

Why Open Source Database Drivers Are Not Viable

Introduction

Open source software is gaining traction in many software infrastructure markets, particularly in the operating system, Web/application server and database server infrastructure areas. Many organizations currently are leveraging open source components or are devising an open source acquisition and management plan. In most cases, organizations are adopting a blended-source approach that leverages both open source and traditionally licensed software. It is not uncommon to see environments that leverage a commercial database, such as Oracle, running on Linux, an open source operating system. This phenomenon is accelerated by interoperability support for open source components provided by traditionally licensed software vendors.

DataDirect Technologies has a proven track record in blended-source deployments. DataDirect's database middleware products are pervasively deployed in production environments, including many scenarios that include open source components.

Successful infrastructure-level open source solutions, such as Apache and Linux, enjoy two distinguishing characteristics that make them suitable for enterprise deployments of critical applications. First, they are supported by a large, vibrant community of developers who ensure robust feature support, solid reliability, and enterprise-class performance and scalability. Second, these infrastructure-level solutions are backed by a consortium (Apache Software Foundation) or commercial venture (RedHat, JBoss) that provides the financial backing and management infrastructure to sustain ongoing research and development, as well as technical and legal support. Projects lacking a vibrant developer community and commercial support are just that — technology projects. Although they may be suitable for certain non-critical use cases, these projects lack the characteristics that make a technology suitable for use with business-critical production systems. Literally, thousands of these projects are available on the SourceForge open source portal — most are developed and maintained by one or two part-time developers.

Although many major infrastructure components (for example, operating systems and Web/application servers) enjoy the critical developer mass and financial/organizational support necessary for success, open source efforts for more modest solutions, such as database drivers, do not use modern development methodologies and quality assurance, and lack organized support operations. Organizations that use an open source database driver assume extraordinary risk because of inferior product quality, non-existent technical support, and the complete assumption of legal liability. This article

contrasts the organizational approach used by open source database driver providers with DataDirect's industry-leading, database connectivity solutions, which are production viable and commercially supported. After reading this article, it will be apparent that the product quality, technical support, and legal liability risks associated with open source database drivers far outweigh any initial cost advantages associated with the open source option.

Open Source Licensing Complexity

Open source software is perceived to be free, but it should not be confused with public domain software. Similar to traditionally licensed software, usage of open source software is limited by a set of restrictions that are imposed by a license agreement. Unlike traditionally licensed software, open source license agreements are written to protect the licensor (for example, code contributor); they are not written to provide legal liability protection for the licensee (for example, the organization using the software). Open source license agreements do not include intellectual property protections, fitness clauses, or indemnification to protect the licensee. The agreements simply provide a vehicle for propagation of the license itself in the licensee's application, and are silent with respect to indemnification, which intentionally benefits the licensor/contributor. It is imperative that the corporate user of open source products is familiar with the restrictions and legal risks associated with the open source solution.

Open source licensing is greatly complicated by various license types, as well as ambiguous language in the license. The Open Source Initiative (OSI) lists 58 approved license types including the GPL, LGPL, BSD, MIT, and Mozilla licenses. Each of these licenses carries a different set of restrictions and obligations — factors that could ultimately determine whether a proprietary or “home grown” software application is subject to release under an open source license agreement. Corporate users of open source software must trust their IT staff to make the correct technical and legal decisions so that the organization does not violate the license agreement. In most cases, corporate developers are completely unaware of open source license obligations and leverage open source software without consulting their legal department. Even if a corporation's legal organization gets involved, the costs involved in the legal review of every open source contract can be considerable. These license agreements often are written in ambiguous language. Legal interpretations vary greatly in terms of the user's obligations, and the courts are just now starting to adjudicate open source license disputes.

Open Source Models

In addition to license complexity, another important factor when considering using open source software is the business or organizational model used by the software project. Initially, most open source projects are managed by a handful of developers in an autonomous manner. Although this approach can lead to a strong sense of community, it does not provide the level of development, testing, and product-release rigor to ensure enterprise-class reliability, performance, and scalability. To address these shortcomings, many larger, infrastructure-level projects have turned to consortiums or commercial ventures to provide the resources for enterprise-class deployments. Many of these organizations are now household names and include RedHat Linux, JBoss Application Server, as well as the Apache Software Foundation (Apache Web Server).

Although commercial ventures leverage a variety of business models (RedHat relies on license revenue, and JBoss sells training/customer support), they all provide the financial, management, development, and technical support resources to ensure that their open source products are viable for use in production deployments. They are careful to distinguish their product offerings from non-commercial, project-based, open source solutions. In fact, JBoss has trademarked the term Professional Open Source™ and explicitly states that they “hire and pay experts in the open source community to write exceptional and innovative software full-time” to ensure that their solution is “the safe choice for end-user enterprises and independent software vendors alike.” These vendors understand the difficulties and risks associated with open source solutions that lack commercial vendor support and are very deliberate about explaining how their approach is different.

Project-based open source solutions, such as those associated with database driver projects, lack the financial, management, development, and technical support resources for production deployments. The remainder of this article addresses the product-quality issues, unreliable technical support, and legal risk associated with using an open source database driver.

Open Source Risk Means Business Disruption

Organizations that rely on open source database drivers are assuming risks that can easily disrupt their business practices. A combination of legal, functional, and support issues can cause the following disruptions:

- **Application downtime and data integrity problems** — Technical problems with the driver may affect application availability and lead to data integrity issues. Depending on the nature of the application, critical users and business processes may be affected. In addition, the cost of

addressing data corruption issues can be significant. Because the open source developer does not offer support, your own IT staff must diagnose and fix any technical issues, often at a significant cost in disrupted projects and other essential activities.

- **Technology replacement cost** — If an organization is forced to replace an open source database driver because of technical or legal reasons, the cost of that replacement can be substantial. The cost to re-develop, re-test, and re-deploy applications is an important consideration. Given the highly integrated nature of today's applications, the impact typically is not isolated to an individual application, which further drives up the replacement cost.
- **Management/legal disruption** — Even if an intellectual property issue does not make its way to court, the potential distraction from an executive-management and legal perspective can be significant. Valuable time that should be dedicated to running the core business is expended on arcane "legalese."
- **Customers/supplier impact** — If an application is used to support customers or partners, the potential for disruption is not limited to the internal organization. Application downtime and data corruption can have a profound affect on partners and customers.

Perhaps the most important question facing an organization that is considering the use of an open source driver is whether they want to expend resources on implementing, supporting, and mitigating the risks associated with the open source project rather than focus their resources on projects that are core to the financial success of their business.

Product Quality

Open source database driver projects lack the financial support and dedicated developer and QA resources to ensure the high quality and rich functionality for success in production-critical environments. Lack of resources results in the following shortcomings:

- **Limited product breadth** — Open source database drivers lack the product breadth offered by DataDirect. Because open source database drivers typically are designed around a single wire protocol, database support is limited. This negates the interoperability advantages provided by a single database driver solution that can support multiple databases. In the case of JDBC, open source database driver solutions fall short in terms of support for JDBC 3.0 features. It is not unusual to see situations where the open source driver implements functionality using proprietary methods, which hinders interoperability and increases switching costs. In addition, it is common for features that are purportedly supported to not work correctly because they are implemented using a stub interface, return code-generated exceptions, or result in application errors

(including data corruption). DataDirect provides many features that complement the database driver specifications, including developer productivity tools such as DataDirect Test™, DataDirect Spy™ and the Performance Tuning Wizards. Comparable functionality such as that provided by DataDirect is not available in open source drivers, including functionality for distributed transactions, failover, load balancing, Single Sign-On, connection pooling, and internationalization support.

- **Unreliable product quality** — Open source database projects lack the development and testing rigor to ensure production-level quality. Compared to the comprehensive test suite developed by DataDirect, open source database drivers test runs are rudimentary. In fact, it is unusual for an open source project to provide any level of transparency into the testing process, much less document or automate their testing strategy. In fact, driver testing may be nonexistent.
- **Limited product roadmap** — DataDirect maintains a development plan of record that provides visibility into the product roadmap. This plan of record is carefully designed to incorporate support for new database driver specifications, new database releases, and customer enhancements. Open source projects are unable to provide visibility into their product roadmap because their roadmap is nonexistent. For the most part, functionality is added randomly, and features can be reworked on the whim of a single developer. This makes it difficult, if not impossible, for an organization relying on an open source database driver to accurately plan their own release schedule.
- **Poor release planning** — Because reliable roadmaps for open source products do not exist, open source projects cannot provide a reliable release mechanism. Typical release planning involves a developer posting a message stating that they feel it is time for a release. This may or may not generate some discussion amongst the team, but invariably, the release happens several days later. The open source project provides no indication of how or whether final tests were executed, how the code base was locked down, and so on. Once the release is available, a fair amount of developer chatter relating to integration issues, memory leaks, and so on can be expected. This process is in stark contrast to the DataDirect methodology, which is time-proven, robust, and has been audited extensively by several of the world's largest independent software vendors.
- **No standards participation or visibility into database wire protocol** — Open source projects are forced to reverse engineer the design of their drivers because they do not have direct knowledge of database wire protocols. In addition, because they do not participate in the database driver standards process, they are relegated to working with the specification only once it has been published. They lack the interaction with the specification designers that is critical to designing highly performing, highly optimized drivers. One could argue that these factors also increase the risk that an open source contributor could

implement code that violates intellectual property or software patents. DataDirect participates directly and often plays a leadership role in the JDBC, ODBC, and ANSI SQL committees. In addition, DataDirect has partnered directly with the database vendors and enjoys direct access not only to the native wire protocols, but also to the database architects responsible for the protocols. This level of participation in the standards process and protocol design is a major reason that DataDirect's drivers are unrivaled in the database middleware market.

- **Limited certification** — Open source projects lack the development, testing, and financial resources to correctly certify their database drivers. In the case of JDBC, open source solutions may claim to be certified, only to fail the CVS test suite. In other cases, they publicly call for financial donations to finance additional development required to pass the certification process. DataDirect is consistently the first vendor that certifies its drivers based on the latest specification. Currently, the DataDirect Connect *for* JDBC driver is J2EE certified using the CTS for J2EE 1.2, 1.3, and 1.4. This level of certification ensures the highest level of quality and reliability, eliminating costly development issues and application errors.
- **Undocumented product** — Open source database drivers do not include product documentation with their software. DataDirect provides extensive product documentation with its products. In addition, a KnowledgeBase, consisting of over 1,800 articles on database middleware topics, is available online. In many organizations, product documentation is needed for initial deployment, basic troubleshooting, and developer training. Products lacking documentation result in higher development and deployment costs.
- **Hidden product risks** — Organizations that leverage an open source driver are subject to hidden risks that can impact their use of the product. Open source projects typically rely on one or two key developers. These projects do not have contingency plans to compensate for the loss of a key developer. For example, suppose a key developer can no longer work on the project because of other work commitments (not unlikely given the lack of financial incentive for open source projects). The entire project can be crippled, leading to significant risk for the organization using the open source driver. If an organization is willing to assume the risks associated with an open source driver, it also should carefully consider the costs of migrating to another solution if application errors caused by the driver become significant. Given that database driver implementations vary and that open source drivers rely on functionality that deviates from the specification, organizations using open source solutions face costs associated with development, testing, source code management, and (re-)deployment.

As is readily apparent, the product abilities provided by a stable, well-financed organization, such as DataDirect, significantly outclass those abilities provided by a small, open source project. What may not be apparent

is the detrimental impact that using an open source driver can have on an organization because of its product shortcomings. The following table clearly states the advantages provided by DataDirect and the risks associated with using an open source database driver.

DataDirect Database Drivers	Open Source Database Drivers	Customer Impact
Massive product breadth based on complete support for the specification, robust support for optional features, and extensive support for abilities that complement the specification.	Not all features in the database driver specification are supported; some that are supported leverage proprietary extensions or result in errors and application failure.	Limited ability restricts application functionality, forcing organizations to build their own functionality (increasing development costs) or sacrifice user features (reducing end-user satisfaction).
Unrivaled product quality based on an extensive certification process, test methodology, and transparent/audited processes.	Suspect product quality resulting from lack of resources and development methodologies.	Suspect driver quality can greatly impact applications, leading to costly downtime, data corruption, and extensive troubleshooting and re-deployment costs.
Specification leader and strategic partner to the database vendors.	No role in the specification process. Forced to build their drivers by reverse engineering database protocol.	Sub-optimal driver design because of limited visibility into the specification process and a reverse-engineered protocol design.

Legal Risk

Commercial software providers of open source and traditionally licensed software provide legal indemnification and quality warranties as part of their standard product offering. In addition, these commercial ventures institute a rigorous methodology that greatly reduces the likelihood of plagiarized code or patent-related violations. These benefits provide a level of assurance for organizations using these products. On the other hand, organizations relying on project-based, open source solutions (for example, open source database drivers) are forced to use the software “as is” and are responsible for potential legal risks if the software is found to be in violation. The legal risks associated with open source database drivers are substantial and include the following items:

- Customer cedes licensing control to an amorphous group of developers** — Open source database drivers are often released under the terms and conditions of the GNU Lesser General Public License (Lesser GPL or LGPL). The goal of the LGPL is to provide protection for the code contributor or licensor, which means that the licensee (the organization using the software) assumes financial and legal exposure. If any code contribution made to the open source project has infringed on other code (copied from, reverse engineered from, and so on) that is protected by traditionally licensed software or a more restrictive open source license, the licensee can be held legally and financially liable. This means that the customer must trust that each code contribution made by an amorphous group of developers does not contain

plagiarized code. Conducting an audit is impossible given the lack of organizational structure and documented project history.

- **Open source projects' code due diligence is highly suspect** — Small, open source projects lack the infrastructure to properly monitor their code base. These organizations accept code contributions from developers who are unknown to the open source community. It is not unusual to view a forum exchange between an unknown developer who is contributing code and a developer within the open source project who has commit status. The exchange basically consists of the unknown developer offering the code contribution and the project member stating that the code will be integrated into the code base. There is no due diligence or proper vetting of the code relative to its origin. Was a portion of the code copied from a commercial project? Does the code infringe on a patent, legal contract, or licensing agreement? This process should be concerning to an organization that is considering using an open source database driver, because it greatly increases the chance that the product contains suspect code. DataDirect follows a rigorous methodology that ensures that the DataDirect code base is clean of code infringement and patent violations. All code that is checked into the source code repository is reviewed, and each employee is contractually obligated to abide by a series of rules and regulations relating to the origin of their work. In addition, DataDirect's product is legally protected by a substantial number of patents as well as pending patents.
- **Patent violations may result in legal action** — In addition to the exposure related to code plagiarism; it's possible that an open source project may have violated a patent. If an open source project is found to have violated a patent, the commercial interest that owns the patent may take legal action against any organization that uses the open source software. At the present time, there is considerable consternation within the Linux community that Microsoft may initiate litigation based on its library of patents. This is a stark example of the fact that violations may not be related to only plagiarized code, but can be related to deliberate or unknown piracy of thoughts and concepts protected by legal patents.
- **Intellectual property indemnification is not provided** — Open source database driver projects do not provide legal indemnification, forcing an organization that uses open source software to be completely responsible for all legal risks and litigation costs associated with a potential violation introduced by an unknown, open source developer. Traditionally, licensed software and commercial open source solutions provide various levels of indemnification support that cover legal damages and associated costs. In addition, the likelihood of a legal breach is mitigated by the fact that most commercial organizations have rigid code check-in and audit procedures that safeguard their code.
- **Organizations using the software are the legal target** — If it is found that an open source project infringes on the copyrights, patents, trademarks, or trade secrets of a commercial software vendor,

organizations that use the software are at high risk. It is likely that the commercial software vendor will pursue legal action against an organization using the software instead of litigating against an open source project that has no financial resources. The organization using the open source software may be liable for license fees and legal damages, and it's possible that their business operations will be disrupted because the rights to use the software can be immediately revoked.

The public stigma associated with legal proceedings can also be extremely damaging. Although this may sound farfetched, this is the exact scenario that is currently happening with SCO and Linux. In addition to SCO suing IBM, SCO sent letters to over 7,000 companies warning them about potential license violations. They subsequently pursued litigation against DaimlerChrysler and AutoZone and reported that it had signed at least one Fortune 500 company to a license agreement for SCO UNIX rights. According to the Yankee Group, 45% of organizations now consider indemnification an important issue (representing a 38% increase from the previous survey). This perception is driven in part by the threat of litigation. From a cost perspective, the American Intellectual Property Law Association (AIPLA) is on record as stating that the average cost for a business with \$25 million in assets to defend itself if named as a third party in a patent infringement suit is approximately \$3 million and can be as high as \$8 million for a firm located in a major metropolitan area.

- **Open source projects based on the LGPL do not include representation of quality** — Open source database drivers released under the LGPL license do not contain representations and warranties of quality. This introduces a major concern given the lack of due diligence applied to code check-ins and the lack of rigor involved in an open source project's testing methodology.
- **Legal interpretations vary by country** — Although software licenses are designed as international agreements, copyright laws are national laws that are regulated by territorial governments. This adds legal complexity for multi-national organizations because the risks and mitigating actions vary by territory.
- **Software license terms are subject to change** — Organizations that use open source software do so without exercising a legal contract. Although the user is bound by the license agreement, a contract is not in place that prohibits the open source project from modifying the terms of the license. This means that the licensor can change the license conditions, including making them non-free, withdrawing the software altogether, or changing to a more restrictive open source license agreement. This is true regardless of whether the license claims to irrevocable.

It is critical to note that the aforementioned legal risks exist for any organization that uses open source software, including open source

database drivers. These risks do not depend on whether an organization modifies open source code. If an organization simply uses open source software, an organization is immediately at risk.

Additional legal risks are associated with the act of modifying the source code; these risks include the following items:

- **Organizations must open source their modifications or contribute them back** — If an organization modifies open source code and if the code is shared outside the organization, the organization is bound by the LGPL, and many other open source licensing agreements, to release the new driver under the LGPL or contribute the code back to the open source community. If an organization releases their own driver, they are tasked with the ongoing effort of maintaining and enhancing the driver. If an organization contributes the source code back, they must rely on the open source community to include it in the next release of the open source database driver, which may not map to the organization's development schedule. Either way, the organization is responsible for additional work and must ensure that they follow the proper procedures to comply with the license terms.
- **Internal use can be compromised easily** — According to the LGPL license, internal use can be compromised by something as simple as an external contractor having access to the code. Although it may seem relatively simple, an organization must implement and enforce rules and regulations to ensure that the code stays internal. This effort places an additional burden on an organization, increasing the cost of using an open source database driver. If internal use is compromised, the organization can lose critical intellectual property of its own and be legally required to share it with the world at large.
- **Corporations restrict what can be contributed** — Typically, organizations have strict policies about the ownership of code developed by employees. Organizations often have formalized policies to protect their own intellectual property. If an organization is planning on actively participating in open source projects, they must account for this in their policies, which is an additional legal burden an organization must manage. In reality, many organizations do not manage this properly; their employees may contribute the code without the knowledge of the organization, which can lead to a number of legal problems.
- **Code contributions can "fork" the development effort** – Even if organizations implement policies and procedures to properly manage code contributions, they must depend on the open source project to implement their changes in a timely manner. First, nothing obligates an open source project to implement the changes, and if they choose to implement them, there is no timeframe to manage this work. This situation can lead to an organization managing

multiple versions of the database driver solution, which introduces additional development, testing, and deployment effort.

The net effect of managing these changes forces an organization to focus on an infrastructure component — a component that is clearly not their core competency. This effort reduces the amount of time dedicated to strategic projects that clearly drive the success of the business venture.

+++++

Independent Software Vendor Alert

In addition to the risks associated with most organizations, ISVs assume an additional level of risk because their software solution is a core component of their business offering. ISVs need to consider the following licensing ramifications:

- **ISVs that modify open source code and embed the driver assume legal risk** — If an ISV modifies open source, database driver source code, they are bound by the LGPL licensing agreement to release the new driver under the LGPL or contribute the code back to the open source community. If the ISV releases their own driver, they are tasked with the ongoing effort of maintaining and enhancing the driver. If an organization contributes the source code back, they must rely on the open source community to accept their changes and incorporate them in the next release cycle, which may not map to their product release cycle. Either way, the ISV is responsible for additional work and must ensure that they follow the proper procedures to comply with the license terms.
- **Derived works must be open sourced** — The LGPL states that works that have been derived from LGPL code must, in turn, be open sourced. If an organization modifies the source code, it may be considered a derivative work and subject to the LGPL. If an organization is careless about their implementation (for example, they link the code instead of implementing the driver as a separate and discrete entity), a legal argument can be made that the ISV's entire product is a derived work, subject to release under the LPGL. An organization must ensure that their developers act within the confines of the LGPL and that their legal and management staff is intimately familiar with open source licenses.
- **In an extreme case, distribution of an open source driver might be ruled to require the software company to open source their own application** — The LGPL purports to protect anyone distributing an open source driver with an application from this sort of legal action by stating that a non-open source program can be “linked” to an LGPL component without infecting the non-open source program with the LGPL requirements to open source. Because litigation involving the LGPL license has not resulted in an official court ruling, there is no assurance as to how a judge or jury would rule on the question relating to “linked” vs. “combined” software. If a judge or jury rules that software

has been “combined” with an LGPL component, the software is subject to terms of the open source license.

+++++

Organizations that rely on open source database drivers assume a tremendous amount of legal risk. Perhaps more importantly, implementing and enforcing the policies and procedures and the additional development burdens associated with an open source solution forces an organization to reduce their focus on their core competency and development efforts that provide their business with a competitive advantage. The following table outlines the clear advantages provided by DataDirect and the risks associated with an open source database driver relative to legal liability.

DataDirect	Open Source Database Projects	Customer Impact
DataDirect has the financial, legal, and development resources required to implement policies that ensure DataDirect's product is free from licensing issues.	Organization cedes control to an amorphous group of developers whose lack of financial, legal, and development resources greatly increases the possibility of licensing issues.	Customer assumes complete legal responsibility if they use an open source database driver. This risk is substantial because the open source project has no accountability and the customer lacks visibility into the development processes used by the project.
DataDirect provides legal indemnification and quality guarantees that provide customer protection and legal assurance.	Open source database driver providers do not provide legal indemnification or quality guarantees. Customers who use an open source database driver assume total responsibility for the “as is” product.	Legal Indemnification is a base requirement when dealing with software acquired from an external source. Without indemnification, customers assume tremendous legal risk for actions that are beyond their control.
DataDirect provides an aggressive product roadmap that ensures its customers receive driver enhancements in a timely fashion. DataDirect customers focus on their application needs instead of investing their time and resources in the database driver.	The nature of open source software forces a customer to modify the source code to compensate for missing or sub-optimal features. These changes are subject to the LGPL, which means that the organization must contribute the changes back to the open source project or open source their version of the driver.	Customers are forced to dedicate valuable development resources to the database driver instead of focusing on their application needs. If the code is contributed back to the open source project, the customer must deal with integration and version control issues. In addition, the customer must manage the risks associated with releasing their driver, or entire application solution, as an open source offering.

Unreliable Technical Support

Reliable technical support is a basic requirement when working with software from an external source. In today's hyper-competitive business environment, reliable, multi-channel (phone, email, and so on), 24x7x365 support is imperative for mission-critical applications. IT organizations are increasingly being held to high levels of service, and in many cases, their applications are

subject to availability, reliability, and performance thresholds established in service-level agreements. If reliable technical support is not available, the IT organization assumes a significant level of risk, because even simple technical support issues can turn into major project delays, lost developer time, application downtime, and so on.

Open source database driver solutions are not backed by organized technical support. At best, these projects provide online forums where a developer may submit questions, bug reports, and so on. The developer has no assurance that a response will be provided, and there is no guarantee or contract in place that regulates the response time of the resolution. Open source database driver project teams simply lack the resources to provide reliable technical support. The organization must depend on its own development resources to resolve issues if an organization runs into problems during installation or configuration of the driver, performance or scalability issues that occur when the driver is deployed in a production environment, problems that occur when upgrading the driver to a new release, and so on. Problems are especially difficult to troubleshoot when a developer did not write the code and may be completely unfamiliar with the mechanics of the driver.

This situation is in stark contrast with the technical support offered by DataDirect, which recently won its fourth consecutive Omega NorthFace Scoreboard Award for excellence in customer service. DataDirect Technologies' SupportLink Technical Service Program was recognized for its outstanding customer service record and strong commitment to customer satisfaction. DataDirect's SupportLink offering includes the following abilities:

- 24x7x365 phone support available toll free in most countries.
- SupportLink Online is your private website for getting the answers you need, when you need them. Online services include automated case reporting, downloadable product fixes and "dot" releases, the industry's largest data connectivity KnowledgeBase, and extensive product documentation.
- SupportLink OnSite consulting services can help you with short-term projects — from performance testing to debugging code to training.
- SupportLink ConnectLab allows you to test your applications and products against most platforms and databases.

DataDirect leads other software providers in terms of the level of quality, responsiveness, and customer satisfaction that it provides to its customers. This level of assurance eliminates technical difficulties relating to database connectivity, which is critical to application availability, and enterprise-class reliability, performance, and scalability. The following table summarizes the advantages of DataDirect's award-winning technical support when compared to open source software offerings.

DataDirect	Open Source Database Projects	Customer Impact
DataDirect provides award-winning technical support that includes phone, email, and Web-based support. In addition, DataDirect provides an extensive library of articles on database middleware topics.	Open source database driver providers do not provide any level of organized technical support. Customers who leverage these components must be prepared to provide their own technical support or risk debilitating application reliability and downtime problems.	With open source database providers, the customer must assume control for providing their own technical support, forcing the customer to make a significant investment in supporting a product for which they have no development expertise. Even simple problems can lead to costly downtime and user-satisfaction issues.

Summary

The perceived advantages associated with the open source movement (improved quality and lower cost) depend on having a large, vibrant community of developers and testers. Although these advantages (to a limited degree) have been realized in large, complex components, such as operating systems and application servers with large and active open source communities, the critical mass necessary to realize these benefits in an open source database driver simply does not exist. Open source projects typically are supported by a small group of part-time developers, offer limited functionality and database support, and lack the commercial backing to provide the legal assurances needed by enterprise-class IT organizations.

These limitations force customers that rely on an open source database driver to assume a significant level of risk — risk that spans the entire application including legal, product, and support elements. The financial costs associated with these risks can easily overcome the initial lower acquisition cost associated with a free license. In fact, when the aggregate costs involved in the project are considered (technical support, development and quality assurance cost, increased end-user satisfaction based on performance, driver reliability, business risk from the legal issues involving the open source license, and so on), the DataDirect driver has proven time and again to be the most cost-effective data connectivity alternative available today.

We welcome your feedback! Please send any comments concerning documentation, including suggestions for other topics that you would like to see, to:

docgroup@datadirect.com

FOR MORE INFORMATION

800-876-3101**Worldwide Sales**

Belgium (French).....0800 12 045
Belgium (Dutch).....0800 12 046
France.....0800 911 454
Germany0800 181 78 76
Japan0120.20.9613
Netherlands0800 022 0524
United Kingdom0800 169 19 07
United States.....800 876 3101

Copyright © 2005 DataDirect Technologies Corp. All rights reserved. DataDirect Connect is a registered trademark of DataDirect Technologies Corp. in the United States and other countries. Java and all Java based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Other company or product names mentioned herein may be trademarks or registered trademarks of their respective companies.



DataDirect Technologies is focused on data access, enabling software developers at both packaged software vendors and in corporate IT departments to create better applications faster. DataDirect Technologies offers the most comprehensive, proven line of data connectivity components available anywhere. Developers worldwide depend on DataDirect Technologies to connect their applications to an unparalleled range of data sources using standards-based interfaces such as ODBC, JDBC and ADO.NET, as well as cutting-edge XML query technologies. More than 250 leading independent software vendors and thousands of enterprises rely on DataDirect Technologies to simplify and streamline data connectivity. DataDirect Technologies is an operating company of Progress Software Corporation (Nasdaq: PRGS).

www.datadirect.com