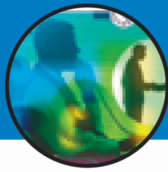
SYS-CON

Practices:
The Importance of Open Source Governance in Mitigating Risk

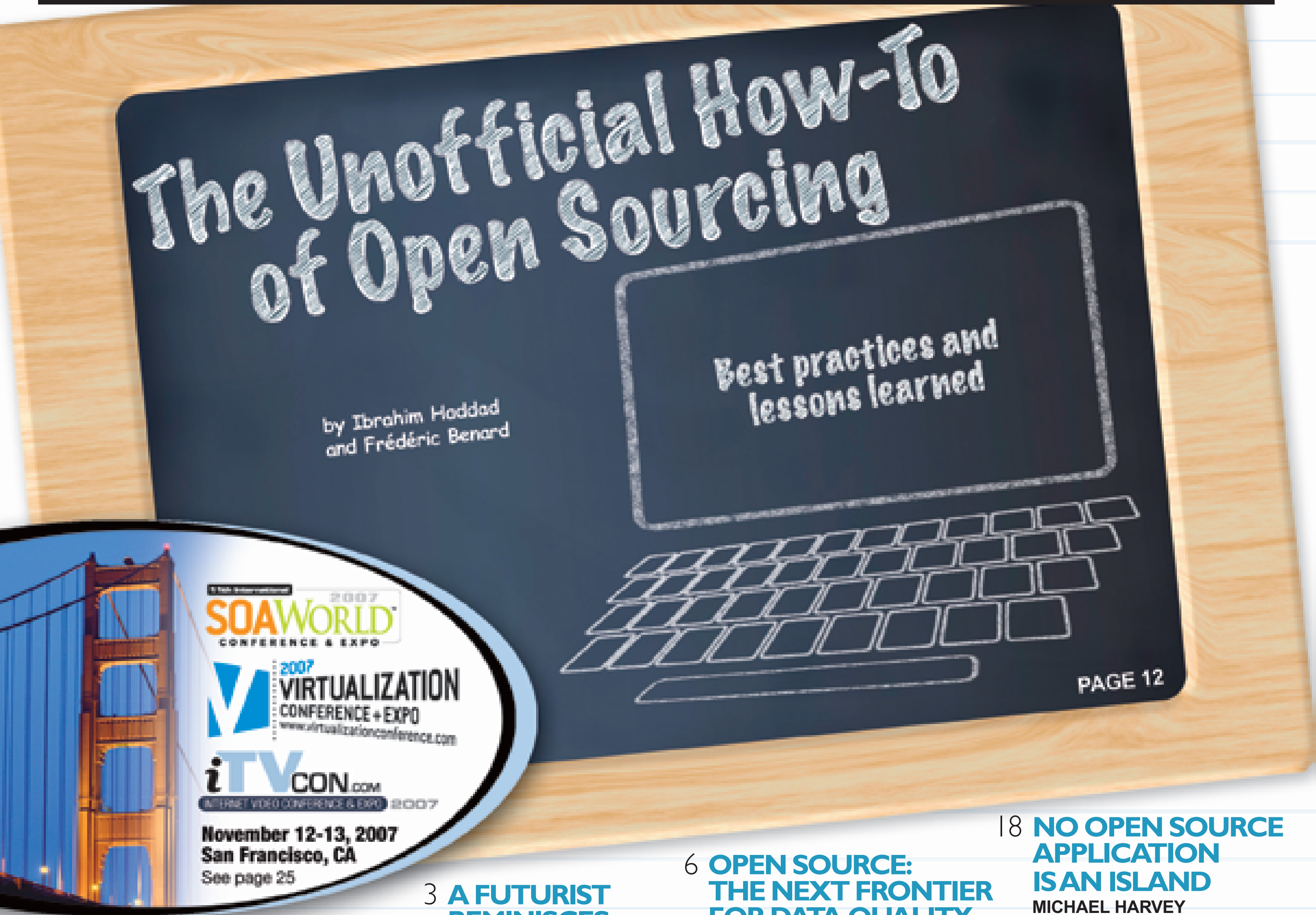


Eclipse:
Using the Eclipse Data Tools Platform with SQL Anywhere

ENTERPRISE OpenSource MAGAZINE

Visit us at **opensource.sys-con.com**

THE LEADING MAGAZINE FOR ENTERPRISE AND IT MANAGEMENT
OCTOBER/NOVEMBER 2007 VOLUME 5 ISSUE 6



PRESORTED
STANDARD
US POSTAGE
PAID
ST. CROIX PRESS

ENTERPRISE> OPENSOURCE MAGAZINE
www.OpenSource.SYS-CON.com

OFFICES

SYS-CON MEDIA
577 Chestnut Ridge Rd. • Woodcliff Lake, NJ 07677
Telephone: 201 802-3000 • Fax: 201 782-9600
Enterprise Open Source Magazine (ISSN #PENDING)
is published monthly (12 times a year)
by SYS-CON Publications, Inc.

Postmaster send address changes to:
ENTERPRISE OPEN SOURCE MAGAZINE
SYS-CON MEDIA • 577 Chestnut Ridge Rd. • Woodcliff Lake, NJ 07677

WorldWide Newsstand Distribution
Curtis Circulation Company, New Milford, NJ

FOR LIST RENTAL INFORMATION:
Kevin Collopy: 845 731-2684, kevin.collopy@edithroman.com
Frank Cipolla: 845 731-3832, frank.cipolla@epostdirect.com

SYS-CON MEDIA www.SYS-CON.com

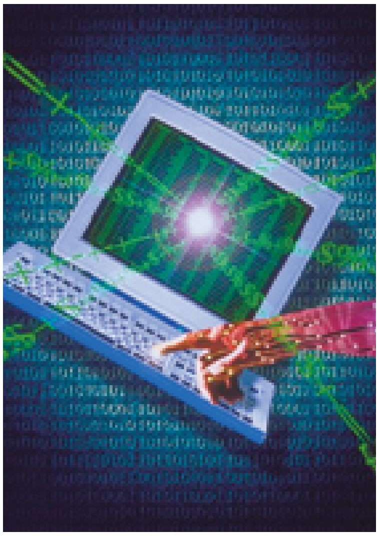
A Futurist Reminisces

By Mark Hinkle

It is with some sadness that I am writing my last editorial for Enterprise Open Source Magazine. As the founding editor-in-chief of this magazine and a past contributor to its predecessor, I am going to miss it. However, all things must end and this chapter of my writing career, I am happy to say, ends on a high note.

I have had the privilege of publishing hundreds of articles over the past years on Linux and other open source technologies read by Linux and open source users worldwide. I also got to meet many folks who have shaped the success of the open source industry. As someone who has spent his career looking forward, I thought I would take this opportunity to take a rare look backward.

I feel that I have the unique ability and good fortune to pick out and become involved in the next big thing. I have always landed smack dab in the middle of the technology trends that have and are reshaping our lives. Here's a short walk down memory lane.



The Internet

I remember when the evening news started to proclaim the birth of the "Information Superhighway" (I am so glad that moniker has fallen out of vogue). It's evident that the availability of instant information is valuable but the ability for global collaboration is what has really fueled the growth of the Internet.

I had the good fortune too during the mid 1990s to work for one of the first Internet service providers, PSINet. During that time I remember the first days of Yahoo!, was a Netscape beta tester, and helped support the earliest Websites of the Fortune 1000. I remember using a Mac everyday (running MacOS 7.5.3) and was the owner of one of the first Windows laptops (it was a Toshiba Satellite well before Dell was the established market leader). I helped launch cable broadband and DSL offerings when the norm was a 28.8 kbps modem. Later my team was one of the first commercial users of Linux desktops supporting one of the world's largest Web hosting businesses.

Finally, I watched the commoditization of Internet access. I saw the writing on the wall that Internet usage would be become as pervasive as cable television and I knew it was time to move on. I also observed the role open source played in this evolution – powering e-mail, Web servers (Apache), and BIND for DNS. Luckily I had the good fortune to dive into Linux, open source, and virtualization in my next endeavor.

—continued on page 5

About the Author
Mark R. Hinkle, editor-in-chief of **Enterprise Open Source Magazine***, is the vice president, Community and Business Development at Zenoss Inc. He serves as a founder of the Open Management consortium and is the author "Windows to Linux Business Desktop Migration" (Charles River Media).*
mrhinkle@encoreopus.com

www.Raritan.com/KVM

Raritan IT infrastructure management integrates KVM, serial and power control into a single, easy to use interface. You'll get a high-performance user experience that's indistinguishable from an at-the-rack connection.

If complexity, time and distance are taking too much of your time, visit our website. Raritan's products will simplify and accelerate your IT operations by letting you access and control more...with less.

Raritan.
When you're ready to take control®

—continued from page 3

I feel that I have the unique ability and good fortune to pick out and become involved in the next big thing. I have always landed smack dab **in the middle of the technology trends that have and are reshaping our lives.**

Open Source

After the Internet, probably the biggest trend shaping technology is open source. I am fascinated by how far things have come in a relatively short time. I remember using Linux on a pre-Pentium Compaq PC well before it had gained its current enterprise server status. The host name of the machine was Sharky and our idiosyncratic Webmaster Phil made me take a bash shell test before I was able to have my own account on the machine.

Today, open source is everywhere, running Websites like eBay and Google. Firefox (Netscape's open source offspring) runs on my desktop and OpenOffice.org is my office suite of choice. Open Source software has become pervasive in our networks and is slowly creeping onto our desktops.

Social Networking and Participatory Media

While I will never lose my interest in the Internet or open source, I must admit my current fascination is with the rise in social networking, online communities, and participatory media. I think that Enterprise Open Source was a good baptism for this as it was filled with articles written by contributors throughout the industry, not by typical journalists, and was a real participatory publication.

I am interested in how online communities grow, and the power they yield to help develop technologies and evangelize new trends. The networks that power them fascinate me. Once again I feel like a student studying the ways peer production can be used to accomplish monumental tasks, such as the development of the Linux operating system.

Thank You

I would like to thank everyone who has read and contributed to *Enterprise Open Source Magazine* and to everyone who has shared their thoughts with me both good and bad.

I am not writing off magazine publishing but I have become more enamored with the frequency of publishing a blog, something about the informality and immediacy is very appealing. I also find it difficult to meet the demands of a rigid publishing schedule with my other commitments. Recently my time has been consumed at Zenoss (www.zenoss.com) where, as the vice president of community, I am fostering an online community that supports the open source Zenoss Core project, perhaps a job that is too good to be true.

Despite my lack of a formal writing schedule, I will be spending what little time I do have authoring my blog at www.encoreopus.com (it's an anagram for open source in case you were wondering). Please feel free to tune in there or drop me a line at mrhinkle@encoreopus.com.

Open Source: The Next Frontier for Data Quality Management

Data quality, a pervasive & critical business issue

by Subbu Manchiraju

Data is the fundamental building block of every business, data in the form of client information, sales information, employee information, and financial information fuels the operation of every business. In today's business environment, which enables data entry from multiple points and through myriad processes, data quality has become an increasing concern for businesses trying to succeed in an ever more competitive atmosphere.

Data quality or data integrity as defined as incomplete, erroneous, or incompatible data is part of every business's day-to-day operation. Furthermore, as new flexible data entry options become available, the opportunity for data quality issues to be introduced into enterprise data increases. Overall business strategy is also increasing the prevalence of data quality issues as mergers, acquisitions, and department consolidations becoming part of almost every business's growth initiatives.

Data quality issues are often latent in an enterprise until a critical business initiative becomes road blocked because the enterprise data can't comply with the needs of the business. Companies of every size in every industry are increasingly reporting issues with data quality. The Data Warehousing Institute reported that 50% of its respondents felt that company data quality is worse than the organization thinks. Furthermore, more than half of respondents indicated their organizations had suffered losses due to poor data quality.

Data encompasses all the critical decision-making variables in an organization, including financial data, employee data, client data, prospect data, and inventory data. Viewing data that is erroneous or incomplete can seriously impact the decisions an organization makes and the strategies it employs. Recent research from Aberdeen indicates that the state of a company's data quality directly impacts its growth, profitability, and ability to compete. Poor data quality obscures an organization's view causing it to miss additional revenue opportunities, risk regulatory issues, and forfeit the intelligence gained from a clear view of business data.

As the prevalence and impact of data quality issues become more apparent, concern over these issues is reaching beyond the IT community to the C-suite. A recent study by the Financial Executives Research Foundation indicates that data quality across the enterprise was its number one concern, surpassing information security and Sarbanes-Oxley. Finance professionals cited information integrity as the key issue impacting overall corporate operations and performance.

Data quality is every organization's sleeping monster. It quietly erodes profitability, impedes growth, and hinders the implementation of mission-critical business initiatives.

The Limitations of Commercial Data Quality Solutions

Once an organization recognizes its data quality issues and their operational impact, it typically evaluates commercially available solutions to address the problem since most companies lack the IT infrastructure and knowledge to address enterprise data issues. However, for most companies seeking a data quality solution, the evaluation process is a sobering one because most commercially available solutions are costly, complex, and require software licenses and term contracts, while only addressing a portion of the overall issue.

Commercially available data solutions are fundamentally flawed in their implementation model. To be most effective data quality processes should be deployed at multiple touch points throughout an organization. Full implementations are almost impossible because they become cost-prohibitive when licenses are expanded to encompass more users and multiple systems.

About the Author

Subbu Manchiraju is a vice-president at Infosolve Technologies (www.infosolvetech.com), which provides business clients with comprehensive data solutions that leverage the power of their enterprise data to achieve business objectives and create strategic opportunities without the burdens of cumbersome licensing agreements, complex term contracts, or expensive hardware requirements. Infosolve embodies its Zero Based Solutions in every project with the Zero Defect Data Guarantee assuring that all results are free from new data errors and inconsistencies. Its Zero Based data solutions suite features signature offerings that are open source including, OpenDQ for comprehensive data quality solutions and OpenCDI framework for customer data integration.

> The Data Warehousing Institute reported that 50% of its respondents felt that company data quality is worse than the organization thinks. **Furthermore, more than half of respondents indicated their organizations had suffered losses due to poor data quality.**

Commercial solutions are also prohibitive to many organizations due to their term contract commitments, software licenses, and implementation requirements. Price tags for traditional solutions can often total in the hundreds of thousands of dollars if not over a million dollars, not including the human capital within the organization needed to manage the solution in concert with the provider. Such price tags make commercially available data solutions inaccessible to many small and mid-size enterprises that need data quality solutions.

Another drawback of traditional solutions is that they offer only cookie-cutter product approaches to data quality. Since most companies have data issues that are unique due to their specific organizational history and infrastructure, traditional cookie-cutter solutions often require significant programming and custom code development — all requiring additional testing, resources, and money, adding significantly to the complexity of the solution for implementation and service management.

Moreover, support for traditional solutions is typically limited to the providing vendor due to the proprietary software and licenses involved in the implementation of the solution. This restriction further increases the price tag of the conventional solution since support, service, and implementation can total as much as 70% of the purchase price of the solution.

Open Source: The Next Frontier for Data Quality Management

While open source has been gaining traction and attention in many business solutions, data quality solutions have remained an area where open source is not widely utilized. Open source, however, is well equipped to address the limitations of traditional software-based solutions or SAAS solutions and create industry-leading data solutions. Open source solutions are inherently better suited to address the needs of comprehensive data quality management with their flexibility, cost efficiency, customization, rapid integration, and turnkey scalability options.

A key benefit of open source data quality solutions is that they can be implemented at multiple data entry points throughout an organization because they require no license purchases. This flexibility creates a more comprehensive and longer-term solution than single-point commercial solutions.

Open Source data quality solutions also provide a significant cost advantage over conventional quality solutions because they require no software license purchases or management. Software licenses can account for up to 20% of the cost of a traditional implementation. This represents a significant cost savings to organizations. Furthermore, software licenses typically come with lengthy contract commitments attached, impacting the cost structure for an organization for a significant if not perpetual period of time.

Moreover, open source data quality software can be easily customized to address the unique data fingerprint of every organization eliminating the need to retrofit cookie-cutter traditional solutions with code modifications and custom programming. This customization ability reduces the complexity of the solutions and offers faster implementations, simpler integrations, less testing, and more rapid results than commercial solutions.

Another benefit of open source solutions is that servicing is more flexible and cost-efficient because it isn't tied to proprietary licensing. Service can then be provided by the technology vendor, secondary vendor, or internal resources. Furthermore, the open source community can also provide support and innovation for solutions as they evolve within an enterprise.

Lastly, open source data quality solutions have the added value of using the new technology processing systems dedicated to providing "pay as you go" (utility computing) processing options for turnkey scalability. This offers a further significant cost advantage over commercial solutions that require licenses tied to hardware. Data solutions are especially prone to scalability issues due to the volume of data undergoing processing, many traditional solutions become easily stressed due to these needs, increasing the costs, delaying results, and reducing the return on investment for traditional solutions.

It's clear that an open source solution for data quality offers many benefits to clients over conventional solutions. Open source provides all businesses access to critical data quality solutions that can positively impact their overall profitability, growth, and competitive position. Furthermore, the existence of the open source community enables a solution users' immediate access to shared knowledge and implementation enhancements, rather than waiting months or years for another software release. Open source can offer organizations the most customer-centric data quality solution available in the marketplace today with flexibility, customization, and significant cost advantages.

Research Sources:

- TDWI. "Taking Data Quality to the Enterprise through Data Governance 2005."
- Aberdeen Report. "{Customer Data Quality, The Roadmap to Growth and Profitability 2007."
- Technology Issues for Financial Executives 2007 Annual Report.

Make Markets Not War

A simple marketing model for open source – Part Two

by Dr. Ian Howells

This is the second part of my two-part series on open source market strategies and implementations. I previously outlined the 10 strategy rules for open source marketing and emphasized building new markets, differentiating, contributing, pricing and innovating, and the customer relationship.

As I mentioned in part one, a year ago I wrote "Howells' 10 Rules for Open Source Marketing." Here we're looking at where Alfresco is a year later in its marketing approach. Many of our open source peers are adopting the same principles, and I believe this evolution of open source and the realizations we've experienced can be used by emerging open source entrepreneurs and commercial open source companies still looking for the best model for their particular business.

Implementing a Marketing Strategy for the Open Source Consumer

Implementation Rule 1 - Get consumerized. You can't beat a gorilla by being a dinosaur.

We used to talk about "Gorillas" like Siebel and Peoplesoft but now even being a gorilla isn't enough. Today there are mega-gorillas like Oracle, Microsoft, IBM, and SAP. All others are caught in a no-mans land and vulnerable to open source. Mega-gorillas will be generic and sell everything.

IT procurement has undergone a consumerization process where users go through:

- Discovery
- Research
- Try/Download
- Join a Community
- Buy – Support, Training, Consulting
- Process – The software is discovered through the Web, Sourceforge, blogs, keyword search, forums, RSS, podcasts, webinars, trials, downloads, traditional media, and word-of-mouth. Don't think you're big enough to play against the mega-gorillas on their turf by micromarketing. You need to get consumerized.

Implementation Rule 2 – Discovery - Consumer convenience means at their desktop.

Consumers evaluate when it suits them using source they trust. That means they read and listen, most often from their desktop. They don't want a face-to-face hard sell. This means the discovery phase is critical to get above the noise of the crowd. We've found search engines, blogs, and traditional PR to be the most effective channels. This has a number of impacts on what you do and don't do and how you measure success. Open source marketing must also be open and trust-based not big budget-based. Users don't trust ads. We don't advertise or use paid-for AdWords. In that game the gorillas can just outspend you. Your advantage is credibility and trust – use it wisely.

PR is traditionally measured by press coverage. Open source is a rapid consumer-driven closed loop. So it's easy to measure the impact of PR through:

- Traditional coverage
- Blog coverage
- Alexa
- Hits on your website (we've seen massive 300% spikes in traffic)
- Downloads
- Trials

There's an interesting story about how one customer chose us. They searched for "open source documentum" on Google and found Alfresco.

Implementation Rule 3 – Research & Try - Don't sell to me. I can make up my own mind.

Consumers don't want a face-to-face hard sell from an enterprise salesman to demystify complex space and complex product. Users want easy access to all the information they need to make a decision. This doesn't mean a face-to-face roadmap presentation after a non-disclosure agreement (NDA) has been signed. Users want to be able to join a community for either information or code that includes access to a demo, a trial,

About the Author

Dr. Ian Howells is chief marketing officer of Alfresco and has more than 20 years of enterprise software marketing experience in the fields of content management, service-oriented architectures, and relational database systems. Ian earned a PhD in distributed databases from University College Cardiff. He has long been on the forefront of technology and marketing, holding early positions at Ingres, Documentum and SeeBeyond. You can read Howell's thoughts on open source marketing at http://blogs.alfresco.com/ianh

> If you deliver a better product for a tenth the cost the impact is 10 times larger on the industry than your revenues **but more importantly you're transforming the industry and making the old model redundant.**

a download, a roadmap, the documentation, and technical tips. This is all that's required for a consumer-driven, self-service decision where the consumer independently qualifies himself as opposed to the salesman.

Implementation rule 4 – People like to talk but not face-to-face.

When an enterprise consumer has decides to go further with enterprise software he wants to talk to a human being not an e-store. This means he will e-mail and ask to be contacted. Salespeople are very important – but over e-mail, the phone, and GoToMeeting – not flying round to big customer meetings.

Implementation Rule 5 – The open source machine and the dinosaur.

Open source is a new world with discontinuity in the business model, marketing model, and product development model. This means that you need a new "open source machine" to cope with the number of people downloading your software, asking questions, accessing your website, accessing demonstrations, trying the product, discussing it in forums, updating the wiki. This is massive compared to a traditional software start-up. The extended infrastructure has to be able to support contributions, bug reports, and fixes from other individuals and companies, take feedback from forums and surveys, and support hundreds of thousands people downloading your software. Amid this, you have to be able to identify those who want to buy support, patches, and updates for a mission-critical environment and those who want to use open source as part of the community. Open source companies have to be masters of the whole open source software value chain to support the massive growth potential.

What's critical is that a new model and new machine is required to support the new price point. The enterprise dinosaurs can't compete with this. Even when they win business it's often at a loss since they have an old model and an old high-cost enterprise sales machine. Over time this becomes unsustainable. Typical short-term tactics are:

- Give the software away
- Bundle multiple pieces of software together to reduce the sales costs
- Stop investing in innovation

In all cases the customer loses and pays in another way – there's no free beer.

Implementation Rule 6 – Think global, act global, be electronic.

They used to say, "Think Global, Act Local." The advantage of an open source blue ocean is it's global. That means a focus on micro-segmentation and micromarketing wastes precious resources and behaving the way a mega-gorilla does.

It means that in the discovery phase enterprise consumers have to be able to access all relevant content easily, attend webinars, and listen to podcasts over the Web. Local now means local time zones and, where necessary, the local language. Gone are the days of hearing, "Things are different in my country…"

The Open Source Impact– A Disproportionate Effect on the Market

Open Source Impact Rule 1 – Be a low-cost machine.

Creating an enterprise software company, or to a certain extent a SaaS company, is expensive, driven by the need to invest in a large sales force or a large infrastructure. Open source companies have a distinct cost advantage that shouldn't be squandered. Open source is not about creating another traditional enterprise software company. Open source is about innovating through a business model and a marketing model and creating a new company with a low-cost model for product development, marketing, and sales that delivers much greater value to the customer. Organizations need to use the principles detailed here to create a low-cost machine. Otherwise the advantages will diminish as the company grows.

Alfresco was built with significantly less VC funding than traditional enterprise software companies.

Open Source Impact Rule 2 – You're a barbarian at the gates. Your impact is far-reaching and disproportionate.

By using a low-cost machine effectively and delivering a new level of value to the customer you have a disproportionate effect on the industry. If you deliver a better product for a tenth the cost the reality is the impact is 10 times larger on the industry than your revenues but more importantly you're transforming the industry and making the old model redundant. This is what the Model T did to the horse and buggy and what Apple did for the masses.

Open Source Impact Rule 3 – This is a gorilla game. You don't want to be a chimp.

Geoffrey Moore was always my marketing hero and in *The Gorilla Game* (1998) he makes some points that are particularly applicable today and to working in "open source time." Moore talks about the differences between "applications and enabling technologies" and says:

Enabling technologies commoditize extremely well, allowing them to proliferate into markets far a field from the original starting points and generate a high degree of network effects. These in turn put pressure on the overall marketplace to standardize exclusively on a single set of components driving market shares to extraordinary levels.

Open source is key driver of commoditization and a modern example of the network effect. These principles explain why many of the leading open source companies like MySQL, Red Hat, and JBoss are infrastructure companies. They can be commoditized more easily and benefit more from the network effect. The traditional enterprise software market moved from hundreds of suppliers in a space to a pack – like with Oracle, Ingres, Informix, and Sybase – to a gorilla emerging form the pack like Oracle did. Here, being a number two was still good. In open source the network effect and commoditization happens much more rapidly - in open source time. That's why in many open source categories there's one clear leader very quickly.

Being an infrastructure software company in a commoditizing market with a massive network effect Alfresco has benefited from these principles. That's what's made it the clear leader in its segment in just over a year.

In open source you get a number one gorilla quickly. The GPL accelerates commoditization and the network effect. A number two in the enterprise open source world is a much more distant number two than in a traditional enterprise software.

Open Source Impact Rule 4 – You can't be half pregnant.

To quote from Blue Ocean Strategy, "Monopolistic practices, by consuming more of society's resources, also incurs a deadweight loss for society at large. Monopolistic practices, therefore, are achieved at the expense of consumers and society-at-large."

Open source is the biggest shift in the market this decade and market shifts are what the old guard hate. It destabilizes their ability to set high prices at will. Tim O'Reilly recently remarked:

Does Microsoft's claim that free and open source software infringes on 235 Microsoft patents remind anyone of Joseph McCarthy's famous claim about communists at the State Department? Whether or not it's true, citing such a number without providing any detail is such a classic FUD move that, to me at least, it just makes Microsoft look ridiculous. More recently, it's reminiscent of the bluster of the SCO case against IBM.

As they say you can't be half pregnant and you can't be half open source and half proprietary either. Open source companies partnering with Microsoft are siding with a company trying to kill open source – the reason they're in business. The open source community recognizes this and these firms risk becoming outcasts. In a separate article and blog – "The Open Source Barometer" – I wrote about the impact of partnering with Microsoft and posed the question, "Is partnering with Microsoft good business for an open source company?"

It's no longer compelling enough just to be the "open source alternative." In a rapidly maturing market, open source software companies must find new ways to differentiate and position their solutions relative to existing products. It takes a combination of technological innovation and marketing innovation to win today; leading open source companies have done both. It's about offering dramatically higher value and innovating, first by recognizing commoditization in the infrastructure and second through simplifying use, installation, rollout, and scalability. This lets you roll out to folks who have previously been denied your technology. A new model for development, business, and marketing is underway. Embrace it and don't try to compete with old companies in old ways. The key is to stay anchored to a philosophy that works for your company while being aggressive enough to challenge traditional ways of thinking. If you do, the impact will be large, far-reaching and industry changing.

"Open source is about innovating through a business model and a marketing model **and creating a new company with a low-cost model for product development, marketing, and sales that delivers much greater value to the customer.**

The Ingres Icebreaker BI Appliance
Powered by JasperSoft

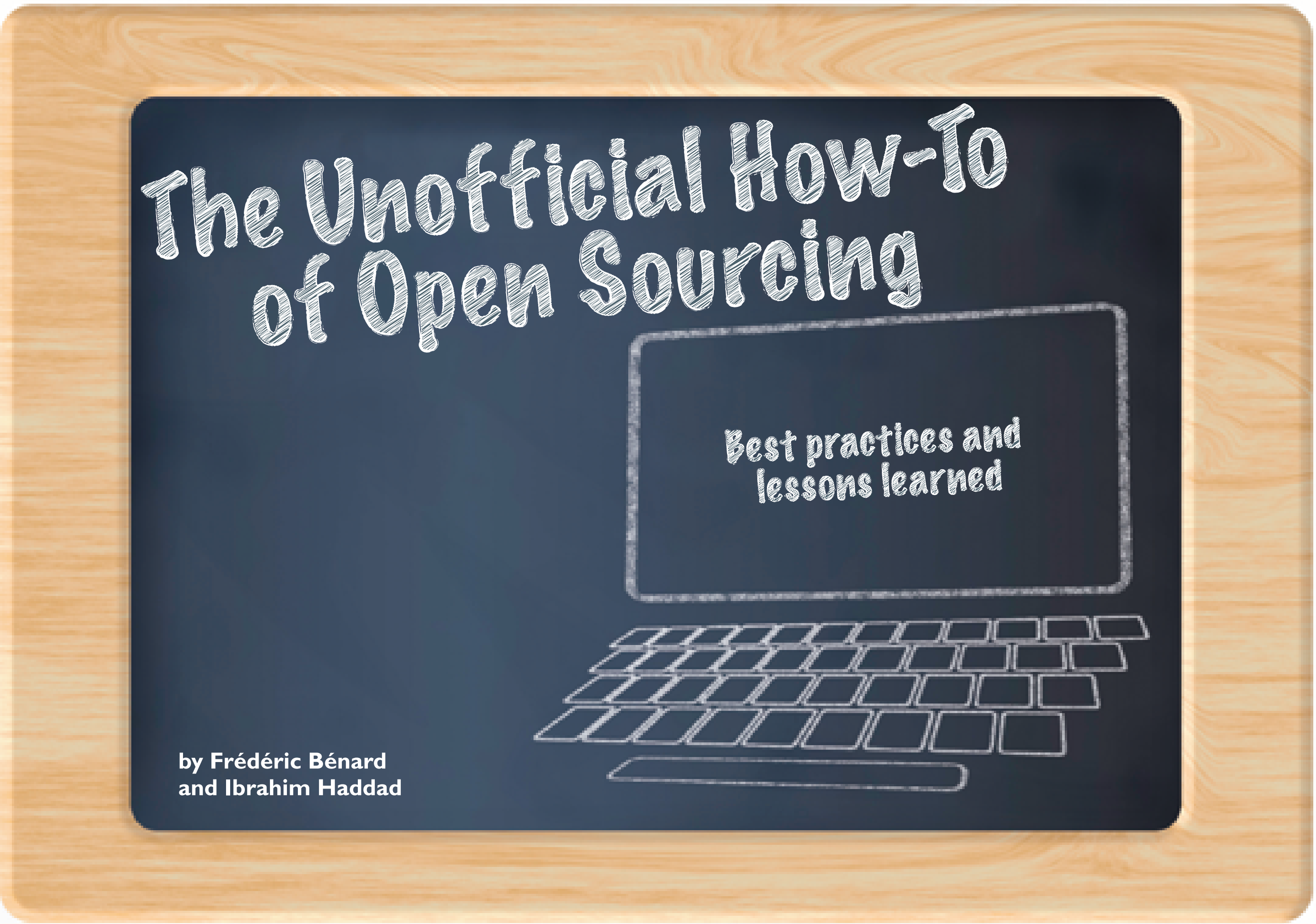
Ingres and JasperSoft have combined our substantial expertise in open source technology and have collaborated to redefine the way Business Intelligence solutions are delivered and supported. In partnership, we have developed the Ingres Icebreaker BI Appliance, the first open source application-specific software appliance.

Available now, the Ingres Icebreaker BI Appliance is a pre-integrated open source software stack featuring the Linux operating system, the Ingres database, and JasperSoft's advanced Business Intelligence Suite. Specifically designed for BI, the Ingres Icebreaker BI Appliance is installed with a single installation process, and is maintained with a single, integrated maintenance stream, eliminating the need to apply separate operating system and database software maintenance. What's more, Ingres is now taking the ability to get up and running with a robust hassle-free BI system to a completely new level by offering targeted BI solutions purposely built to address the specific needs within key verticals, from manufacturing to financial services to government.

Combined with Ingres' world class Support Services, customers can get personalized, proactive, enterprise-class service 24x7x365 whenever they want, wherever they are. What could be easier?

www.ingres.com/biappliance

INGRES®
BUSINESS OPEN SOURCE



Open source software is shifting the software industry into a new paradigm, moving from developing proprietary code behind closed doors to developing code that can be shared, modified and redistributed openly. Key benefits associated with this shift is reducing development cost and software components complexity, developing re-usable common-off-the-shelf software assets, while increasing flexibility and using common enablers. Organizations that embrace the open source model and follow it when it influences positively their ways of building software, will increase their chances to retain their competitive advantage. In this article, we review some of the best practices to follow when taking a proprietary technology and making it open source.

About the Authors

Dr. Frédéric Bénard is Engineering Manager at Motorola and leads the Open Source Software Center of Excellence, which is part of the Motorola "Embedded Systems, Open Source and Linux Technology Group". He holds a B.Sc. in Physics from McGill University, a M.Sc. and a Ph.D. in Physics from the University of Toronto, and an MBA from McGill University.
frederic.benard@motorola.com

Dr. Ibrahim Haddad is Director of Embedded & Open Source Technology at Motorola. Prior to Motorola, Dr. Haddad managed the Carrier Grade Linux and Mobile Linux Initiatives at the Open Source Development Lab (now the Linux Foundation). He received his Ph.D. in Computer Science from Concordia University in Montreal, Canada.
ibrahim.haddad@motorola.com

1. Open Source for the Right Reasons

To be successful in open sourcing a project, you must have the right reasons or motivations. In a previous article published in Enterprise Open Source Magazine, we discussed the top good reasons to open source proprietary software, which included:

- Providing a reference implementation to a standard
- Ensuring that critical software remains viable
- Ensuring that new features are implemented
- Taking control of your own destiny
- Undercutting the competition
- Commoditizing a Market
- Partnering with others and promote goodwill for your company in the developer community
- Driving market demand by building an ecosystem

- Offering your customers the ability to support themselves and add their custom features

Open sourcing for the wrong reasons will not support your goal and will have a negative effect on your progress and relation with the open source community.

2. Legal Review and Understanding Intellectual Property Implications

The second step in the process is to audit the code base which you plan to open source thoroughly to verify that all of the code is owned by you or, in the case of open source packages that might be included in your code, that you have the appropriate distribution rights. This audit may be partially automated with the use of scanning tools that are commercially available today.

Furthermore, you must also evaluate if any of your intellectual property (IP) will be released as part of open sourcing the code. Note that, in the case of a large company, one division may not be aware of the IP from another division. In that situation, it is important to have your open source project reviewed by a group that is familiar with all of your company's IP.

3. Select an Open Source License

The adoption of a simple, well known and popular open source license will go a long way to encourage community participation in your project. Therefore, instead of creating your own open source license, it is most preferred that you use an existing license already approved by the Open Source Initiative (OSI). A list of OSI approved licenses can be found at http://www.opensource.org.

It is highly recommended to involve your company's legal department in the license selection process. In all cases, there are some general guidelines that we present below:
- Contribution to an existing open source project must follow the project's license.
- Contribution to the Linux kernel must be released under the GNU General Public License (GPL) Version 2.
- Creation of a new open source project requires choosing a license that matches your business goals, and preferably use an existing and OSI approved open source license.
- If you want all future derivative work of your contribution to be distributed in source code format, then the GPL is a logical choice.
- If you care contributing to a library, then you might consider the GNU Library or "Lesser" General Public License (LGPL).
- If you want your contribution to be usable within both open source and proprietary (commercial) products, then you might consider the BSD license or the Apache 2.0 license.

4. Train Your Employees

Companies that use and participate in open source projects are highly recommended to provide open source training to their employees. There are specialized companies that offer such educational services or can help your company create and tailor specific open source courses based on your needs. Most common topics covered in basic open source training include:
- General open source concepts
- Open source licenses
- Risks associated with open source software
- Your company's open source policies and compliance rules
- Open source development model
- Integrating open source software within your software development model
- Working with the open source community

Working with the open source community is very different from the traditional corporate development environment and has a different process and set of values from traditional proprietary development model. Training will help bridge the gap and will educate your employees on the working methods of the open source community.

5. Build the Open Source Project Infrastructure

As part of the open sourcing process, you need to build the infrastructure for your project which allows the project to be visible to the outside world and offer communication and software development tools to the project team and to the open source community. A typical project infrastructure includes a web site, a code repository system, one or more project mailing lists, a bug tracking system, a release or patch tracking system and a feature request tracking system.
- *Web site*: The web site of any code contribution must specifically:
  - describe the purpose of the contribution
  - explain the problem the contribution solves
  - explain how the contribution works
  - describe the benefits of the contribution to the common user
  - provide test cases and test scripts so that open source developers can experiment with the contribution and see its benefits
  - advertise news related to the project
- *Code repository system*: Two popular code repository systems are Concurrent Versions System (CVS) and Subversion (SVN). Select one and populate it with your source code. It is also useful to define a coding standard and enforce it.
- *Mailing lists*: You need to host one or more project mailing lists. For instance, your open source project might require a mailing list for developers involved in the project (whether they are your company's developers or external developers from the open source community), another mailing list for the user community, and possibly a third one for quality assurance or software testing. The goal of having more than one mailing list is to keep discussions focused. Typically, the core team of the project would be subscribed to all mailing lists, while other contributors would be subscribed to the list that is of most interest to them.
- *Bug tracking system*: A bug tracking system allows individuals or groups to report bugs against your project; it allows you and others to keep track of outstanding defects.
- *Release or patch tracking system*: A release tracking system allows individuals or groups to keep track of your project software releases or patches.
- *Feature request tracking system*: You should define a process for users to submit feature or enhancement requests and select an appropriate request tracking system.

SorceForge.net offers this entire infrastructure for open source projects, for free. Thousands of open source projects are currently hosted on that site. As an example, you can visit http://sourceforge.net/projects/ppacc/, the web site for a Motorola contribution to the Linux kernel called Precise Process Accounting, and examine all the features provided by the web site.

6. Announce the Project

Once you have created your project infrastructure, you are now ready to announce the project to the world and invite people to provide feedback and contribute to the project. The primary method used to announce your project is to send an email to the relevant mailing lists. Projects can also be announced at conferences, through press releases, or through articles published in Enterprise Open Source Magazine, or any other Linux focused publication.

When making the announcement via mailing list there are some general guidelines to respect:

- Use subject line "[ANNOUNCE] X" to announce the contribution where X is the name of the contribution
- Give some background and introductory text
- Include motivations for your contribution
- Explain how your contributions is different from existing or similar code
- Explain how people will benefit from it
- Don't attach any documents in the email; instead point to a web site.

7. Follow Open Source Development Model and Community Practices

Now that you have announced your project to the world and hopefully have attracted the attention of open source developers, it is your team's responsibility to respect and follow the open source development model and open source best practices. Below we outline the most important practices:

- *Listen to the open source community*: The feedback received from open source developers over the project mailing list can sometimes be negative. Don't worry about it. It is important to review the feedback and understand what the developers are trying to say. It is best not to take such negative feedback personally. After all, the intention is to improve code quality through intensive review. Take the good suggestions you receive and incorporate them into your code. If there are solid technical reasons why the suggestions are not valid then explain those reasons over the mailing list.
- *Embrace code reuse*: Open source developers promote and encourage the development of reusable software. In line with this practice, if someone has already implemented the capability or feature you need, you can use it and build on top of it. In the event that you are starting a new project from scratch, keep software reuse in mind and develop code in modules that can be used by others and by you without many modifications.
- *Be open*: It is important to be open in terms of disclosing problems, bugs and challenges. This openness is much appreciated and is also expected by the open source community, who will help you with immediate workarounds and fixes.
- *Release early and release often*: Open source projects tend to make software releases available early to the user community and then issue frequent updates as the software is modified. This practice is called "release early, release often". The open source community believes that this practice leads to higher-quality software because of peer review and testing by a large base of users who will report bugs and contribute fixes. A side benefit of having many people looking at the code is that the code is reviewed for adherence to coding style; fragile or inflexible code can also be improved because of these reviews. Furthermore, this practice allows the release of small incremental changes that are easier to understand and test.
- *Follow community coding style*: The open source community follows a strict coding style to make it easier to understand the code, review it and revise it quickly.

8. Be Visible

It is important to be active and visible not only when you first launch your project but throughout your project's lifecycle, as this will allow you to rally an increasing number of contributors. You can write articles about your project in open source magazines such as Linux Journal, Linux Magazine, Linux Planet and Open Source Enterprise Magazine; you can attend and present at open source community events and conferences such as the Ottawa Linux Symposium and Melbourne's linux.conf.au conference.

> As part of the open sourcing process, you need to build the infrastructure for your project which allows the project to **be visible to the outside world and offer communication and software development tools to the project team and to the open source community.**

9. Be a Good Open Source Citizen

Being a good open source citizen starts by being part of the open source community, contributing to the community, following and respecting its practices and processes, and leading by example, i.e., getting things done by doing them.

Conclusion

This article reviewed some of the best practices to follow when open sourcing a proprietary technology. Many companies have tried to go the "open source way". Some have bailed after failed trials; other companies are not doing so well and others succeeded and are being regarded as raw models for working with the open source community. It is your responsibility to learn, understand and follow the working methods and best practices of the open source community.

In a follow up article we will describe how we went through the described process to open source a contribution from Motorola to the Linux kernel called the "Precise Process Accounting", which is available today from: http://sourceforge.net/projects/ppacc/.

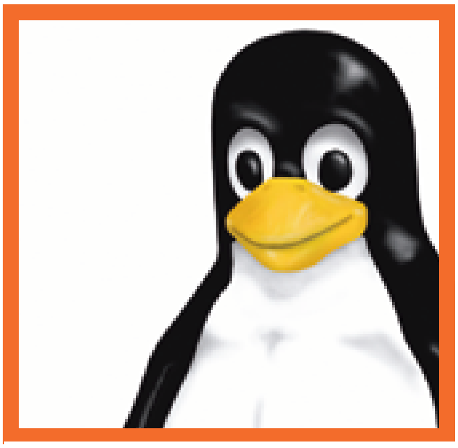
Stay tuned!

Working with the open source community is very different from the traditional corporate development environment **and has a different process and set of values from traditional proprietary development model.**

Linux as a Server Operating System in the Enterprise

A readiness checklist

by Ramnath Cidambi



You may have heard statements such as Linux is inexpensive, reduces TCO and dependency on vendors, and is easy to maintain, from promoters of Linux in the enterprise. But, for those who still haven't made the jump – this article discusses implementation hurdles, cost of implementation, and a readiness checklist. If you're part of a non-technology company, this article is a must-read.

Linux has matured from the days when Linus Torvalds created the first version. Numerous for-profit and not-for-profit organizations have customized it to their needs, deployed it on appliances, and sold modified versions. Novell and Red Hat have developed significant revenue streams from Linux.

Linux has been a panacea for technology start-ups. But, outside e-commerce companies that deploy hundreds of images of the same platform, it hasn't been the breadwinner for technology product companies that are trying to sell their wares to corporations.

But, now let's get to the mature enterprise that's dependent on numerous packaged applications (SAP, Siebel, etc.), software products (Oracle, WebLogic, etc.). Each of these software providers typically has a different reference platform and it's typically not Linux. These software products are also not available on "your" flavor of Linux on the day the new software version or upgrade is generally available (GA). What does this mean to you? Either you have to wait six months for the vendor to make the product available or convince the vendor to provide you a port on "your" flavor of Linux.

I've been part of implementations where software vendors have tried to pass on ports on Red Hat Linux for SUSE Linux (or vice versa). Usually, the engineers installing/configuring the software have to waste hundreds of hours troubleshooting before they realize that the software provided was built on a different flavor of Linux.

Where To Use Linux and How To Introduce It

- Clearly define where you want to use Linux. Start with systems that are least risky.
- Start off on greenfield projects that have high transaction volumes and require a very large installation of hardware, such as a cluster of more than 25 servers.
- Ensure every vendor that you purchase software from for the project has a common denominator in terms of flavor of Linux (Red Hat, SUSE, etc.). Select the common denominator as you choice of vendor-supported Linux.
- Make sure that every software vendor supports the selected flavor as a Tier 1 platform, i.e., it's one of the operating systems on which they release the product when they GA an upgraded version, patch, etc.
- Evaluate the hardware that you'll use to implement Linux. Make sure that the CPU selected is supported by the Linux vendor. Make sure the hardware vendor has certified your flavor of Linux as being able to run on the hardware.

About the Author

Ramnath Cidambi is a technology professional with over 15 years of experience. He currently manages the Unix/Linux and middleware environment of a large corporation. He has significant experience implementing various enterprise systems on Linux, and has presented case studies and papers on the topic. He's also a contributor to various technology publications.

- Consider the following:
  – How is high availability going to be supported?
  – If SAN connectivity is needed, how is it going to be established?
  – How are you going to create a certified image of the software stack and how you are going to replicate the stack on new servers?
- Do a complete proof-of-concept of the stack – hardware, OS, clustering software, operations software (Openview, BMC Patrol agents), applications to be implemented on the platform.
- Do stress or load tests on the proof-of-concept stack.
- Do a TCO analysis for the stack. Remember maintenance is the biggest cost. Compare it to an alternate (such as a platform you use now). You could just do it on the software/hardware maintenance cost.

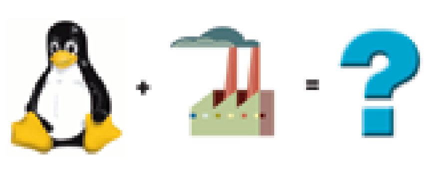
Are you ready and still convinced that you have to go the Linux route for your project?

Here are other things necessary for a successful implementation.

- Recruit skilled Linux engineers who have exposure to the software stack. Make sure they know the flavor you've selected.
- The less dependencies the better, i.e., fewer vendors to depend on.
- Make sure you buy support for the Linux flavor you use. It's not worth using the Internet for support.
- Get every vendor together at least every other week for the duration of the project to ensure implementation issues are solved in a team setting. Sometimes you may have to be the mediator for the vendors to talk to each other!
- Create an operational support structure and processes that will resolve incidents in a timely manner.

Some Tips

- It doesn't make sense to blindly use Linux everywhere. If you go this route, you're setting yourself up for failure.
- Don't bite off more than you can chew!
- Don't believe vendor promises – they never come true. Get written commitments.
- Typical cost of ownership is significantly higher in the first few years!
- Remember bleeding edge is not good for an enterprise whose sole purpose is something else (other than technology!).



Measuring Success

- The TCO of a Linux-based stack should be less than the TCO of an alternate OS-based stack over a three-year period. You should get at least a 25% cost saving. If not, it isn't worth your effort.
- The availability of applications on the Linux stack versus the availability of a similar system on an alternate OS-based stack for a consistent period.
- You've been able to keep your system upgraded and patched and vendor support has been similar to what you'd get for software on other operating systems.

Projects/Migrations That May Backfire

If you have an application written solely in C/C++ hosted on a non-Linux environment such as HP-UX or AIX, and it uses vendor libraries; you may be in for an unpleasant surprise during migration. Don't underestimate these kinds of migrations. The unknowns are many – the compiler you typically use will not be available in the Linux you plan to implement.

If a vendor supports only one flavor of Linux, and it's not the one you've selected, be cautious. Even if you force the vendor to support your flavor of Linux, the level of support isn't going to be the same.

Customized applications written on top of application servers are a very unusual case. The application server vendor may support a flavor of Linux but the application software vendor may not. This is usually because of system calls they may make. Get everyone to support their product on the flavor of Linux you're going to implement.

Summary

Implementing Linux in your data center isn't a non-event. It's an important milestone that has to be carefully orchestrated, its value measured and then implemented. If done well, you could derive significant cost savings for some expensive systems. Large e-commerce Web sites have been popular success stories. Using Linux for the appropriate application in a well-planned implementation will drive you to success.

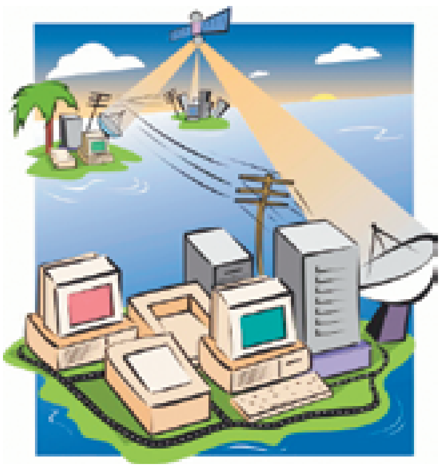
You should get at least a 25% cost saving from a Linux-based stack. If not, **it isn't worth your effort.**

No Open Source Application Is an Island

Keeping the multibillion software giants at bay

by Michael Harvey

The stakes are high. As an Open Solutions Association (OSA) board member and executive VP of Centric CRM, my number-one fear is that the creativity and dynamism of the past decade could come to nothing if open source application vendors don't stand together and take a collective position. As isolated entities they run the risk of getting picked off, one-by-one, leaving an industry landscape dominated by multibillion-dollar giants like Microsoft and Oracle.



To avoid such a scenario, the major challenge facing open source vendors is to move beyond a community of enthusiasts and make open source applications a credible option at both the enterprise and SMB level.

There's no doubt that the initial success of open source software development and adoption — what me might call "Open Source 1.0" — has been one of the most important phenomena in the IT industry of the past decade. It has consisted of the emergence and widespread deployment of various open source software solutions, primarily at the operating system and middleware levels.

At the application level, however, the open source phenomenon is still in its early stages and the market is still very small and fragmented. There are many competing business models with little consensus as yet as to how best to commercialize open source software development.

Much of the energy and headline-grabbing buzz around open source applications has come from relatively loosely structured communities making use of technologies that in the view of many observers may not be appropriate for widespread adoption by enterprise-level businesses. The very earliest releases of applications based on such technologies have appeared in the marketplace in the past year or two. The impact they will make on the SMB market in particular is unclear to say the least.

Open Source 2.0: Reasons To Be Cheerful

One major source of optimism is the recent emergence of what we might call Open Source 2.0. These are open source applications that have been conceived and developed from the ground up as enterprise-class solutions. Such applications are architected with the security and scalability requirements of commercial enterprises in mind. In general they have been developed by serious companies for serious companies.

True, their business models and licenses vary considerably. However, there are also some striking similarities among this latest crop of solutions. Notably, many have been developed in Java and are primarily aimed at the larger end of the SMB market. All these companies share the common trait of providing serious enterprise-ready commercial open source solutions. This emergence of genuine commercial open source applications and vendors is the main driving force behind Open Source 2.0. Their ranks include Openbravo, Hyperic, JasperSoft, Centric CRM, Adaptive Planning, and Talend in fields covering ERP, network/resource monitoring, business intelligence, CRM, business performance management, and data integration. In addition, closely allied companies such as SpikeSource provide support and infrastructure management, while integrators such as Unisys and CorraTech help customers stitch these products together.

All of these companies, and others like them, have banded together in common cause in the Open Solutions Alliance. Furthermore, this list of vendors together represents horizontal solutions that could be logically grouped together to solve a broad

About the Author

OSA founding member and executive VP of Centric CRM, Michael Harvey reviews the market and asks whether, despite the buzz surrounding the success of Open Source 1.0, lack of interoperability is putting the market at risk of being run by a handful of multibillion dollar enterprises.

> I believe the time is ripe for the emergence of a new type of open source community, **something we might call Community 2.0.**

array of business problems. There is also a nice international flavor to this list, with vendors from the U.S., Europe, and Australasia represented.

The Open Source Advantage

A key value proposition for open source solutions is the so-called ecosystem that can form to deliver compelling customer value. Unlike the closed model of, say, a Microsoft, not all solution components need be provided by the same vendor. The open nature of the products — in theory, at least — allows for a level of integration typically difficult to achieve with proprietary systems from competing vendors.

A major challenge for open source providers is to build an application that simultaneously embodies the spirit of open source while fulfilling all the requirements necessary to be seriously considered ready for the modern enterprise.

This means solving a number of potentially conflicting requirements. On the one hand, it must appeal to the open source community, be accessible, understandable, inexpensive or even free, and run entirely on an open source infrastructure. On the other hand, however, it must be built on an enterprise-class architecture, support large installations, and be capable of running on any application stack the customer specifies with little or no change.

From Religious Enthusiasts to Commercial Realists

To achieve products with this kind of balance requires a fundamental change in culture in the open source community, a change that is occurring as we speak. In the same way that open source product development moved from Open Source 1.0 to Open Source 2.0, so the open source community needs to grow up and transform itself from a loose agglomeration of well-meaning enthusiasts to a well-organized band of realists sharing similar economic objectives.

I believe the time is ripe for the emergence of a new type of open source community, something we might call Community 2.0. Such communities will seek to marry the advantages of open source development — large groups of developers improving the code, viral spread of solutions, and the essential appeal of open source products — with a commercial business model.

This new breed of community will be assembled and driven by vendors motivated by economic considerations. The vendors in such a community will likely share a common set of concerns, such as competing with Microsoft on a general level. In their own category they will compete with high-end proprietary vendors such as Oracle, Siebel, SAP, Cognos, and OpenView. In terms of lower-end offerings they will share rivals from other open source solutions such as Nagios, SugarCRM, Compiere, and standalone desktop applications such as ACT! And Goldmine.

Other shared concerns include preventing code forking, the need for effective distribution channels, and the ability to convince enterprise customers to accept open source licenses and technology. Delivering professional service and support, creating vibrant, growing communities, and the need to drive PR and marketing campaigns are of equal importance.

Perhaps most pressing of all will be developing and promoting effective interoperability solutions between various open solutions by rationalizing disparate data models, support offerings, and licensing schemes.

Given these shared set of concerns, and the fragmented nature of the enterprise open source marketplace, significant benefits will accrue to vendors that can come together to work on common objectives. The benefits of cooperation include the generation of "critical mass" for a variety of situations; the "heft" to compete against Microsoft; the stronger appeal for large OEM and distribution deals; and the greater allure for significant partnerships with large infrastructure providers. The benefits to customers of close cooperation are obvious: best-of-breed open solutions that deliver both the benefits of integrated product suites and of open source solutions.

Enter the Open Solutions Alliance

The Open Solutions Alliance was formed last November in a bid to achieve the benefits listed above. Fundamental to our working relationship is an understanding that participating vendors need to be fundamentally aligned in their business objectives and outlook. An evolutionary approach to assembling the alliance and tackling its common objectives is in place to reduce "implementation risk." The OSA has already achieved some impressive goals like recruiting leading ISVs, integrators, and even customers to its ranks. It has published several roadmaps outlining best practices and implementation approaches for integrated offerings. It has hosted several successful end-user events in a variety of settings. And at LinuxWorld, in August 2007, unveiled several working interoperability prototypes that include common customer view and single sign-on. This early success is in large part due to the establishment of high levels of trust and a sense of shared mission among members. In short, transparency and an "open ethos" prevail and help to ensure our success.

Next Steps

To make the initiative successful, each vendor will have to make some meaningful contributions. Contributions might ultimately include some or all of the following components: infrastructure (for example, Centric CRM contributed its community management tools for the Web site); financial support; staffing support (for committees or board); mutual engineering support for integration issues; lead sharing; collaboration on core business activities such as VAR recruitment; and making some number of seats/servers "free" for bundling and distribution deals.

United We Stand, Divided We Fall

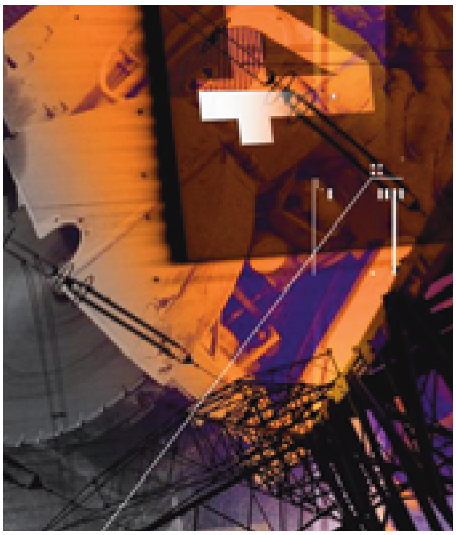
Now is the time for a leading group of enterprise open source vendors representing a broad array of best-of-breed solutions to stand together for mutual benefit. With effective collective action dedicated to improving interoperability and awareness for enterprise-class open business solutions, I believe we have a good shot at keeping the multibillion software giants at bay.

The Fate of the Big Four

Empires under siege

by Harper Mann

Earlier this year, the 451 Group reported that the Big Four systems management vendors (BMC, CA, HP, and IBM) were ripe for a shakeup, and had been for some time. Poised at the top of a healthy, lucrative market, it hardly seems likely that these market leaders could be threatened by newcomers. Nevertheless, they are being challenged and the enterprise IT world is paying attention to the outcome. How real is the threat, and what changes will it eventually bring to the IT infrastructure management software?



About the Author

Harper Mann is a senior engineer at GroundWork, a San Francisco-based IT operations management start-up. He has completed hundreds of IT monitoring and management software installations at customer sites during the last 20 years. Prior to GroundWork, Harper worked as a Linux and Solaris IT manager at E-Loan, and prior to that ran the virtual enterprise data center for Unicenter TNG at Computer Associates. Early in his career, he was part of the team at Cyber Cash that did the first credit card transaction completed on the Internet.

Let's take a look at the lifecycle of an empire. Empires rise and fall with a fascinating yet predictable regularity. At the risk of vastly oversimplifying, we can say that empires come to power gradually, emerging from economic chaos or the ruins of war. They build power through their ability to establish order and achieve prosperity, often with deep investment in military might. Then, at some point, the empire becomes irrelevant. Choices are made not for the common good but with a view to preserving the status quo. Because of this, the empire loses its competitive edge and eventually fades into inconsequence.

It's hard to speak of empires without experiencing a secret glee at their eventual fall, and it's with this same indulgence in *Schadenfreude* that observers in the IT world watch the Big Four companies in systems management contend with challenges to their dominance. This is not to say that they are empires on the verge of toppling; in fact, Gartner reports that the market is $10 billion strong and growing, with the Big Four easily owning 55% of that.

But much has been said about emerging challenges to the Big Four as other large vendors, such as Microsoft, Oracle, EMC, SAP, and Symantec, aim for a piece of the market. And the threats to the Big Four aren't limited to those from proprietary vendors. A growing number of open source solutions have arisen to challenge the Big Four with the promise of lower cost, flexibility, and lack of vendor lock-in. In fact, a key finding of the 451 Group's report, "Managing in the Open: The New Wave of Systems Management" states plainly, "Open source is breathing new competitive life into systems management, ultimately forcing the established vendors to respond in their products, pricing, and strategies."

Nora Denzel, former senior VP and general manager of the software unit at Hewlett-Packard, agrees. Denzel retired from HP last year and now serves as a director or member of several software advisory or boards of directors, including GroundWork Open Source, a company that is counted among the new challengers to the Big Four. "Open source is encroaching in several arenas, but in particular with management software that helps companies manage their sprawling networks," Denzel says. "We haven't even begun to see the full impact of open source in this space."

According to Denzel, open source works because it's easy to deploy and is a small fraction of the cost of proprietary solutions. Further, she argues that open source is easier to use and faster to install owing to its modular rather than monolithic structure. As for the extensive feature packages that the Big Four vendors boast of, Denzel says, "In reality, studies show that over 50% of the features in the proprietary versions are never installed. Think of what features you use — in Microsoft Word, for instance — what percentage of the features do you estimate you use in that?"

Customers who've opted for open source over proprietary tell a similar story. Tom Lamb, CTO of the University of North Carolina at Charlotte, needed to upgrade the school's monitoring system to include every aspect of its complex IT environment. Lamb initially looked at HP's OpenView and Compuware Vantage but discovered the cost of purchasing, deploying, and scaling either system was beyond the reach of his IT budget.

"The initial licensing fees for both OpenView and Vantage were in the hundreds of thousands of dollars and simply far beyond what our budget would allow," Lamb says. "To get the customized dashboards we wanted, we would have incurred additional consulting fees — the enterprise license agreements did not allow our staff to customize these systems. And when we talked about scalability with these vendors, the overall price tags began to approach the million-dollar mark." Lamb eventually selected GroundWork Monitor Professional to do the job.

Another GroundWork customer made the acquisition decision based on the flexibility and control that open source offered. Richard Farley, director of IT at Ingenuity Systems, Inc., a developer of biology interaction modeling software, noted that proprietary systems with the functionality his organization required were too expensive while affordable proprietary packages didn't provide enough functionality. Additionally, proprietary systems came with modification restrictions and Farley wanted a package that could be modified to meet Ingenuity's requirements.

Farley was pleased with his open source solution: "We gained immediate results with the software right out of the box, yet we were also able to make unique configuration changes simply not possible with other enterprise-class tools." Such success stories are not uncommon; similar tales can be found among case studies from other open source system management vendors, such as Zenoss and Hyperic.

There's no doubt that open source presents attractive alternatives to proprietary software on many levels, but how valid is the argument that open source presents a threat to the Big Four? I'd argue that it's not open source itself, but the inherent maneuverability of the new players in the market segment and the necessity of remaining relevant to customers that is forcing the Big Four to react.

You can look at telecommunications companies for an example of an industry dominated by giants that was forced to adapt and change in response to challengers in the field. Few of us believe that Skype will topple AT&T any time soon, but AT&T is surely paying attention to Skype's growing user base, and well it should.

Likewise, the Big Four take a keen interest in the successes of their new competitors, whether proprietary or open source. Each customer they lose sends them a message with regard to feature packages, pricing, or the ability to customize. To stay relevant, they'll be forced to respond, and indeed are already responding with tiered pricing levels, customized feature menus, and open source components. In several instances, they've solved the problem simply by acquiring up-and-coming competitors, eliminating the threat and capturing the capabilities in one swoop.

The pressure to lower costs, innovate, and respond quickly to customer needs is quite real in the IT systems management market and this is good news for customers all around. The real shakeup may not be a question of open source versus proprietary software. Ultimately, innovation comes from listening to the customer's needs and meeting them, and whoever does this best will emerge the winner. But – just as an aside – open source has always excelled in this area!

Resources

- The 451 Group Special Report: CAOS (Commercial Adoption of Open Source) – Report Four: Managing in the Open http://www.the451group.com/caos/caos_detail.php?icid=419
- Denise Dubie, Network World: A Look at the Big Four, Part One http://www.networkworld.com/newsletters/nsm/2007/0604nsm1.html
- Denise Dubie, Network World: Emerging management challengers: what do you think? http://www.networkworld.com/newsletters/nsm/2007/0723nsm2.html

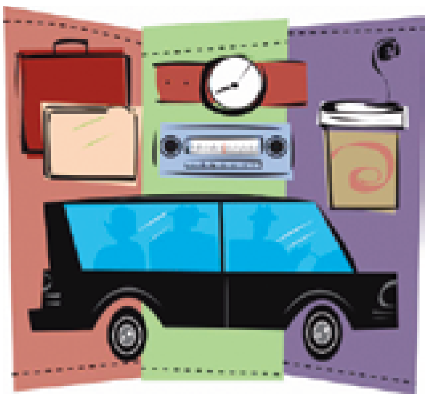
"The real shakeup may not be a question of open source versus proprietary software. **Ultimately, innovation comes from listening to the customer's needs and meeting them, and whoever does this best will emerge the winner.**

Optaros Creates Dynamic Online Community for Chicago Public Radio

What the Internet does best

by Marc Osofsky

Chicago Public Radio (WBEZ 91.5 FM Chicago, WBEQ 90.7 FM Morris, and WBEW 89.5 FM Chesterton) is a community-supported, non-commercial public service broadcasting institution for 7.7 million people throughout Chicago and surrounding regions. Chicago Public Radio produces, acquires, and distributes engaging, thoughtful, and entertaining programs of depth, breadth, diversity, and substance. The organization prides itself on helping listeners learn about issues and ideas that affect the community, the nation, and the world. The institution has more than 120 employees across multiple offices in greater Chicago.



In planning the Vocalo Web site, Chicago Public Radio was breaking new ground. The Web site would not just promote its companion radio station, but provide an avenue through which a diverse audience could be reached, intrigued, and attracted to the radio station, and then directed back to the Internet again. For Chicago Public Radio the project involved exploring uncharted radio and Web territory, with no previous model to serve as a guide. So Chicago Public Radio contracted Optaros to help bring its vision to life.

Chicago Public Radio evaluated numerous companies and selected Optaros because of its online community design experience and the Optaros Assembly Methodology (OptAM) that assembles solutions faster and more flexibly than traditional build, buy, or rent options. Chicago Public Radio recognized Optaros' consulting and development services as critical to the construction of the site and, more importantly, to determine the business strategy of the overall project.

Optaros had extensive experience and success in building online communities that provided a driving force behind this project. Chicago Public Radio got a lot of feedback from Optaros through the design and assembly phase, which was invaluable as Chicago Public Radio learned the intricacies of developing an online community.

Optaros implemented the Vocalo site in an aggressive time frame, leveraging a large range of Internet-based services and open source components. Optaros developed a flexible social networking site that targets all population segments to resonate across age, sex, education, race, and address, while speaking to each individually to encourage repeat visits. The Optaros team was persistent in introducing user scenarios including demographic profiles and helping Chicago Public Radio figure out how each segment interacts with social networking sites. Optaros worked with Chicago Public Radio on many iterations of the site, adjusting the architecture depending on where the vision was leading the team. Optaros helped Chicago Public Radio formulate its strategic direction and then built the site accordingly.

Optaros assembled a flexible solution that accommodated frequent requests for changes as the project evolved. This led to some very dynamic design sessions, with Optaros modifying prototypes in real-time as quickly as Chicago Public Radio requested new features and design elements. The site was developed using the Drupal content management system and the Optaros video Web Service plug-in for Drupal.

Integrated Assembly Environment

The Optaros Integrated Assembly Environment (IAE) provided for a seamless transition from solution assembly to support and ongoing enhancements. The IAE provided Chicago Public Radio all project requirements, Web designs, code, communications, and testing results all in a single environment for easy transition to support a "perpetual beta" approach for continual enhancements.

Optaros' IAE helped Chicago Public Radio build Vocalo dynamically, allowing for continued evolution as it crystallized its vision. The work with Optaros provides a foundation of education and a solid architecture as the online community grows. Given the flexibility that the OptAM allows, Chicago Public Radio is able to adapt continually and cost-effectively to its growing membership and better serve the community.

Available at Vocalo.org and at WBEW 89.5 FM (Chesterton, Indiana), Vocalo reinvents the relationship between a radio station and a Web site. It is no longer the radio station's Web site – it's the Web site's radio station. Rather than standard public radio fare, radio programming consists of a collage of commentaries, personal stories, local satire, music, and other content originally uploaded to the Web site.

The Vocalo site now does what the Internet does best — allowing users to seek the information they need, interact, give opinions, and make their mark as individuals by uploading video, audio, pictures, and text. In fulfilling its mission, Chicago Public Radio created a community and programming that speaks with many voices and reflects the distinctive and diverse Chicago area.

About the Author

Marc Osofsky is the vice president of marketing for Optaros and leads the company's go-to-market efforts and solution development. He has over 15 years of experience in helping businesses leverage new technologies to achieve their business objectives. Marc has an MBA from MIT Sloan and a BA from Brown University.

The Importance of Open Source Governance in Mitigating Risk

Best practices for managing risk when developing an effective open source policy

by Steven Grandchamp

Programmers naturally gravitate toward the best software packages and components for development. They are increasingly choosing a broad range of enterprise-grade open source packages from Apache and Tomcat to Axis and Eclipse. But imagine for a moment this all-too-common scenario: a programmer at a Global 2000 is faced with a looming deadline and after a little bit of research, picks an open source package that he thinks will meet his technical needs and enable him to get his job done more quickly and effectively. Although the open source package may have all of the functionality needed, the programmer doesn't take into account some of the broader, longer-term issues that can have a significant impact on the enterprise, such as:

- How will we get support for this package once it goes into production?
- Will we be able to get the service levels we need for support?
- How might the license impact our company?
- What will we need to do to stay in compliance with the license?
- What IP protections do we need to put in place?
- Will the project still be active over the years to come?
- How will we manage potential changes to source code?

While open source software provides a functional, flexible, and cost-effective option to enterprises, organizations need an effective, scalable policy for evaluating, managing, and governing the use of open source. Govern open source strategically, and you'll find a venerable gold mine of open source solutions at your disposal. Ignore these open source governance issues, and you take unnecessary legal, financial, and operational risks.

Why Open Source Is Worth Considering

The popularity of open source continues to grow as a wide range of innovative open source software components enable business agility and increase ROI. The functionality and flexibility of open source cannot only decrease time-to-market of new solutions, but help extend the life of legacy applications.

Today, companies don't need to give up the value-add services of commercial software when they choose to use open source solutions. There are now a variety of commercial open source companies that offer support, indemnification, and maintenance for popular enterprise open source packages, making it easier to leverage these open source solutions into your homegrown applications.

Making Open Source Successful

As open source becomes more ubiquitous, business and IT executives must identify ways to confidently incorporate a variety of open source packages to meet business demand. Creating an effective open source policy to govern the adoption and use of open source in an organization will mitigate any potential legal, financial, and operational risks and is a critical step towards making open source successful in your organization. As companies use more open source software, they are more likely to create an open source policy. For example, according to a recent survey conducted by OpenLogic, 83% of the organizations polled that are currently using more than 25 open source projects have an open source policy, are developing a policy, or plan to create one.

However, many companies aren't aware of the extent of their open source use. As with our programmer, most open source packages are downloaded by programmers, bypassing the normal procurement controls. Enterprises have often dozens to hundreds of open source solutions deployed without an appropriate level of review.

About the Author

Steven Grandchamp is the CEO of OpenLogic, Inc., a provider of open source solutions that enable enterprises to acquire, support and control open source software. He has over 25 years of experience in the software industry, serving in executive roles at Information Management Research, American Fundware, and was a founding partner of Formation Technologies Inc.

Writing an open source policy will establish a framework for communication between business management, legal teams, IT managers, and developers about how open source will be deployed in their organization. First, a policy can help direct and monitor IT plans by ensuring that investments in IT generate the desired business value and ROI. Second, when a policy is enforced, it will mitigate and manage legal risks including intellectual property infringements and license violations. Lastly, a policy will ensure that companies can continue to meet operational cost and uptime requirements whenever open source is deployed.

You Can't Get Something for Nothing

The first thing to consider when creating an open source policy are the choices your business may face in balancing risk reduction with business demand. For example, though the flexibility, functionality, and quality of open source may help businesses maintain a competitive business advantage; enterprises often consider open source solutions merely to reduce costs.

What business and IT executives need to keep in mind is this: software is software. All software, both open and closed source, comes with responsibilities and requirements that businesses can't ignore. Whether proprietary or open source, management teams must devote resources to developing, deploying, managing, and supporting all of their software assets. Although businesses can cut costs significantly when moving to open source solutions, it's critical to invest in open source governance to avoid unnecessary legal disputes, unexpected costs, or unforeseen operational issues. In other words, you can't get something for nothing. Even though the procurement of open source most often begins with a free download, you need to be vigilant about what software you're using and how it's being used.

How Open Source Software Is Different

Although open source software is "just software," there are a few critical differences you need to consider for governance purposes. First, open source packages carry open source licenses that have unique characteristics. Second, open source packages are typically created by a number of independent authors, which can raise potential intellectual property concerns. Finally, open source is typically procured differently than commercial proprietary software, which may dictate adapting existing processes.

Although the most widely known open source license is the GNU General Public License (GPL), the Open Source Initiative (OSI) has approved over 50 open source licenses. And many free or open source packages use licenses that haven't been approved by the OSI. In either case, legal staff must familiarize themselves with the terms of open source licenses being considered to determine that the license is compatible with the particular use that the enterprise is planning. The enterprise must also put audit and control processes in place to assure that the organization complies with all of the terms of those licenses. There are cases of enterprises paying out-of-court settlements to open source copyright holders due to violations of open source licenses, so license review and compliance is a critical piece of your open source governance plan.

Another major concern of companies using open source is intellectual property violations. Since open source packages are typically created by a number of independent authors, there's some risk that an author might have inadvertently or purposely infringed on another party's intellectual property. There have been a few highly publicized lawsuits or legal actions around intellectual property infringements by open source packages. Although many of those legal actions may be without merit, companies that use open source are concerned about the potential cost of defending these suits. One option for mitigating this legal risk is through indemnification. There are several open source solution providers that offer indemnification for the packages they support, giving clients some financial protection in the case of a legal action.

Finally, when acquiring open source, you'll have to ensure that you have the appropriate procurement processes in place. If you're working with an open source vendor, you'll probably encounter a subscription model or an entitlement. If you're looking to significantly increase the use of open source in your enterprise over any period of time, you'll want to develop a contract process specifically designed to handle subscription-type acquisitions. Above all, remember that your worst enemy is a recurring "don't ask, don't tell" situation where developers assume that using a minor open source component doesn't require approval. Sweeping the acquisition of open source under the table to speed development will impede overall knowledge of what your organization is using and how best to leverage your open source tools.

> “ The amount of risk you're exposed to depends not just on the presence of open source, **but how well you understand and manage open source use in your business.**

Tips & Tricks for Governing Open Source Software

Creating an open source policy to manage risk and govern the use of open source doesn't have to be difficult. With a few tips and tricks for dealing with the main concerns around open source adoption, you'll be able to reap the benefits of an efficient, scalable open source policy that gives you security and confidence in your software.

You can't manage what you can't measure

Though open source can have a rogue reputation, the first thing to recognize when beginning to create a policy for open source governance is that you're probably already using open source. Somehow, somewhere, there's open source installed in your company. However, you can't manage it if you don't know it. If you're not exactly sure what open source software is installed and in production, start by taking an inventory to identify and list all the open source you have deployed. Today, you can even find free tools to scan multiple computers for open source components and provide an inventory of what's installed across your systems. Once you get a comprehensive picture of what your organization is using, you'll be able to analyze how you're currently governing and managing it. Then you can build from there.

Think about mitigating, not eliminating risk

Every business decision you make has risk associated with it. Just as with traditional proprietary software, you shouldn't expect to eliminate all the risk associated with open source software. Instead you need to evaluate the level of risk you're already exposed to and the level you're willing to be exposed to. Instead of throwing up your hands, make informed choices about acquiring open source based on a risk assessment.

Open source governance isn't "once and done"

Like any compliance program, open source governance isn't "once and done." Once you've identified and inventoried the open source installed on your system, keep up-to-date audits for posterity as well as upper management. Your open source policy will also have to evolve over time, so build in the appropriate processes for doing so.

Writing an Effective Policy

Once you understand how you're using open source, you're ready to begin writing an efficient and clear policy. Here are some easy-to-follow steps in drafting a policy to govern the use of open source in your organization.

1. **Define objectives and strategy**

   Your best approach is a pragmatic one. Sit down and ask yourself: What are the legitimate concerns here? What are my organization's goals? Where do these intersect? From the answers to these questions, you can develop specific procedures and controls that are case-specific. Keep all guidelines consistent and simple to foster compliance and to reduce instances of unintentional divergence.

   When communicating your policy don't be afraid to focus on the bottom line. IT teams need to know that business goals aren't only driving development but your open source policy as well. Build your policy to fit the best interests of every stakeholder including all practitioners and influencers.

2. **Divide policy into three distinct areas**

   The policies you'll need to create should address three distinct areas:
   - The acquisition of open source,
   - Governing what you'll do with open source technologies in your organization, and
   - The rules by which you'll interact with the community.

If these three issues don't come up naturally when discussing the tenets of your policy, make sure to throw them in!

Acquiring Open Source

You need to determine guidelines for how you'll review and adopt open source software and then apply these rules to your whole organization. Find the balance between too much control and too little control. For example, your policy might allow developers to freely download open source software for evaluation and experimentation, but require approval at the time when it's chosen for use in a production application.

As part of your policy, you'll have to build in an approval process for using open source technologies in the applications you're building. Whether it's centralized or decentralized, keep in mind that this process has to be scalable – so design the process with efficiency in mind. Whenever possible, make it easy to reuse previous decisions such as "open source package A is approved for use in software that will not be distributed."

Using Open Source

You must also decide how open source will be used in your organization. Determine guidelines for the level of risk you're willing to carry and decide under what circumstances developers are allowed to change and modify source code. You also need to put audit and compliance processes in place to ensure that the approval process is being followed and that you're complying with the terms of the open source license.

Interact with the Community

Ask yourself how your business, as a brand, will converse with the community: will you be active forum participants, contributors, evangelists? Not all organizations interact publicly with the community surrounding a project, so if you decide to become a contributor, make sure your development teams know what can be shared and how you'll structure your contributions. You should also build a base of ready responses to common situations and make sure that all levels of your organization are on the same page when it comes to community attitude and interaction.

3. **Describe behavior to guide behavior**

   The best policy is one that continues to take feedback into account from enterprise developers, as well as the changing legal landscape of open source software. Your policy should ultimately describe what kind of behavior is necessary in response to the open source technologies you're adopting and should align with an overarching organizational stance with regard to risk and a standard value system around software.

4. **Are you using Linux?**

   If you've deployed Linux in your company, what kind of infrastructure exists around it? Can all or part of that policy be expanded to open source? This is a fairly classic example of how many companies come to find Linux as the gateway to tens of thousands of stable open source projects, and then begin to structure their use of open source around their Linux policy.

5. **Bring in the lawyers**

   You will need legal advice throughout the creation and implementation of your open source policy so don't be shy about calling on legal counsel. If you have a bunch of developers writing a policy, when you actually go to enact it, you'll find lawyers knocking down your door with red flags. Avoid this heartache and unexpected legal fees by getting lawyers engaged early and often. Make sure open source policies are based on specific license types and versions and do your research to see what common legal problems have surfaced in the past. If you can anticipate the roadblocks you might potentially encounter, you'll be prepared to write a policy to address issues before they arise and will have a much better outcome.

6. **Don't hide your policy**

   Once you've poured your sweat and tears into creating an open source policy, shout it from the rooftops! Well, maybe not literally. But do make sure that your policy isn't collecting dust and cobwebs in some dark corner of the office. One of the most important things about creating a policy is identifying a central repository for all of the knowledge and information you gather. If your policy isn't easily accessible, it won't empower the individuals who are looking to adopt open source. This might actually impede open source adoption and you might find yourself reinventing the wheel down the road as similar issues arise over and over in the absence of a marked policy to guide development.

7. **Software is software**

As mentioned above, software is software. You'll save time by creating an open source policy that mirrors whatever guidelines are already in place for reviewing, managing and governing proprietary software. By measuring open and closed source solutions with similar metrics, you'll maintain consistency in your overarching software policy. In addition, keeping the same management team and review boards for both open and closed source solutions will save you the time required to coordinate guidelines between two separate governance silos. Depending on your level of open source use, you may be able to simply add an extra layer of rules and guidelines to your existing proprietary software policy for open source cases. With a little time and patience, your policy will hopefully evolve into a nicely branded "software policy" for all kinds of source code, but with an addendum for anything that's unique around open source.

8. **When in doubt, find or create a library**

One of the top challenges to evaluating open source in the enterprise is ensuring that the code is coming from a trusted, stable source. In other words, browsing the 153,000 registered projects among over 1.6 million registered users on Sourceforge.net can be intimidating and overwhelming. Don't fret. By implementing a central library or repository of vetted open source projects, you can immediately reduce risk and rest assured that you're shopping mature, tested, and proven open source code. You can subscribe to a repository from an open source solution provider or start one from scratch. Your repository should contain open source that's pre-screened, supported, and certified for use in your enterprise.

Clearly communicate with your development teams so they know how to access your open source repository, and ensure that the repository holds relevant information about licensing agreements, support options, and approval status. Implementing a central repository will ultimately speed development and help you avoid legal problems.

Conclusion

Creating a policy for governing open source in your business requires diligence, hard work, communication, and ongoing maintenance. Once you've developed an effective, scalable policy, you can incorporate it into your daily development and management practices, and then into your company culture. Your policy will determine how your business as a "brand" adopts, uses, and contributes to projects and the larger open source movement and will establish lines of communication for constant feedback from programmers. Opening channels for feedback between business management and IT development teams will also keep open source use in line with business objectives and responsive to development goals.

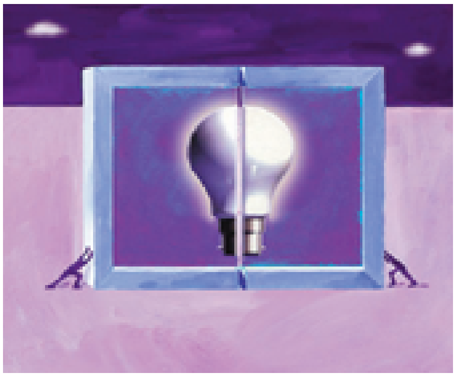
At the end of the day, remember the amount of risk you carry depends not only on the presence of open source, but how well you understand and manage open source use in your business.

Using the Eclipse Data Tools Platform with SQL Anywhere

A new top-level project at eclipse.org

by José Ramos

The Eclipse Data Tools Platform (DTP) is a new top-level project at eclipse.org. Originally proposed by Sybase in February 2005, DTP has attracted strong community support and is currently managed by a committee comprised of Sybase, IBM and Actuate. It is an open source initiative designed to provide solutions in the data framework and tooling domains.



Examples of what the DTP plug-in lets you do from inside Eclipse include:
- Connect to a database (or multiple databases)
- View database object trees (tables, views, stored procedures, etc.)
- Generate data definition language (DDL) from database objects
- Execute DDL commands against a database
- Execute standard query language (SQL) queries against a database
- View query results in table format

To learn more about DTP, resources can be found at http://wiki.eclipse.org/index.php/Data_Tools_Platform_Project.

SQL Anywhere is a suite of databases and data exchange technologies that pack a lot of power and performance into a small footprint. The SQL Anywhere Server database supports 32-bit and 64-bit servers on all major platforms. Many ISVs embed the SQL Anywhere Server database into their applications because not only can it scale from client/server environments with hundreds, even thousands, of users storing hundreds of gigabytes of data, but it can also be easily hidden in an application that runs on desktop and laptop computers requiring little or no IT support. You can learn more about SQL Anywhere by visiting http://www.ianywhere.com/sqlanywhere.

This tutorial shows you how to install the Data Tools Platform Eclipse plug-in and how to use some of its features against a SQL Anywhere 10 database.

Requirements
- SQL Anywhere 10.0.1 or higher
- DTP 1.5M6 End-user or later
- Depending on what version of DTP you download, you will need specific versions of:
  - Eclipse SDK. At the time this writing, Eclipse 3.2.2 SDK was required.
  - EMF runtimes (EMF + SDO). At the time of this writing, EMF 2.2.2 (EMF +SDO) was required.
  - GEF runtimes. At the time of this writing, GEF 3.2.2 was required.
- JRE 1.5 or higher
- Sybase jConnect 6.05 – available at http://www.sybase.com/products/allproductsa-z/softwaredeveloperkit/jconnect.

Before You Begin Downloading the Data Tools Platform

The first thing you need to do is download the DTP plug-in and include it in your Eclipse folder. You may also have to download the EMF and GEF plug-ins if you don't already have them. When you download and extract all the files, you have to merge them into your current Eclipse folder, where you typically launch the application. Here's a link where you can download DTP; it also has links to all of its dependencies (EMF

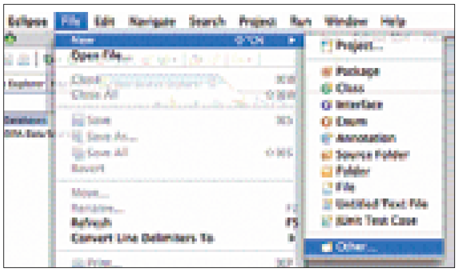


Figure 1

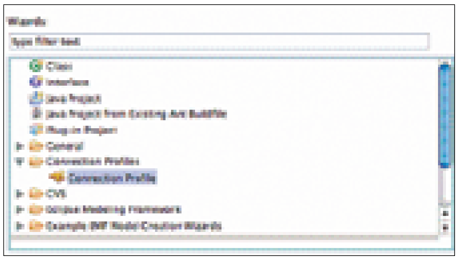


Figure 2

About the Author

José Ramos is a product manager for Sybase iAnywhere and is responsible for the direction and life cycle planning of Sybase iAnywhere products. With more than seven years of experience, he also plays an active role in customer demonstrations and education to ensure the continued success of SQL Anywhere, the industry-leading database management and data movement technology built for frontline environments. José holds a Bachelor of Mathematics in Computer Science from the University of Waterloo in Ontario, Canada.

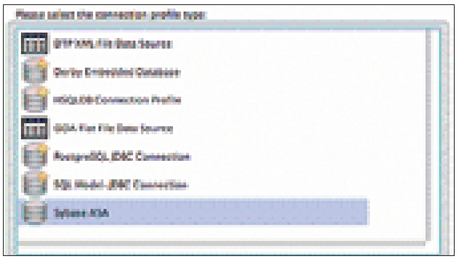


Figure 3


Figure 4

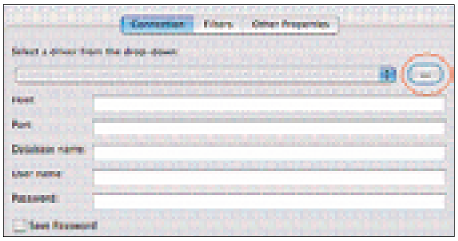


Figure 5


Figure 6

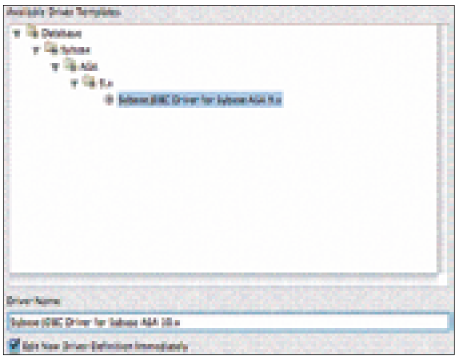


Figure 7

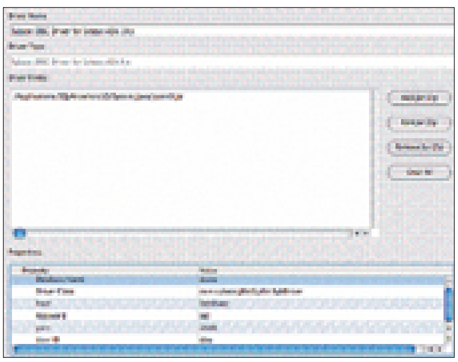


Figure 8

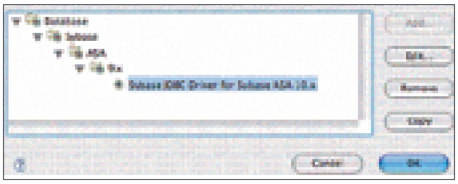


Figure 9

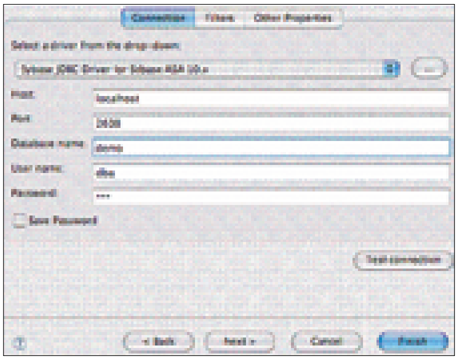


Figure 10

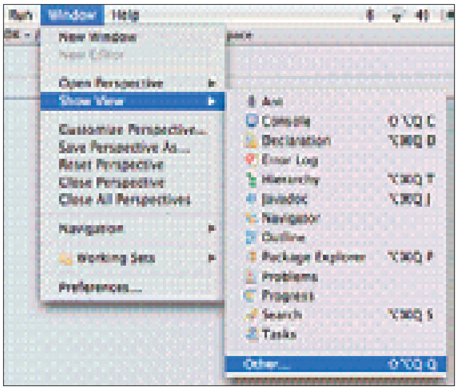


Figure 11

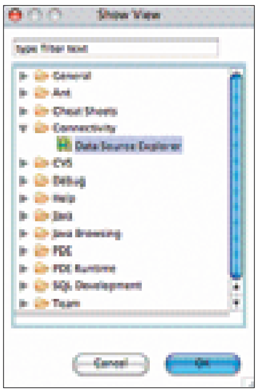


Figure 12

and GEF): http://www.eclipse.org/datatools/downloads.html.

Starting the Sample Database

This tutorial uses the sample database provided with the installation of SQL Anywhere. It's called demo.db and you can find it in the samples folder of your SQL Anywhere installation. If you're running on Mac OS X, Linux, or Unix, you'll need to configure the samples by sourcing the file 'samples/samples_config32.sh' from your SQLAnywhere 10 installation folder. If you're running on Windows, the samples folder is defined in the %SQLANYSAMP10% environment variable.

Stop all the databases running on your machine, find this file, change to its directory, and run the following command in a command prompt: dbeng10 demo.db.

This will start a database engine called 'demo', listening on port 2638 with one database called 'demo'.

Creating a Driver Template and Connection Profile

In this section we'll cover how to define a driver template and create a connection profile to be used to connect to our sample database.

1. In Eclipse, click 'File' | 'New' | 'Other' (see Figure 1)
2. Choose 'Connection Profile' (see Figure 2)
3. Click Next. Specify 'Sybase ASA' as the connection profile type (see Figure 3) (*Note*: The SQL Anywhere Server database was called ASA in previous versions of the product.)
4. Click Next. Give the connection profile a name, type 'sqlanywhere10' (see Figure 4)
5. Click Next. Here we have to choose a driver, but we have no driver templates defined. Click the browse button circled in the image below (see Figure 5)
6. Click 'Add' to add a new driver template (see Figure 6)
7. Choose the 'Sybase JDBC Driver for Sybase ASA 9.x (see Figure 7)
   – Even though we're connecting to a SQL Anywhere 10 database, we can use this driver template and change the appropriate settings
   – Change the driver name to 'Sybase JDBC Driver for SQL Anywhere 10.x'
   – Make sure the 'Edit New Driver Definition Immediately' box is checked
8. After clicking 'OK,' press the 'Add Jar/Zip' button to browse for the jConnect 6.05 JAR file: jconn3.jar
9. Set the driver properties at the bottom of the screen as show in Figure 8
10. Click 'OK'
11. Now that we've defined a driver template, we can return to this screen, choose our new driver, and click 'OK' to continue (see Figure 9)
12. Set the connection profile settings as shown below and click 'Finish' (see Figure 10)

Now that we've created a connection profile, we can use the Data Source Explorer to connect to our sample database and view database objects in the object tree.

Connecting to SQL Anywhere 10

To connect to our SQL Anywhere sample database, we need to open the Data Source Explorer view.

1. Click 'Window' | 'Show View' | 'Other' (see Figure 11)
2. Choose 'Connectivity' | 'Data Source Explorer'
3. Click 'OK' (see Figure 12)

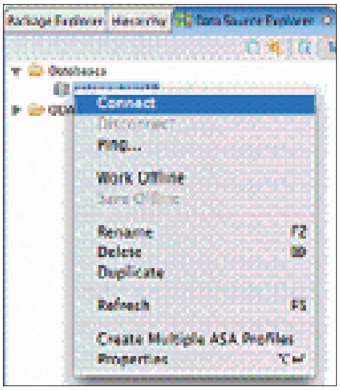


Figure 13

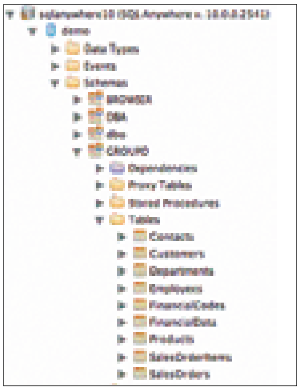


Figure 14

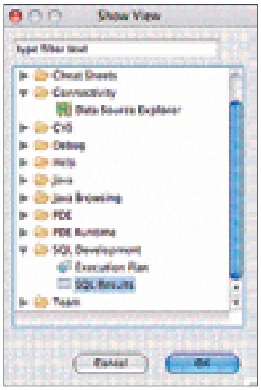


Figure 15

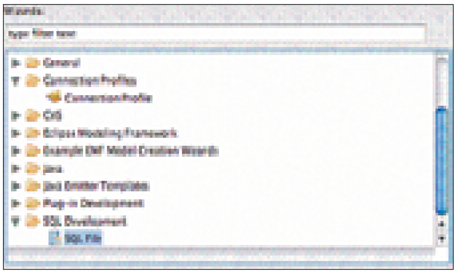


Figure 16

Figure 19

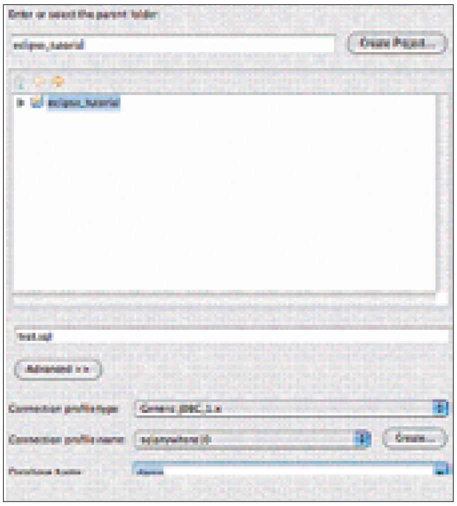


Figure 17

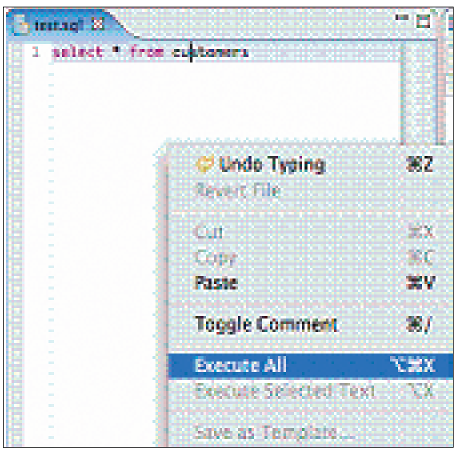


Figure 18

The Data Source Explorer initializes in your Eclipse environment. You can click and drag the title to dock it wherever you like.

If you expand the Databases folder in the view, you should see the connection profile you just created, and any other connection profiles you've created.

4. Right-click the connection profile you just created, click 'Connect' (see Figure 13)

If the connection is successful, you should see results similar to the image below. The tables for the SQL Anywhere sample database are located in the GROUPO schema, so expand 'demo' | Schemas' | 'GROUPO' | 'Tables' and you should be able to view all of your tables. You can also expand any table for a list of columns and column-types (see Figure 14).

The Data Source Explorer offers a great deal of functionality in itself. Right-clicking on any object lets you generate DDL for that object. In the image above, generating DDL for the object 'demo' (top of the tree) will generate DDL for the entire database, whereas generating DDL for the table 'Contacts' will just give you the DDL pertinent to that table. This tool can be used to generate a data definition of any data source you can connect to. Right-clicking on a table or view also gives you a 'Data' sub-menu. From this sub-menu, you can choose to edit the information in that table or view, or sample its contents.

Querying the SQL Anywhere Database

First we must open the SQL Results view. This view is a window view that will be populated with data in table format when a query is executed.

1. Click 'Window' | 'Show View' | 'Other'
2. Choose 'SQL Development' | 'SQL Results' (see Figure 15)
3. Click 'OK'

Next we must open a new SQL File, which we can use to create and execute SQL commands.

4. Click 'File' | 'New' | 'Other'
5. Choose 'SQL Development' | 'SQL File'
6. Click 'Next' (see Figure 16)

We must now specify what project this SQL File will belong to and what connection profile it will be executed against.

7. Fill out the form as show below; be sure to choose the right connection profile and database name. Create a new general project if necessary (see Figure 17).
8. Click 'Finish'
9. Enter any SQL statement into your new SQL File
   - For example - "SELECT * FROM Customers"
10. Save this file, right-click anywhere in the SQL File window, and choose 'Execute All' (see Figure 18)

If your SQL statement executes successfully, you should see the results of the statement in the Results view that we opened earlier. If the statement was a query, the Results view will display the returned records. If it's not a query, you'll be notified of the statement's success and how many rows were affected. Query results should look like the image below (see Figure 19).

This wraps up the tutorial. You may now close Eclipse and shut down the demo database we started earlier.

Summary

In this tutorial, we learned how to:

- Define a driver template
- Create a connection profile
- Connect to a SQL Anywhere 10 database
- Run SQL statements against a SQL Anywhere 10 database from within the Eclipse IDE

The activities we walked through in this tutorial are just an introduction to some of the features available in the DTP framework and tools. We hope it will give you enough familiarity to continue exploring on your own. Since DTP is an open source project, it depends on involvement from the community to a large extent. Please feel free to contribute bug reports, feature requests, and general comments to the DTP project team. Their project home page is located at http://www.eclipse.org/datatools.