insightsoftware

Wire Protocol ODBC Drivers and ODBC SDK

Version 10.3

October 2024

Copyright

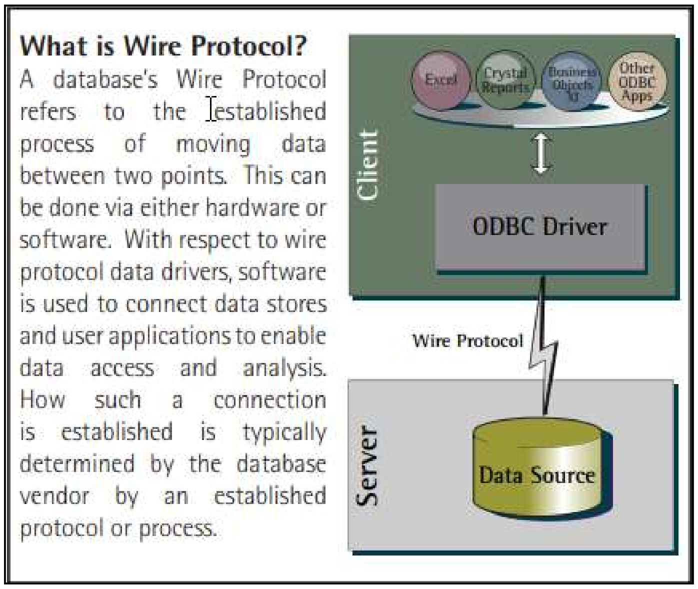
Contact Us

Magnitude Software, Inc.

www.magnitude.com

Wire Protocol ODBC Drivers and ODBC SDK

An ODBC connectivity question commonly pondered is what's the difference between a third party wire protocol ODBC driver and an ODBC driver from a database vendor? The short answer is that there isn't a lot of difference. The real advantage of ODBC drivers from a third party vendor is that in a heterogeneous environment, third party ODBC drivers can insulate from differences between database vendor distributed ODBC drivers. However, this only works when the third party driver vendor has its own ODBC SDK and it uses this ODBC SDK to build all of its drivers. Some driver vendors have ODBC drivers and an ODBC SDK, but for whatever reason, they don't build their ODBC drivers using their own SDK technology. Let us explain. ODBC is a database standard that was introduced by Microsoft in the early 1990's and has remained fairly stable. ODBC 3.8 is the latest version of the standard, having been introduced in 2009.



What is Wire Protocol?

A database's Wire Protocol refers to the established process of moving data between two points. This can be done via either hardware or software. With respect to wire protocol data drivers, software is used to connect data stores and user applications to enable data access and analysis. How such a connection is established is typically determined by the database vendor by an established protocol or process.

The usual access pattern to a database is as follows:

- The database exists on a server in a network
- A client machine using an application like Microsoft Excel or SAP Business Objects' Crystal Reports accesses the database
- There is an ODBC driver on the client machine that Excel or Crystal Reports connects to
- The ODBC driver connects to the database via a wire protocol

When we look at the wire protocol used by the ODBC driver, there are two options:

- Use the wire protocol defined by the database vendor (Figure 1)

- Use a proprietary wire protocol defined by a third party (Figure 2)

In option 1 above, the wire protocol is standard to the database, so there is nothing to be installed on the server. In option 2, the wire protocol is from someone other than the database vendor, so it will need something installed on the database server.

The vast majority of ODBC drivers today use the wire protocol that is standard to the database, as in option 1.

However, if different ODBC drivers use the wire protocol defined by the database vendor, what then is the difference in the ODBC drivers? The answer is that there isn't a lot of difference. For example, if you have ODBC driver A that comes from the database vendor and ODBC driver B that comes from a third party and they both use the wire protocol defined by the database vendor, then in terms of their performance across the wire, they should be identical. There is no difference in query performance since both queries execute on the same database server.

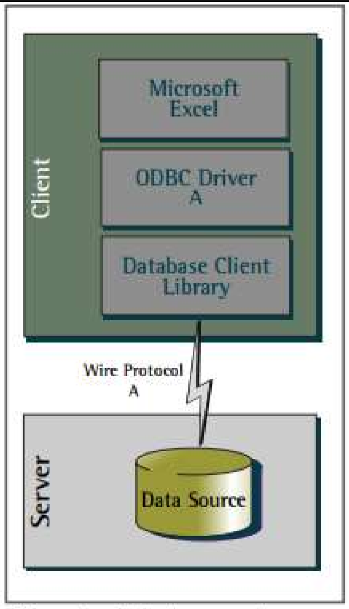


Figure 1 - Database vendor ODBC wire protocol utilizing Database Client Library

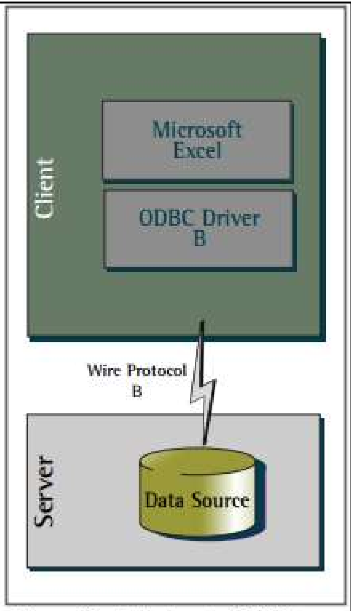


Figure 2 - Third party ODBC wire protocol

Where you may notice a difference is how the ODBC driver utilizes the database client libraries. For example, you could have ODBC driver A that uses the client libraries that come from the database vendor and ODBC driver B that writes directly to the wire. Here, there could be a perceivable difference, and more likely ODBC driver A is better.

In the above example, the reason ODBC driver A may be better is that most database vendors do not publish details of their wire protocols. As such, anyone that wants to build an ODBC driver for the database is told to use the database client library. One has to figure that since the database wire protocol was developed by the database vendor, the vendor's own client libraries would be most optimized to use that wire protocol.

If you wanted to write directly to the wire without the database client libraries, as in the example with ODBC driver B, you would have to reverse engineer the wire protocol. However, you would have to hope that the database vendor didn't change anything because if they did, your ODBC driver would likely end up with critical issues.

Some people say that it is a lot of work to configure a database client library, or that to do so is extra effort. Actually, if your ODBC driver was built with the database client library, the ODBC driver probably already includes the database client library within it as a redistributable component. In fact, many people don't know this. Therefore, it wouldn't be necessary to configure it.

Yes, a wire protocol ODBC driver is a good thing. Any ODBC driver that does not require a special server-side database component is using the same wire protocol defined by the database vendor. By using the client libraries from the database vendor for an ODBC driver, you are best assured that the database vendor has optimized the client library for the database wire protocol. In addition, when using the client library, there is much less risk of the ODBC driver becoming inoperative due to vendor wire protocol changes. Using the database client library insulates you from wire protocol changes you would otherwise be exposed to, if reverse engineering the wire protocol formed the basis of the ODBC driver's development.

Ultimately, the biggest advantage of using an ODBC driver from a third party vendor stems from the third party's use of a common framework to build its ODBC drivers. With a common framework, one knows that ODBC drivers for various different databases will function similarly. This adds insulation from the myriad of database differences that make configuration difficult and ultimately cost time and money.

Many ODBC driver vendors have ODBC SDKs that allow development of custom ODBC drivers. If the ODBC driver vendor uses its own ODBC SDK to build its drivers, then you know it has a common ODBC platform across different databases. It's important to ask an ODBC driver vendor if it builds its ODBC drivers with its own ODBC SDK.

Contact Us

If you have difficulty using this product, please contact our Technical Support staff. We welcome your questions, comments, and feature requests.

Note:

To help us assist you, prior to contacting Technical Support please prepare a detailed summary of the Wire Protocol ODBC Drivers and ODBC SDK version and development platform that you are using.

You can contact Technical Support via the Magnitude Support Community at http://magnitudesoftware.com/online-support/.

You can also follow us on Twitter @SimbaTech and @Mag_SW

Third-Party Trademarks