

The Wisdom Report

A PUBLICATION OF  WisdomGroup

Fourth Quarter 2008

WisdomGroup.com

What is Open Source?

When Performance Matters

Who creates the most heavily used web server software? Who has a bigger share of the web server market than anybody else? Hint: It's not Microsoft.

More than half the web servers on the Internet use a powerful blend of open source software: a web server called Apache and an operating system called Linux. One market research firm reports that 75% of *new* web servers launched in September 2008 run Apache on Linux. Google, Yahoo, and Amazon all run Apache and Linux. They've bet their companies on this decision.

Given that several successful companies have bet their livelihood on open source, you might ask these questions:

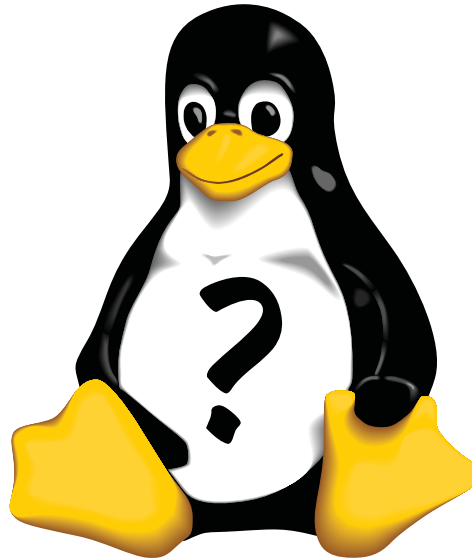
- What is open source?
- Why should I care?
- How can my company benefit?

Source Code, Executable Code

Let us first define the word "source," which in this case refers to source code. Source code is a human-modifiable form of software. It's software as created by programmers. Source code must be translated into executable code (through a process called compiling) before a computer can run it.

Water is a useful metaphor for this discussion. Water can exist in three different states (steam, liquid, and ice) without changing its chemical properties. It's still water. Likewise, software can exist as source code or executable code – it's still software. We use water in one of its three states depending on whether we want to wash a car (liquid), cool a drink (ice), or clean a carpet (steam).

In the world of software, executable code is difficult for humans to read, but easily digested by computers. Source



code is easily read & modified by humans, but indigestible by computers.

One Way Translation

Translating source code into executable code is a one-way street. Once a programmer has compiled source code into executable code, it is virtually impossible to convert it back into source code. You might think of a compiler as a blender that chops the food into a form easily digestible by the computer. After it's chopped, the food never returns to its original form.

That's why programmers always keep a copy of the source code handy. You might need to make changes (fix bugs) and re-compile.

Trade secrets that are hidden in executable code can be easily read in source code. So developers go to great lengths to protect their source code, except in the open source community.

Closed Source Tradition

Most of today's business software is closed source. Two examples: Microsoft Office and Apple iWork. Both Microsoft and Apple consider the source code for their respective office suites to be highly confidential. You will never (legally) download the source code for a closed source product. Security is so tight at Microsoft, for example, that engineers are only allowed to view that portion of the source code that is relevant to their current project.

Contrast that against open source software. Open source products make the source code freely available for viewing, comments, and modification by the software community at large.

Open Source Chaos?

But doesn't such openness lead to chaos? Intuitively, you would think so, but it

In this field...	the equivalent of source code is...	and the equivalent of executable code is...
Music	Sheet music with notes written on paper.	A music CD playable on your stereo system.
Architecture	Blueprints and details structural diagrams.	The finished building.
Pastry Arts	A recipe.	A cake.
Software	Source code.	Executable code, made available for CD-ROM or downloaded from the web.

doesn't. The culture of the open source community has its own rules of conduct that most members willingly follow. Rules are enforced by the community in ways that developers respond to.

It is difficult to imagine an apparently chaotic system that produces robust software. How counter-intuitive! But a long list of successful open source projects (available through your favorite search engine) shows that the method works. Programmer & anthropologist Eric S. Raymond explores open source cultural norms in his book, *The Cathedral and the Bazaar*.

The Banking Metaphor

Some makers of security software believe that hiding their source code will prevent malcontents from finding & exploiting vulnerabilities. Their reasoning is like that of a bank owner who keeps the building blueprints secret so thieves don't know how to drill the safe from next door.

Other security software companies take a different view. They want many people (including malcontents) to view the code so that vulnerabilities can be discovered and patched early. Their philosophy could be summed up in a statement by Linus Torvalds, the creator of Linux: "Given enough eyeballs, all bugs are shallow." In other words, when you have thousands of developers working on the same project they are likely to be widely dispersed across the code. Wide coverage means that a given bug is more likely to be discovered by at least one developer.

In our banking metaphor, the open source company would be like a bank owner who wants many people to review the blueprints, find the weak spots, and fix security holes before the bank gets robbed.

It's Not All-or-Nothing

Choosing an open or closed source strategy is not an all-or-nothing decision. Google, the busiest search engine on the web, has bet the company on open source software. And yet their most strategic software, the secret mathematical engine that drives their search results, remains closed. So how does a company decide what to open and what to close?

Open source works when the upside of sharing outweighs the downside of exposed code. In the Google example, using & contributing to Linux gives Google a very stable operating platform at a relatively low cost. In effect they are leveraging the time and talent of thousands of developers around the world, while those developers are also leveraging Google's expertise.

It doesn't matter that Google's competitors also benefit from that same stable Linux platform because Google differentiates itself through advanced search techniques. By leveraging the community for assistance with the platform (Linux, Apache, and other tools), Google frees up engineering time & talent to focus on their secret sauce (search methodology & mathematics).

If Your Business Isn't Software

So...What if you don't run a software company? How can you profit from open source software?

There's always a chance that open source won't work for your company. Maybe you run an enterprise that doesn't benefit from the web. In that case, your current software options may suit you just fine.

Of course, you may have already benefited from open source. There is at least a 50% chance that your company's web site runs on Linux. If that's the case, then rest assured that you're running on a very stable and robust platform.

Think Big, Start Small for ROI

If you really want to see ROI from open source, encourage your technical team (software developers, network support, etc.) to explore open source tools. Chances are that they've already experimented with open source and they're waiting for an opportunity to do an official company project.

Start small, perhaps with an automated network monitoring project or a small database. As you explore the technology, new ideas will become apparent to you. View this as a way to accelerate & strengthen the software development process.

Open Source and Automobiles

BMW is arguably the best automotive engineering company on the planet. Does

BMW invent the wheel every time they design a new automobile? Absolutely not. BMW leverages thousands of years of "open source" wheel technology, and they focus their design energy in areas where they can make a difference: engine, transmission, suspension, and luxury amenities. A similar model can be applied to software development. In fact, it already has.

Suggested Reading

The Cathedral and the Bazaar, by Eric S. Raymond. The author is both participant and anthropologist in the open source community.

About the Author

Raymond T. Hightower is president of WisdomGroup, creators of outstanding web-based software. WisdomGroup leads the open source user group ChicagoRuby.org, and we created the WindyCityRails conference. For more information, visit WisdomGroup.com.

Glossary

Linux

A free Unix-type operating system that runs on inexpensive personal computers. Created by Linus Torvalds and improved by thousands of developers around the world.

Unix

A robust, enterprise-class (and expensive) operating system typically used for large, compute-intensive operations. Created by AT&T with improvements by developers around the world.

Free Software

Software that can be used, studied, and modified without restrictions. Conceived by Richard Stallman in 1983. When the word "free" is used in the context of open source, it does not mean that you don't have to pay for it. In clarifying this point, Stallman says "Think free as in speech, not free as in beer."