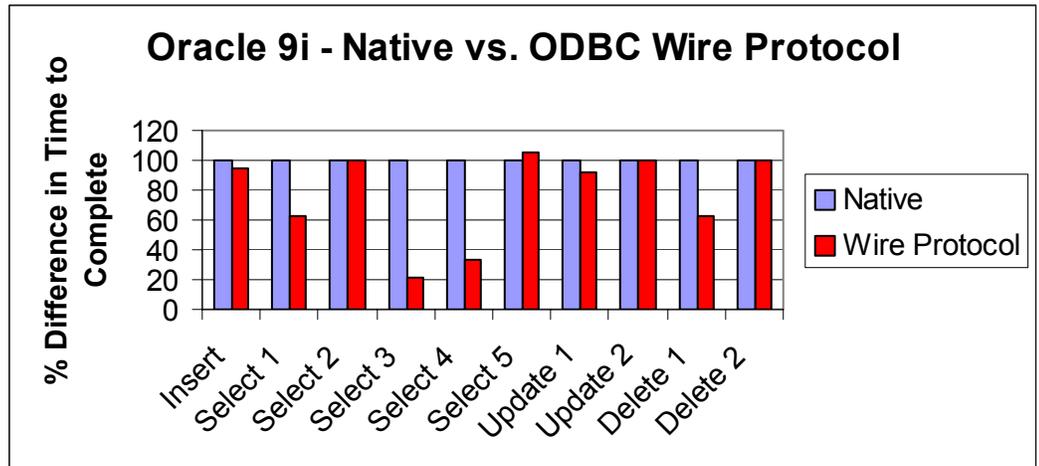
The database client libraries are large files that need to be installed on every client that is running a database application... until now. DataDirect Technologies' wire protocol technology eliminates the need for the client library. Now you can deploy your applications without requiring users to make sure they have the correct version of the client library on their system. This dramatically simplifies deployment and management of your applications.

Wire protocol drivers actually *outperform* coding to the database's native interface

A new independent study from database expert Ken North has proven that in most situations applications written with DataDirect Technologies' wire

protocol ODBC drivers actually outperform native programming by a wide margin. Ken's tests consist of two sets of runs on the exact same hardware and software with the same application. The only variable is the interface. For one set of runs the application uses a proprietary native database interface, for the other it uses an ODBC interface with wire protocol drivers. (Click [here](#) to obtain a complete copy of Ken's report.)

The results are very impressive and show that ODBC with wire protocol technology almost always beats native performance, and usually does so by a wide margin. For example, please see the graphic below comparing Oracle 9i performance with ODBC wire protocol vs. native.



This bar chart graphs the percentage difference in performance between native and ODBC. For example, the “Select 3” statement shows that ODBC wire protocol outperforms native by 80%, i.e., the smaller the bar, the better the performance.

This graphic shows the results for ten different tests covering a range of INSERT, SELECT, UPDATE, and DELETE statements with the performance of the native interface set to 100 in all cases to give a standard view of relative performance. In only one case is the native interface faster (by about five percent) while in six of the ten ODBC/wire protocol is clearly faster (often by 40% to 80%!) And ODBC wire protocol performs best in the area of SELECTS, where most applications do their work. Imagine getting your existing application to perform tasks in one-half to one-quarter of the time they used to, just by switching to an open standard data interface, which has numerous benefits in and of itself.

How is this possible? Because our wire protocol technology is built from the ground up with a streamlined design that is optimized for network performance. For example, our Oracle driver uses compressed data packets that are allowed by the wire protocol that our drivers use, but not by the wire protocol that Oracle Call Interface (OCI) uses. When fetching large amounts of data, an analysis of the sheer numbers of network packets used by our drivers vs. OCI generally shows significantly fewer packets transmitted

across the wire with our drivers. Less traffic across the wire means better performance.

A key part of our design was to eliminate the need for the database client libraries. Not only does this mean you don't have to install and maintain these large pieces of software, your data doesn't need to go through the extra performance-inhibiting elements such as double buffering and extra processing that are inherent in the client library architecture. Our drivers only need to support the functionality necessary for an ODBC driver. We don't need all of the overhead necessary to support the client.

Don't even think about native interfaces for new applications!

The evidence laid out here creates a compelling argument for always using ODBC with wire protocol drivers instead of native interfaces whenever your applications are accessing relational data. With ODBC and wire protocol drivers, you get:

- Better developer productivity by not requiring that your developers learn the intimate details of each databases' interfaces.
- Full access to all of the native database functions via ODBC pass-through mode or escape sequences.
- *Better* performance than native.

This is one of those rare cases where you can have your cake and eat it too. ODBC with wire protocol technology can deliver the absolute best performance and flexibility.

Replacing native interfaces in existing applications with wire-protocol ODBC drivers can yield an attractive financial return

While the case for new applications is straightforward, the case for replacing native interfaces in stable, existing applications with ODBC and wire protocol drivers may not be as readily apparent. But there are compelling reasons to do so.

Your support costs will be lower

With native interfaces you need different development and support experts for each of the different databases your applications connect to, and you'll have to update your applications every time there are changes to the database versions and database client libraries. Wire protocol drivers eliminate the need for the client libraries and eliminate programming changes for every new database version.

Covering additional databases is very easy

Having an ODBC interface will also make it very easy to extend the database coverage of your application. When you want to support a new database, all you need to do is get a wire protocol driver for the new database. If you are a commercial software vendor, this can open a broader market and new revenue for your existing offerings.

Your application performance will be better

Faster performing applications translate into improved competitive position and cost savings, no matter if you're creating software to license to others or for your company's internal use. Performance improvements are something that your users notice, driving broader deployments and add-on sales.

There are very few options to improve the performance of a maturing application. Generally this has to come via a technical innovation in software or hardware. Wire protocol ODBC technology is just such a breakthrough and allows you to improve performance while reducing costs at the same time.

Lower costs and better performance contribute to an attractive ROI

Every situation is slightly different so each user has to create their own analysis of the financial payback from removing native interfaces and replacing them with a wire protocol ODBC interface. But the arguments laid out here should make the financial return compelling in all but the smallest applications. With ODBC and wire protocol drivers you get improved performance and reduced costs. The only "investment" you need to make is building the ODBC interface and licensing the wire protocol technology. Our company would be happy to help you create this kind of analysis.

Conclusion

The mythical ODBC performance penalty has now been officially dispelled. With ODBC and wire protocol drivers from DataDirect Technologies you can now get all the benefits of ODBC, all the functionality of native programming, and better performance. There is now no reason to build a new application with a native interface, and in most cases you'll get an attractive return when you rebuild your existing applications to work with ODBC and wire protocol drivers.

At DataDirect Technologies, our sole focus is on building great data connectivity components. We were the first commercial software vendor to introduce a line of ODBC drivers and we've led the market ever since. Wire protocol technology is just one example of the kind of innovation we're known for across the industry. DataDirect Technologies can provide you with a complete line of ODBC wire protocol drivers to access all the major databases on all the major platforms, allowing you to get the best technology on the market from just one vendor. Over 270 ISV customers like webMethods, IBM, Oracle, Software AG, Hyperion, Manugistics, and over 4,000 corporate customers like CSFB, Avon, and Northwestern Mutual all consider the cost of our technology a small investment that provides great returns in many ways.

FOR MORE INFORMATION

800-876-3101

ddinfo@datadirect-technologies.com

Worldwide Sales

Belgium

(French Belgium)..... 0800 12 045
(Dutch Belgium) 0800 12 046
(Netherlands) 0800 022 0524

France 0800 911 454

Germany 0800 181 78 76

Japan 0120.20.9613

United Kingdom 0800 169 19 07

United States 800 876 3101

© 2002 DataDirect Technologies, Inc. All rights reserved.
DataDirect Connect is a registered trademark of DataDirect Technologies. Java and JDBC are registered trademarks of Sun Microsystems, Inc. Other company or product names mentioned herein may be trademarks or registered trademarks of their respective companies. 07/02



DataDirect Technologies is the leading provider of components for connecting software to data, providing standards-based technology that ensures consistent behavior and performance across diverse environments such as J2EE, .NET, Web, and client/server. With the most comprehensive support for ODBC, JDBC, ADO and XML, DataDirect Technologies provides the easiest experience connecting software to data.

datadirect-technologies.com