

# db4objects – Innovating Object Databases with Open Source

by Paul F. Johnson <editor@accu.org>  
and Roberto Zicari <roberto@zicari.de>

Many open source products and projects can easily be mapped to closed source products and often match their scope, concept and capabilities. Linux as an operating system, MySQL as a relational database, OpenOffice as an office suite immediately spring to mind as examples of this. Their main competitive advantage often is their license price being zero. As a consequence, it is often claimed that open source is commoditising software, driving down prices and giving buyers a direct alternative to often monopolistic, high priced, closed-source products.

However, away from the limelight of the open source main street, there's also a series of open source products that are not followers but leaders to the way we use and build software (such as Apache, Hibernate, and db4o.)

## db4objects

### db4objects?

db4objects is the creator of db4o, the leading open source object database for Java and .NET. Open source since November 2004, it has seen an impressive growth and is poised to change (or add to) the way we think of persistence (a fancy way to say: database) in object-oriented programming environments.

Recently, I had the chance to interview Christof Wittig, CEO of db4objects ([www.db4o.com](http://www.db4o.com)).

Object databases have failed in the 1990s. Why would db4o's open source object database be more successful?

*db4o is far more successful than any vendor has ever been to introduce a persistence solution that simply fits object-oriented programming (OOP) languages such as Java and .NET perfectly.*

*In just 8 months we have had over 250,000 downloads, built a user community of more than 6,000 professional Java and .NET developers and have signed commercial contracts with the likes of BMW, Hertz and Bosch. The main reason for db4o's successful launch is the fact that we're open source which gives us a powerful tool to enter the market at low cost, build a large user base very fast, and outpace any closed source vendor in setting de-facto standards.*

So you just go open source and that makes the difference?

*No. There are also differences in the market segment we target and the advent of OOPs in the main stream.*

*When object-oriented databases came with great fanfare to market in the early 90s, the protagonists saw them as a displacement for relational databases in the data centre. We believe they targeted the wrong segment with the right product.*

*In contrast to 1st generation object-oriented databases, db4o is the embeddable persistence solution for all client-side Java and .NET applications. As an example, Eastern Data in Hatfield Perrevel, Northeast of London, builds its future line of mobile PDA applications for field force automation (FFA), e.g. for van deliveries of milk, with db4o. They enjoy db4o's zero-administration capabilities, reliability and fast performance. Not only is it possible to bring products to market faster, but also to build more*

*object-oriented, better software which is easier to re-factor and reuse in the long run than when using relational database technology.*

*We also benefit from the increased adoption of Java and .NET. Today it is clear that OOP are the future. Hence, the so-called object-relational mismatch, the inherent incompatibility of OOP with relational databases has become a real business problem, not an academic challenge as it was in the 90s.*

*In addition, we were able to write our product in native Java and .NET which makes the database even less intrusive than old-style databases written in C, more easy to work with and to deploy in large numbers. After adding a single JAR/DLL file to your project, it takes just one line of code to store any object - no matter how complex your object model is!*



Christof Wittig, CEO of db4objects

Relational databases such as those of Oracle, IBM, or MySQL are the market leaders. How do you want to compete with them?

*They all also target the server side data storage and do a great job there. But they fall short on clients, in zero-administration environments. The reason for this is the inherent incompatibility of relational data models and the object schema used by Java and .NET developers.*

*As a result, the big three, Oracle, IBM and Microsoft combined control about 85% of the overall relational DBMS market, while they command only 25.1% of this embedded DBMS market. We believe that these facts prove that customers seek specialized capabilities in various segments of the embedded market, that go beyond what RDBMS can offer.*

Let me play devils advocate: There's a lot of data legacy. How do you want to mitigate the immense efforts it takes to switch a database.

*That's a great question and surely another reason, why old-style, server side OO databases failed: They weren't able to provide a painless migration path. This is totally different for db4o, the embeddable object database: There's no legacy in devices, on the client side. Every instance of a BMW car, for example, that is shipped with db4o, has a fresh database instance running. No data migration here. Also there's no "legacy in mind" because we have no DBA and his set of tools such as report writers. Usually you don't write ad-hoc reports against the database running in your game box, do you?*

But what about object-relational mappers like open source Hibernate from JBoss or closed-source Toplink? Aren't they a solution for the object-relational mismatch problem?

*Object-relational mappers are a good solution on the server side where resources are abundant and/or performance not critical. On the client side, i.e. in packaged software, in mobile or gaming devices, and in real-time control systems they are prohibitive. The reason is the lack of zero-administration capabilities through the added complexity and the huge drain on performance.*

*Open source benchmarks ([www.polepos.org](http://www.polepos.org)) have shown that db4o is up to 44x faster than a MySQL + Hibernate stack, for instance. However, we're very happy about the immense growth of these platforms, as they validate the extent of the object-relational problem in our industry.*

Back to open source. We understand that you get a lot of hype. But how do you make money?

*We adopted the dual license model as pioneered by Berkeley DB and MySQL. We provide our software in its entirety under two licenses, the GPL and a*

[continued from previous page]

### Summary

Java 5 introduced new language features that allow developers to write code that is more robust whilst at the same time reducing the amount of boiler-plate code that needs to be written. If you're new to Java or are still using JDK 1.4, then it's worth while investigating all the new features provided in Java 5.

Dave Salter

### Resources

Apple Java Website - <http://developer.apple.com/java>  
EJB 3 - <http://java.sun.com/products/ejb/docs.html>  
JAX-WS 2.0 Web Services - <https://jax-rpc.dev.java.net/>  
Sun Java Website - <http://java.sun.com>  
Xdoclet - <http://xdoclet.sourceforge.net/xdoclet/index.html>

commercial license. The GPL licensed version is available for free download from our website [www.db4o.com](http://www.db4o.com). People can use it, evaluate it, getting educated about db4o's immense benefits.

However, the GPL has the obligation to open source your derivative work under the GPL, too, if you start to redistribute it. Therefore the GPL is a no-go as an input component to most commercial, product developing companies. For these customers we provide an additional, affordable commercial runtime licenses which frees them from this obligation and enables them to ship closed source products with embedded db4o. To commercial customers, we also provide direct support and vendor relationship, e.g. to discuss product roadmaps, which are important criteria when evaluating a platform as central as a database, in form of a db4o Developer Network (dDN) membership subscription.

However, we do not provide engineering and anything other than product-related support services. We focus on making the product easy to understand rather than artificially building a complicated product that needs expensive training and consulting to be deployed properly.

With db4o you are up and running in 5 minutes. A great interactive tutorial guides you through the few basic APIs that help you to get most out of the product. The number of users and customers asking us for training is basically zero.

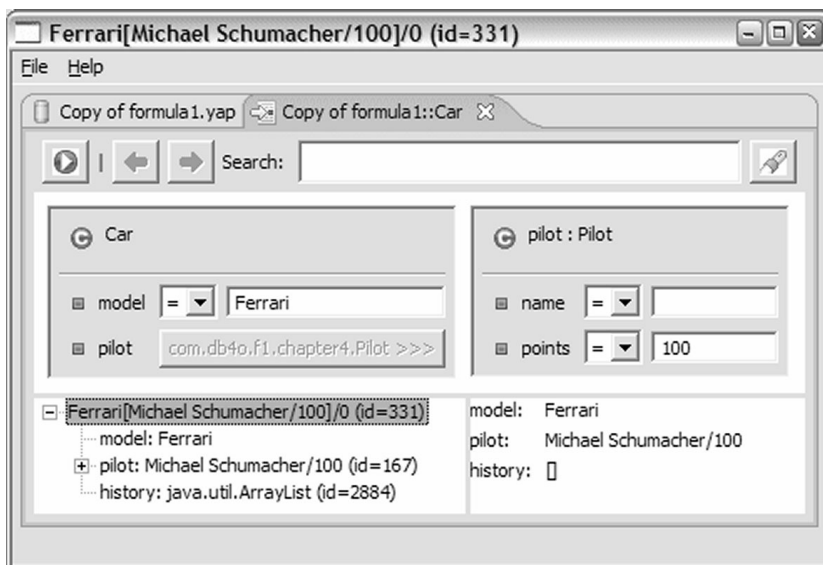
How affordable is your affordable pricing and how can it be sustained?

Affordable means up to 10x less expensive runtime prices than closed-source vendors, such as IBM, Oracle, Sybase, and a long tail of smaller vendors.

We can sustain to charge much less because the open source model basically saves us a lot of spending for product, sales, and marketing.

It is a triple win situation: Commercial customers get better software at lower prices, the community gets a great product for free, and we are able to build a sustainable business. This is only bad news for conventional companies with over bloated sales and marketing departments that will suffer.

You mentioned product development. How does your development model look like?



Product development is firmly embedded in the community of many 1,000s of developers that use db4o and a few that are actively writing the core, which are generally paid for by us.

These employees are recruited from within the community. They can be as dislocated as Brazil, Siberia or Germany, but still work very efficiently together.

We use extreme programming blended with open source collaboration techniques and tools. Central communication is through our newsgroups, but we also set up Skype sessions for voice interaction and a Skype/TightVNC combination to run virtual pair programming sessions. Bi-annually we meet face-to-face in destinations as exciting as San Francisco, the Bavarian Alps, or Salvador de Bahia in Brazil to discuss the product roadmap, design proposals, and build team spirit.

All this works with great success for us. We were not only able to commit the smartest guys to our vision, such as Klaus Wuestefeld, author of Prevaler, Rodrigo de Oliveira, author of Boo, and Dave Orme, Eclipse Visual Editor lead, but we also managed to make them work with an efficiency that compares

with a 10x over what I know from old-style, collocated software fabs. What do you look for in a new employee?

I am glad that you mention this. It is very important that each individual software engineer starts to envision how he would fare in this new, dislocated development model. Having experienced its power, I can hardly imagine how siloed software fabs will be able to compete in the long run against it. You can see IBM and others starting to embrace this model already.

We look for individuals with

- outstanding, relevant accomplishments,
- team orientation and good communication skills, and
- the ability to self-manage.

Relevant accomplishments obviously will differ for each area of interest. Someone who has build an open source object-relational mapper "for fun" is certainly of great interest to us. We will look at the quality of the source code and take into account the person has displayed a lot of passion for the subject of our work.

Team orientation is essential. Lonely stars will be rejected by the team and the community. There are great developers out there, but nothing matches the power of a hot core team effectively interacting with a large, diverse community. As a prerequisite, good communication skills, especially via newsgroups and e-mail are required.

Being dislocated and working from your basement also requires a certain discipline as to keep track of priorities and manage and balance life. It is a great benefit for our employees that they can, beside work, take care of kids, travel, live in remote areas, etc. Each of our developers has his reason why he likes the dislocated model - be it the ability to move away from crime-ridden Sao Paolo or to ski in the Alps whenever the weather permits. However, all this requires self-management to bring life in pace with work. We have no line managers but only very broad directives from senior management. Evaluation happens by results not attendance and other behavioural observations. So be sure to find your way to deliver in time.

And here is my advice to people seeking a job: When we hire, we want to see proof of all of the above. We're not training new employees in these skills, we expect them to come with these skill. So think about start building a track record of accomplishment in the open source community that clearly shows proof of your ability to work in the environment I described.

We recently rejected a very qualified candidate with an impressive vita. His job description included extensive posting in newsgroups. However, he didn't have a newsgroup track record at all and the few postings we saw were poor and not very supportive in its nature. Why should we expect he would suddenly change when hired by us? Remember that any code you contribute or any posting you make in a newsgroup is stored by Sourceforge and Google Groups forever. And recruiters will start to look at them (or their absence)!

db4o has recently announced a new way of writing a database query, called Native Queries, using entirely programming languages such as Java and C#. What is the difference with respect to the traditional SQL-way of query a database offered by relational database systems?

Native Queries (NQ) are a new, additional API to db4o which uses the programming language itself - Java or .NET - to query the db4o database. Native Queries are based on Safe Queries as proposed by William Cook, Prof. at University of Texas, at the 27th International Conference on Software Engineering (ICSE).

Over the last 15 years, there was a lot of thoughts and proposals around building a new OO query standard which would be the equivalent of SQL for ODBMS and ORMs -- OQL and JDO are examples for this. None of them have become mainstream and hence fail to be a standard.

We think that an embedded ODBMS doesn't need an additional query language such as SQL. SQL was mainly designed for DBAs that want to query the database directly, e.g. for ad-hoc reporting. Embeddable databases are in zero-admin environments. The only user of the API is the developer who already knows and uses one language: The programming language. So we decided to standardize on the standard he or she already uses: Java or .NET.

As a result, using native queries, you can use a lot of the productivity enhancements provided by your IDE. You get a 100% typesafe code (no

[concluded at foot of next page]

# Setting up a Subversion Server for Remote Use

By Craig Ringer <ringer@scribus.info>

Revision control is a critical part of any significant development project. Having secure full time access to your repository from wherever you are can be important. In some environments, such as open source projects, it's absolutely crucial to the functioning of the development team.

My first exposure to the Subversion[1] revision control system was when I was searching for something to replace CVS for the Scribus project, whose CVS server I administer. There were a number of problems with our use of CVS over ssh, namely server security concerns, cross platform issues, and configuration complexity.

After some research, I decided that Subversion was a good candidate to replace CVS. Subversion is a fairly new revision control system with the stated goal of being a "better CVS than CVS." One particularly attractive feature is the use of HTTPs for a secure, fast, encrypted transport that eliminates the need for an SSH tunnel. Additionally, Subversion is becoming increasingly popular in the open source development community, so more and more useful tools and graphical clients are becoming available.

I'll be covering how to set up a Subversion server for your team, so they can work on your code wherever they are without introducing major security risks. Specifically, I'll be explaining how to set up a Subversion server configured for use with WebDAV over HTTPs, using SSL client certificates for an additional layer of authentication. The goal is revision control that's fast, secure, and easy to use from anywhere. I'll be assuming that you're using a UNIX variant, but it should be quite possible to set all this up on Mac OS X or even Windows NT/2k/2k3 as well.

This article won't try to explain how to use Subversion, why you might want revision control, basic UNIX command line use, or any related topics. We're going to focus specifically on setting up reasonably secure remote access to a subversion repository. As this article also avoids going into extreme depth on the use of subversion and other finer points of its configuration, it is strongly recommended that you examine the excellent Subversion book[2] (freely available on-line) for more detail.

It is important to note that your author is not a security expert. I am an experienced system administrator who has operated Internet-accessible

systems hosting public services for some time, but beyond that has no specific security qualifications. This article does not provide some magic recipe for a secure server configuration – but it should help you get started along the way.

## Why use Client Certificates?

The use of client certificates provides an extra layer of authentication. A user can't even attempt to authenticate against your svn server until they've provided a client certificate that you can verify is signed by your CA and has not been revoked. For a bit of extra security, you can also store client certificates separately from the client host – on a USB key or potentially even a smart card – and remove them when not in use.

Client certificates are also useful for controlling access to more than your source code repository. They can be used to help secure SSL/TLS-based mail services such as encrypted IMAP, POP3, and SMTP. They are also useful if you wish to offer controlled HTTPs-based remote access to your organization's intranet web server for roaming users. You can also use the same client certificate infrastructure to permit users to encrypt and/or sign email within the organization using S/MIME (though recipients who have not imported your CA certificate won't be able to verify signed mail).

## What You'll Need

First, you'll need a server with the Apache 2 web server, the `mod_dav_svn` Apache 2 server module, and the subversion tools installed. On Debian GNU/Linux 3.1, just `apt-get install apache2 libapache2-svn subversion`. For other platforms, if you're unsure how to go about installing Apache 2 or `mod_dav_svn`, the Subversion web site[1] has plenty of information.

OpenSSL[3] will be required to create certificates, so unless you have an existing CA and x.509 PKI scheme you'll need to install that.

If you want to use your repository from anywhere, you'll need an Internet-accessible IP, or a port forwarded through your firewall. You can use a VPN if you prefer to further limit the accessibility of your server.

## Certificate Creation and Management

The creation and management of SSL certificates can be a complicated business. Once it's familiar you will find that it's not generally an issue, but the initial process can appear somewhat daunting.

If your organization has existing X.509 based PKI infrastructure, you may

[continued from previous page]

strings!), 100% refactorable code, and 100% object-oriented code, which is easily optimizable.

We believe that this powerful, open concept will find wide industry adoption and become the standard way to query databases in an OO way. db4objects is the first industry player to adopt the standard and puts the power of the open source model behind it. A preview version (V5 milestone 2) of the new API is available for free download from our website. Also I invite to read the free whitepaper by William Cook and Carl Rosenberger, available on [www.db4o.com/about/productinformation/whitepapers/#nq](http://www.db4o.com/about/productinformation/whitepapers/#nq) which elaborates more on the design concept and goals and discussed advantages and disadvantages.

db4o was originally started in Germany. How did you get to base your business in Silicon Valley? What role does Europe and the UK play?

The product started in Germany, when Carl Rosenberger realized his dream on 1/1/2000, to free OO developers from the OR mismatch.

The corporation started last year in Silicon Valley, the central trading place for ideas and technologies. With private investors such as Mark Leslie, founding CEO of Veritas, and Diane Greene, founding CEO of VMware we had found the right people to put their names and resources behind the idea and launch db4objects, Inc. While we don't produce in Silicon Valley, we see it as the place to do global marketing, sales and finance.

Europe is very strong for us, given our origins and the lead Europe has in mobile applications, for instance. The UK are constantly ranking among the top 5 countries for db4o, together with the US, China, Japan and Germany. On September 29, we will host our first local event in London's Imperial College, where Glasgow University's Professor Jim Paterson will introduce db4o (more information on [www.db4o.com/about/productinformation/events/fall105rsj](http://www.db4o.com/about/productinformation/events/fall105rsj)).

Tell us about how you see the DBMS market evolve? What role does the embedded DBMS play and how does this affect your business plans?

I leave this answer to the leading analysts.

According to IDC's estimates, the embedded DBMS market grew 15% to \$1.86 billion in 2004, and is expected to blossom to \$3.18 billion in 2009: "Object-oriented DBMSs could well enjoy a second growth period as embedded DBMSs due to the efficient and flexible data management they offer object-oriented applications, and open source DBMSs are also attractive as embedded DBMSs because of the technological control they offer ISVs as well as flexibility in licensing," says Carl Olofson, research director for information and data management software at IDC. "db4objects is in the interesting position of offering the benefits of object-oriented DBMS technology and open source licensing, making its value proposition appealing on two fronts."

Chris Lanfear, director at Venture Development Corporation (VDC), says: "Especially on the client side, such as in stand-alone devices and other zero-administration environments, engineers look for innovative persistence solutions that meet their immediate specifications and help them outrun the competition. As a result, more than 50% of embedded and device software developers still build their own database tools today. With the advent of standardized object-oriented platforms, such as embedded Java and the .NET CompactFramework, we expect object databases to become a universal solution for OO persistence - with db4o's open source offering leading the charge."

I have nothing to add to that!

Christof, thank you for your time in giving this interview.

Paul Johnson  
Roberto Zicari

More information on db4objects can be found at:

<http://www.db4o.com>