

# Asking the Hard Questions about Open Source Software



## Why does IBM consider open source software (OSS) important?

- OSS can be a major source of innovation
  - Innovation can happen any time, anywhere
  - Development through “open communities” leads to potentially broad ideas and creativity
- OSS is a good approach for developing emerging standards
  - Popular open source projects can become de facto / open standards
  - Wide distribution deployment
- OSS is a source of competition in marketplace
  - Office productivity applications (word processing, spreadsheets, presentation)
  - Operating systems (Linux for servers, desktops)
  - In some areas, perhaps only growing competitor to a single established vendor

# What is some of IBM's history with open source software

## 1999 - 2001

- IBM forms Linux Technology Center
- Leads Apache projects Xerces (XML4J), Xalan, SOAP
- Creates OSI-approved IBM Public License
- Strategic participation in Mozilla
- IBM becomes founding member of OSDL
- Founder of Eclipse.org and Eclipse Consortium
- Creates internal bazaar using OSS methodology

## 2002 - 2003

- Linux contributions to scalability (8-way+), reliability (stress testing, defect mgmt. doc)
- Leads Apache projects: Web Services (WSIF and WSIL), Pluto (Portlet API) and WSRP4J (Remote Portal)
- Leads Eclipse projects GEF (editing), EMF (modeling), XSD (XML Schema), Hyades (testing), Visual Editor, AspectJ, Equinox rich client
- Globus Toolkit contributions for OGSA, OGSF

## 2004 - 2006

- IBM and Novell/SUSE achieve security milestone (EAL4+ and COE compliance)
- Eclipse Foundation, Inc. becomes independent - IBM contributes UML2, Voice Tools, Aperi, COSMOS, Ajax Toolkit Framework
- Globus Toolkit 4 is WS-I compliant
- Pledged 500 patents to open source
- Partner with Zend PHP
- IBM enhances Apache partnership
  - Contributes Derby database
  - Contributes voice recognition
  - Supports Geronimo J2EE project
  - Acquires Gluecode for skills
- IBM contributes accessibility code to Firefox

## 2007 - 2009

- IBM leads Open AJAX initiative and announces partnership with the Dojo Foundation
- Leads Apache Tuscany project
- Contributes code for security management to Eclipse Higgins
- IBM contributes to Apache Lucene project and announces OmniFind Yahoo! Edition
- IBM joins OpenOffice.org
- IBM joins Open Health Tools, moving code for medical record management from Eclipse OHF
- Contributes Open Services for ALM – Jazz REST Services samples
- IBM creates ODF Toolkit Union
- Contributes Lotus Notes/Domino app templates to OpenNTF.org
- Incubates Apache Wink (JAX-RS)

**More than 1000 IBM developers involved in OSS projects**

**IBM leads 80+ OSS projects**

**IBM contributes to 150+ OSS projects**

## What does it mean to say that IBM supports open source?

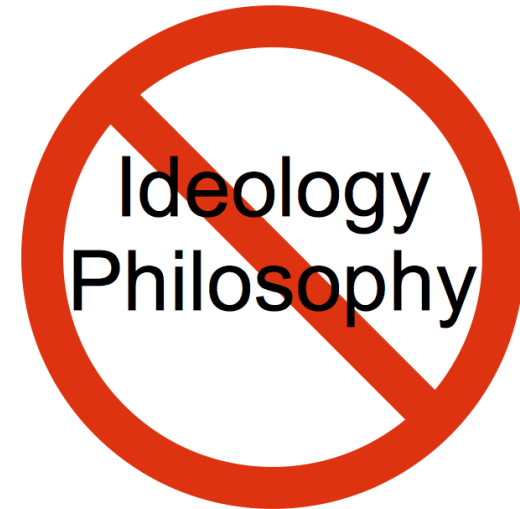
- IBM has more than 800 developers working full time on open source projects, especially Linux, Eclipse, and Apache projects.
- IBM has invested billions of US \$\$ on open source.
- IBM participates in and supports pro-open source organizations like the Linux Foundation, the Software Freedom Law Center, and the Open Invention Network.
- IBM collaborates with many other companies and people around the world on open source projects.
- However ...
  - IBM does not “bless” every open source project in the world.
  - Some open source projects compete with our products and we compete against them.

## How does IBM use open source?

- To run our business
- In our hardware
- In our software
- As part of service engagements
- As an R&D collaboration vehicle
- As a way of influencing the direction of the IT industry
- As a way of leveling the playing field
- As a way of invigorating stagnant market categories

## Is software good software, just because it is open source?

- It depends of your definition of “good,” but by most definitions, the answer is “no.”
- As of August, 2010, the popular code repository SourceForge had over 240,000 registered open source projects.
- Statistically, you might imagine that some are better than others.
- Some are very, *very* good, but not all.
- Your definition of “good” is critical.



## Is the code well architected and implemented?

- Great code may start with the germ of a fantastic idea, but it eventually gets rewritten one or more times to be faster, more reliable, more secure, and more extendable.
- If you are not an expert yourself, seek independent assessments of the quality of the code.
- The quality of the documentation and user interface are important considerations in their own rights, but may also give you an idea of how well designed the core of the software is.

---

## Who are the founders, contributors, and users?

- *People* write code and drive software projects and products.
- Unreliable people may place the future of the software in jeopardy, and thus also your investment.
- Work out “what if” scenarios for what you will do if the code gets abandoned, forked, or acquired.
- Learn what other users have done with the code and about the quality of their experiences with the software and those who created it.

## What is the form and governance of the community?

- Find out if the open source code you are considering is being developed by a healthy, democratic, and meritocratic community or if it is really just a controlling company “coding in public.”
- Learn if the community also includes documenters, graphic designers, and evangelists in addition to coders.
- Look at the project forums, Facebook, Twitter, and other social networking tools to get a sense of the health of the community.
- Don't ignore warning signs of trouble in the community and things that may make you uneasy about it.

## Are there intellectual property issues involving copyrights or code provenance?

- Ignoring legal issues with software can be one of your most expensive mistakes and can literally put you out of business.
- Learn about open source licenses and consider hiring an intellectual property attorney as a consultant when you are considering use of software or negotiating a contract.
- Don't mix open source licenses unless it is legal.
- Make sure the developers of the software you want to adopt played by the legal rules.
- Don't pretend to be an attorney if you are not.

## Does the license suit all your future plans for the code?

- Some open source licenses can be combined and others cannot.
- Some open source licenses allow for free use in commercial, “closed source” applications and others do not.
- Some open source licenses specify some restrictions when you host software-as-a-service.
- Be especially careful if you want to use open source code libraries.
- Understand if the software you plan to use can be hosted on either a private or public cloud.

## Do you have proper legal controls and business processes in place to deal with open source software?

- That is, what is your open source governance strategy?
- Five years ago, it was not uncommon for that strategy to be defined as “you shall use no open source software.”
- You need to understand the legal risks and responsibilities for any software you use, and weigh those against the business value.
- Work out a plan that specifies what business and legal controls are in place to approve use of open source in your organization or in your products, and make sure you have a well defined escalation path.

## Is the software enterprise-ready?

- There's been a lot of discussion about whether open source software is more secure than proprietary software.
- Which open source software and which proprietary software?
- In addition to security, you need to look at reliability, availability, scalability, interoperability, and performance.
- Make sure the software is available on the right hardware platform so you can optimize the environment for your workload.

## Who will maintain your installation of the software?

- If you are planning for your IT staff to install and maintain your software, make sure it doesn't get orphaned when you have personnel turnover.
- When software updates come along, you will need a plan to decide which ones to install and when, especially if major releases come along every six months or so.
- If you customize open source code for your organization, are you prepared to propagate those changes into newer versions of the code?

## How easy is it to integrate the software with your data or other software you already use?

- Does your software use recognized industry standards or does it have its own way of formatting data?
- Are the developers of the software involved in creating the standards that will allow interoperability?
- If you adopt the software, who will do the integration tasks?
- Is the software certified for use on the operating system and hardware platform you plan to use?

## Are benchmarks available to allow performance evaluations of the software with comparable products/projects?

- While benchmarks can be abused, they can be important in learning if particular software is really usable in your business.
- You might worry less about published benchmarks and more about proofs of technology and head-to-head comparisons among the software choices you are considering.
- Consider your software provider's response to such requests for “bake offs” when making your adoption decision.

## What internal education does IBM do for open source?

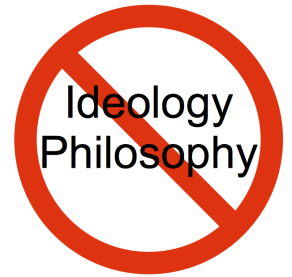
- Enablement material for sales professionals
- Key topics in IBM's general open source education
  - Introduction to open source software
  - Open source software strategic goals
  - Common open source software risks
  - Participating in open source software – IBM time, personal time
  - Third party open source software – access, share, distribute rules
  - Specific license agreements
  - Open source software review and approval process
- Focused education sessions for open source software teams
- Regular refresh and recertification

---

## Can you summarize the areas I need to consider?

- Quality and performance of the code
- Community history, health, and governance
- Legal issues
- How open source and proprietary software will work together
- Standards
- Code maintenance, deployment, and service management
- Your governance of creation and use of open source

## Conclusion



- First and foremost, open source software is *software*.
- When it comes to business and especially enterprise use, open source software should get no immediate free pass because it happens to be open source.
- Conversely, proprietary software should also be measured on a level playing field with open source, and get no special initial treatment.
- All those things that you worried about when choosing proprietary software—security, performance, reliability, availability, interoperability, support, maintenance—are also areas to investigate when considering open source software.

# Legal

## Trademarks and Disclaimers

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries. For a complete list of IBM Trademarks, see [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml):

IBM, the IBM logo, BladeCenter, Calibrated Vectored Cooling, ClusterProven, Cool Blue, POWER, PowerExecutive, Predictive Failure Analysis, ServerProven, Power Systems, System Storage, System x, System z, WebSphere, DB2 and Tivoli are trademarks of IBM Corporation in the United States and/or other countries. For a list of additional IBM trademarks, please see <http://ibm.com/legal/copytrade.shtml>.

The following are trademarks or registered trademarks of other companies:

Java and all Java based trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries or both Microsoft, Windows, Windows NT and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both. Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. UNIX is a registered trademark of The Open Group in the United States and other countries or both. Linux is a trademark of Linus Torvalds in the United States, other countries, or both. Cell Broadband Engine is a trademark of Sony Computer Entertainment Inc. InfiniBand is a trademark of the InfiniBand Trade Association.

Other company, product, or service names may be trademarks or service marks of others.

## Notes

Linux penguin image courtesy of Larry Ewing ([lewing@isc.tamu.edu](mailto:lewing@isc.tamu.edu)) and [The GIMP](#)

Any performance data contained in this document was determined in a controlled environment. Actual results may vary significantly and are dependent on many factors including system hardware configuration and software design and configuration. Some measurements quoted in this document may have been made on development-level systems. There is no guarantee these measurements will be the same on generally-available systems. Users of this document should verify the applicable data for their specific environment.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

Information is provided "AS IS" without warranty of any kind.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices are suggested US list prices and are subject to change without notice. Starting price may not include a hard drive, operating system or other features. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Any proposed use of claims in this presentation outside of the United States must be reviewed by local IBM country counsel prior to such use.

The information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any