

“The Trend Towards the Use of Open Source Software in E-Commerce: Technical and Business Implications”

Our last two articles reviewed the threats posed to E-Commerce Servers, and solutions to those threats. One has to keep in mind that threats are usually designed to take full advantage of a security flaw or vulnerability in the Operating System, Database, or the Web Server. Many E-Commerce websites have traditionally used Windows as the main Operating System to host the website, and either Access or SQL Server as the “backend”, which is the database that contains the vital company data and information on customers. However, in the past we have seen such systems prone to serious threats and large scale attacks.

Are there other alternatives that are more secure? Yes there are, and it is called “Open Source:” Software. There has been a move in the business community towards the use of this “Open Source” Software. Specifically, the “Open Source” Software that is predominantly used today by business for E-Commerce is Linux vs. Windows as the Operating System; Apache vs. Internet Information Services as the Web Server, and PostgreSQL or MySQL vs. Access and SQL Server as the database.

This article is divided into the following sections:

- 1) A Formal Definition of Open Source Software;
- 2) A History into the Creation and Development of Linux, Apache, and the other SQL databases (focusing on PostgreSQL and MySQL);
- 3) The Advantages and Disadvantages of Open Source Software;
- 4) The Trends Towards Using Open Source Software.

A Formal Definition of Open Source Software

As stated previously, many servers and computers today use some form of Windows as the Operating System. This software can be referred to as “Closed Source”, because although you can use the services and software applications that are available with the Operating System, you cannot see the programming code that makes up the Operating System. As a result, you cannot modify the environment of the Operating System in order to fit your needs and desires. In other words, with Closed Source Software, “What You See Is What You Get”, and you as the E-Commerce business owner, have no choice but to accept what is given to you. A formal definition of Closed Source Software is: “. . . the customer will only get a binary version of the computer program they licensed and no copy of the program's source code, rendering modifications to the software practically impossible from the technical side, because the usual way to modify a program is to edit its source code and then compile it.” (Source: 1).

However, “Open Source” Software is just the opposite of Closed Source software. With the Open Source software, usually, the software is free to download and use. But what is more important is that you can choose what services and applications you want to install and use, and the bottom line is that with Open Source software, you have access to the Operating System code, and as a result, you the E-Commerce business owner, can modify the environment of the

Operating System in order to fit the needs and desires of your business. A formal definition of Open Source software is: “. . . software for which the underlying programming code is available to the users so that they may read it, make changes to it, and build new versions of the software incorporating their changes.” (Source: 2).

There are advantages and disadvantages of using Open Source Software over Closed Source Software, especially with regards to licensing and redistribution rights, this will be discussed in a later section.

A History Into the Creation of Linux, Apache, and the other SQL databases (PostgreSQL and MySQL)

The history of the development of Linux, Apache, and the various SQL databases is very rich. Since these tools are Open Source Software applications, there have been many people from around the world who contributed to the development of these applications. These people have ranged from experienced software developers to end users who want to implement a brand new Web Hosting System. This section will detail the history of the development of Linux, Apache, and the SQL databases (PostgreSQL and MySQL).

A History Into the Creation and Development of Linux

Linux has its origins from the UNIX Operating System, also simply known as UNIX. UNIX was developed back in 1969, at Bell Laboratories. The group at Bell Laboratories had three primary objectives when developing UNIX:

- 1) “To keep the Operating System elegant and simple”;
- 2) “It should be written in a high level language rather than assembly language”;
- 3) “Allow, or permit re-use of the code.”

SOURCE: 3

In order to adhere to the above goals, a number of principles, or philosophies have been incorporated into UNIX, which as a result, has also made Linux popular. These principles, or philosophies are:

- 1) “Rule of Modularity: Develop software code out of simple components with clean interfaces”;
- 2) “Rule of Composition: Allow programs to be connected to other programs”;
- 3) “Rule of Clarity: Write the software code in such a manner that software developers will be able to read and maintain the source code well into the future”;
- 4) “Rule of Simplicity: Develop the software code that “. . . resists bloat and complexity. . .”, in other words, make the software code simple”;
- 5) “Rule of Transparency: Develop the software system so that any software developer can “. . . look at it and immediately understand what it is doing and how”;
- 6) “Rule of Least Surprise: Design software programs which require the least new learning from the end user”;

- 7) “Rule of Repair: Develop software code that will work with, or cooperate with other software code, or if not, it will not work together at all”;
- 8) “Rule of Economy: Develop software code that can be changed and modified with the constantly changing world of software development”;
- 9) “Rule of Generation: Keep software code specifications simple”;
- 10) “Rule of Representation: Transfer complexity from the software code to the data itself”;
- 11) “Rule of Separation: When developing the software code for the application, separate the Front End processes (for example, the Graphical User Interfaces) from the Back End processes (for example, the database). The Front End and the Back End processes should be connected together (or communicate together), “. . .via a specialized application protocol. . .”;
- 12) “Rule of Optimization: Always first prototype any developed software code”;
- 13) “Rule of Diversity: Do not design and develop, rigid, closed software code”;
- 14) “Rule of Extensibility: Develop your software code to be extensible, meaning, allow your software code to have the functionality to be “. . .able to plug new functions into the architecture without having to scrap and rebuild the architecture . . .”.

SOURCE: 4

The above stated objectives as well as the principles, or philosophies, incorporated into UNIX have allowed it to stand the test of time, and give it certain advantages over Closed Source Software, which will be explored in detail in the next major section.

However, at the time it was developed, UNIX was closely guarded and not open to the public. In order to insure its dominance, the various UNIX vendors priced it high enough so that everyday computer users would not be able to use it, but only institutions and business organizations that could afford the big price tag could use it.

An alternative was developed, and it was called MINIX. It was developed by Andrew S. Tanenbaum, a Professor of Computer Science at the University of Amsterdam. The goal of developing MINIX was to show how an Operating System worked, and was primarily designed to work on the Intel 8086 microprocessors. MINIX had 12,000 lines of program code. Although this was not the most robust Operating System, it had the distinct advantage in that it was Open Source, and it was available to anybody who wanted to further enhance the software code. In fact, the code was made available in a book called “Operating System” which was written by Tanenbaum himself. This book gripped the minds of computer science professionals all over the world. One person that was captivated by this book was Linus Torvalds, the scientist who created Linux.

At the same time MINIX was developed, there was another movement in the computer science world, and this was the “GNU Project”, initiated by Richard Stallman. The major goal of the GNU project was to create quality software that was free to distribute, and not be restricted by software licensing regulations. Richard Stallman’s vision for the GNU project was “. . .that unlike other products, software should be free from restrictions against copying or modification in order to make better and efficient computer programs” (SOURCE: 5). In March 1985,

Richard Stallman published the “GNU Manifesto” in *Dr. Dobbs' Journal of Software Tools* **10** (3). This is considered to have been the launch platform for the GNU Project. Under the GNU Project, Richard Stallman developed many software applications such as “emacs” (stands for Editor MACro\$), which is essentially a text editor used by software developers to develop and compile software code. Another famous application developed by Richard Stallman under the GNU Project is the GNU C Compiler, or GCC for short. A compiler is “. . . a software-development tool that translates high-level language programs into the machine-language instructions that a particular processor can understand and execute” (SOURCE: 6).

However, despite all of these developments of applications under the GNU Project, there was still no Operating System, and the Operating Systems that were under development in the GNU Project were still years away from completion. As a result, the ambitious Linus Torvalds was eager to create and implement an Operating System. The following e-mail sent by Linus Torvalds to the MINIX news group can be considered to be the starting point for Linux:

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)

Newsgroups: comp.os.minix

Subject: What would you like to see most in minix?

Summary: small poll for my new operating system

Message-ID: <1991Aug25.205708.9541@klaava.Helsinki.FI>

Date: 25 Aug 91 20:57:08 GMT

Organization: University of Helsinki

Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things). I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Linus (torvalds@kruuna.helsinki.fi)

PS. Yes - it's free of any minix code, and it has a multi-threaded fs.

It is NOT protable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-).

SOURCE: 7

The first version of Linux (Version 0.01) came out in September 1991. The second version (Version 0.02) then came out on October 5th, and by December, Version 0.10 became available. But despite the progression of these versions, Linux was an Operating System still in a “bare bones” format. But, work and research continued, and in time, literally hundreds of thousands of people joined in on the development of Linux, and eventually, Linux became licensed under the

“GNU General Public License”, which guaranteed that the Linux source code would be made available to anybody interested in copying, studying, modifying, and enhancing it.

Linux still remained a “Command Line” Operating System, and since it was Open Source Software with no licensing restrictions, other vendors moved in quickly in order to create versions of Linux which provided for a Graphical User Interface, or GUI, in a manner very similar to the Windows Operating System. Some of these vendors include Red Hat (www.redhat.com) and Debian (www.debian.org).

A History Into the Creation and Development of Apache

Apache is the Open Source Web Server application, which is used to host and serve websites. The development of Apache closely parallels that of Linux. The idea was generated by a few people, and once its advantages were discovered, many other software and web developers joined in to further refine and enhance Apache, and to make it the market leader. In fact, when compared to other Web Servers, especially Microsoft’s Internet Information Services (IIS) Web Server, Apache has 69.32% of the market share, versus Internet Information Services market share of only 20.45%, in April 2005. Since then, the lead has increased slightly. This will be discussed later in the article. (SOURCE: 8).

Apache has its origins all the way back to February of 1995. Rob McCool of the National Center for Supercomputing Applications, University of Illinois, Urbana-Champaign, developed the original software code for a protocol called HTTPd, which stands for Hyper Text Transfer Protocol Daemon, which is essentially a Web Server, but is no longer in use. HTTPd was designed to be Open Source, in that it was made free to the public, and if any bugs were found or any enhancements were to be made, the public would notify Rob McCool, and he would then incorporate the fixes and enhancements in later releases and versions of HTTPd.

However, Rob McCool subsequently left the project, and although there were many people using HTTPd, and making patches and upgrades to the software code of HTTPd, there was no central authority to coordinate, monitor, and make record of these patches and upgrades. Later in 1995, Brian Behlendorf and Cliff Skolnick created a mailing list to discuss and review these patches and upgrades that were being made to HTTPd. Eventually, six other contributors (Roy T. Fielding, Rob Hartill, David Robinson, Cliff Skolnick, Randy Terbush, Robert S. Thau, and Andrew Wilson) joined in, which thus formed the Apache Group. This group had no formal organizational structure, never physically met, and maintained communications only via e-mail.

The Apache Group utilized HTTPd Version 1.3 in which to implement all of the patches and upgrades. The first, official Open Source release of Apache (Version 0.6.2) was made available to the public on April, 1995. Despite the success of this first version, the core code still needed a major revamp and redesign, and in August of the same year, Apache (Version 0.8.8) was released, which included the revamps and redesigns. Eventually, Apache (Version 1.0) came out in December, 1995. Today, the most recent Apache is (Version 2.0). Finally, in 1999, the Apache Group formed the Apache Software Association (www.apache.org) “. . . to provide organizational, legal, and financial support for the Apache HTTP Server. The foundation has placed the software on a solid footing for future development” (SOURCE: 9).

The primary advantage over other Web Servers (especially IIS) is that the core of Apache is considered to be very “light-weight”. Any other functions that are needed can be simply installed into the core. This means that the end user, such as a business owner, can implement the functions and applications they need to implement, without having to burden the Web Server or Processor resources with extra functions and applications that are not needed: “. . . you can keep the size of the executable down by leaving out functionality that you don't need. It also means that if there is some functionality missing that you do need, you can write your own custom module to plug into the core” (SOURCE: 10).

A Review of PostgreSQL and MySQL

When one thinks of a database, the image of a spreadsheet or a table containing data (such as a customer database) comes into mind. However, there must be a way to access, manipulate, and analyze the data that is in the table. This is where Structured Query Language, or SQL, comes into play. A very simple definition of SQL is: “. . . is a standard interactive and programming language for getting information from and updating a database” (SOURCE: 11). SQL was developed by IBM in the 1970's and has become the de facto standard for databases.

Although the intent of SQL was to be an open, standard language without any claims to corporate ownership, different software vendors have taken SQL and modified it or enhanced it in order for it to fit their existing software product lines. Two examples of this are Oracle and Microsoft. Oracle has created a database programming language called PL/SQL, and Microsoft created its own database programming language called SQL Server. While both of these languages contain the core SQL programming statements, these languages also contain other features which make it compatible and work with other Oracle and Microsoft software products. However, PL/SQL and SQL Server are Closed Source Software, therefore an end user does have to pay for them, often with license restrictions in place as well.

But, there are two Open Source Databases, which are called PostgreSQL, and MySQL. They both also use the core SQL programming statements, but since they are Open Source, they are made freely available to the public, and there are no licensing restrictions.

PostgreSQL has a rich history, which dates all the way back to 1986. The basic concepts for PostgreSQL appeared in a paper called “The Design of POSTGRES”, and the initial data model appeared in another paper entitled “The POSTGRES Data Model”. A “demoware” version of PostgreSQL was made available in 1987, and Version 1 of PostgreSQL was released to users in June 1989. Versions 2 and 3 of PostgreSQL were subsequently released in 1990 and 1991, respectively. Initially, PostgreSQL was simply known as “POSTGRES”. With later versions coming out, it eventually became known as “Postgres95”.

Then in 1994, Jolly Chen and Andrew Yu added the SQL component to Postgres95. It was then released over the Internet as an Open Source Database. However, by 1996, Postgres95 was renamed to PostgreSQL in order to “. . . to reflect the relationship between the original POSTGRES and the more recent versions with [the] SQL capability” (SOURCE: 12). Today, PostgreSQL Version 8.0 is available, and can be downloaded at www.postgresql.org.

The development of MySQL dates all the way back to 1979. Back then, Michael Widenius was a software developer for a company in Sweden called TcX. While at this company, he developed an in house database called UNIREG. By 1994, TcX ventured into developing Web based applications, and used UNIREG as the primary database. However, UNIREG did not interact with dynamic Web pages, so as a result, TcX began to look for alternatives. One alternative being given serious consideration was SQL and mSQL. However, mSQL was still in its early stages of development, in 1.x Versions. So, the compatibility of mSQL with UNIREG was in question, and as a result, TcX decided to focus on building a database that would meet its own internal development needs.

Rather than building a new database from the ground up, TcX decided to enhance upon on UNIREG by “. . . writing an API into its system that was, at least initially, practically identical to the mSQL API. As a result, an mSQL user who wanted to move to the TcX more feature-rich database server would only have to make trivial changes to any existing code. The code supporting this new database, however, was completely original” (SOURCE: 13). Finally, by May 1995, TcX had the database it was looking for, which was named MySQL 1.0. Eventually, MySQL was released over the Internet. Today, MySQL Version 4.1 is available, and can be downloaded at www.mysql.com.

The Advantages and Disadvantages of Open Source Software

As can be seen from the history of Linux, Apache, PostgreSQL, and MySQL, the concept of Open Source Software has been around for a long time. However, the public is still unfamiliar with Open Source Software, because of the dominance of the major vendors. When people do hear about Open Source Software, the initial reaction is often strong hesitation, because people are used to only the vendor dominant software packages, such as those from Microsoft, Oracle, etc.

Probably the biggest advantage that people hear of about Open Source Software is that it is free, and that you do not have to pay thousands of dollars in licensing fees. While this is true, Open Source Software also has other advantages, as well as its disadvantages. This section will outline them.

The Advantages of Open Source Software

- 1) With Open Source Software, the actual software code is made available to the public for free, and the end user has the right to modify it or enhance it the way they see fit, or as business needs dictate.

Business today, especially in E-Commerce, is changing constantly, especially the technologies that are involved. If you use Open Source Software, for your E-Commerce website, you can easily transfer the code to new hardware (for example, migrating to a newer and faster server); you can modify or enhance the code to your E-Commerce website as your customer base is dynamically changing, and as you further make modifications or enhancements to the code, it is much easier to track and isolate bugs, and fix them on the spot. This also helps to prevent from having to develop software

from the ground up each time, and having to constantly “reinvent the wheel”. With Open Source Software, if there is an application which you have found and would like to modify it your own business, all you have to do is freely acquire the core code, and make the necessary modifications and enhancements in order to meet the needs of your E-Commerce business.

- 2) You can redistribute modifications and improvements made to the software code.

With redistribution rights, you have immediate access to the newest ideas and the latest developments from large software communities and software developers, working in parallel with each other for a common cause, and not against each other, as you would see in a Closed Source Software environment. This is a very important advantage for an E-Commerce business, especially in terms of staying ahead of the competition.

- 3) There is no single authority which can dictate in a unilateral fashion how the software can be used.

This is especially true when newer software versions become available. As an E-Commerce business owner, if you use software from a commercial, Closed Source vendor, you will have to buy an entirely new product if your existing software system becomes obsolete, and is no longer supported. However, with Open Source Software, as newer software is developed, you can still keep your existing software system, and simply apply the upgrades as they become available. There is no forced upgrade path with Open Source Software. As an E-Commerce business owner, this results in true cost savings. Also, as your E-commerce business grows (for example, you add more E-Commerce servers), you can install the same software on different servers without concern of licensing fees and violation of licensing agreements. In other words, the costs of scalability are almost zero. As a result, this will also insure software platform consistency for your E-Commerce business.

- 4) There are no “blackboxes” with Open Source Software.

With Closed Source Software, the underlying theme is essentially “What You See Is What You Get”. You are forced to use with what the vendor provides to you. For an E-Commerce business owner, you could be spending a lot of money for software that you really do not need the functionality of, or very often, which does not give you all of the functionality that you need. With Open Source Software, especially with Linux and Apache, since the code is transparent and you can access it, you can modify/enhance the software code to exactly fit the needs of your E-Commerce business, and you can also install only those applications that you need, as well as add more applications as your business needs dictate. Also, Open Source Software allows you, the E-Commerce business owner to stick the mantra: “Stay close to your customer”. For example, you can modify and enhance the code in order to meet the specific needs of your customers. As a result, you can also broaden your product line more, and attract and build a wider and broader customer base.

5) The levels of quality and security are better in Open Source Software.

Since there is a transparent process in Open Source Software, there are many more software developers examining and testing the code, thus there tends to be higher standards of quality within the software code. With Closed Source Software, there are not as many software developers examining the code, and their standards of quality could often be misguided by having to meet corporate objectives and unrealistic deadlines. In terms of security with Open Source Software, there tends to be immediate notification and awareness of any threats and vulnerabilities, since there is literally a worldwide community monitoring the software code. However, with Closed Source Software, threats and risks only become known to the public after they occur, and cause widespread damage, even though they may have been internally known ahead of time to the vendor. Also, there is a much further delay in releasing software patches and fixes with Closed Source Software than with Open Source Software. This could have serious consequences if you are an E-Commerce business owner. For example, suppose you utilize Closed Source Software to support your E-Commerce website. If your website is hit with a serious threat or risk, your E-Commerce business could be literally shut down for an indefinite period of time until the vendor comes out with the appropriate software fixes.

6) The cost of acquisition is almost zero.

The only cost that you would have to pay for Open Source Software is the cost of the media, if you order the software on a CD through an Open Source vendor such as Red Hat, rather than downloading it for free on the Internet. This is a very strong advantage for E-Commerce business owners, especially in the start up phase of the business. This means that the seed money acquired can be spent on marketing and getting customers rather than spending the money on expensive, Closed Source Software. In other words, the Total Cost of Ownership is much lower with Open Source Software than it is with Closed Source Software.

7) Open Source Software insures a common, computing environment.

As an E-Commerce business owner, an Open Source Software environment will allow you to share common protocols which will permit for different applications to interface and work with each other, at a very cost effective solution. With Closed Source Software, you will have to purchase expensive software applications with no guarantee that they will interface and work together. With Open Source Software, an E-Commerce business owner can build and leverage upon an existing computing infrastructure, whereas Closed Source Software would force an E-Commerce business owner to build an entirely new computing infrastructure from the ground up.

8) With Open Source Software, there is longevity and low risk of abandonment.

Suppose, as an E-Commerce business owner, you purchased a proprietary software application from a Closed Source Software vendor. But over time, this vendor has had low sales for their software products, and as a result, was forced to cease business

operations. This could have detrimental effects to you. Not only will you have to purchase and acquire another software application from another vendor, but you have also lost your Return On Investment (ROI) on the original software application. However, this is not the case with Open Source Software. Since Open Source Software is not vendor dependent for development and production, the software code will always be available, with many people around the world working on it, depending how many people are devoted to that particular Open Source Software project. In other words, “. . .the larger the adoption, the safer the investment [and there is] no single point of failure” (SOURCE: 14).

9) There tends to be better technical support for Open Source Software.

There is often the tendency to think that if something is free, the follow up support will be poor, and if you purchase an expensive Closed Source Software application, the expectations will be that you will receive high levels of technical support, because “you get what you pay for”. However, with Open Source Software, this is the opposite. As an E-Commerce business owner, how many times have you been put on hold or have had to dial numerous phone options in order to get technical support for your Closed Source Software application? For that matter, how times have you been told that a technical support specialist will call you back, and you never get that call? With Closed Source Software, there also tends to be a per call or incident fee. Because of the breadth and scope of Open Source Software, there are always Online User Group Forums in which you can instantly post a message or send an e-mail about a technical support question, and probably (although not guaranteed) within a short period of time, you will receive a response from an expert, literally, half way around the world, at zero cost.

10) There are no licensing or per seat fees with Open Source Software, and thus innovation and creation can be stimulated.

As stated previously, Open Source Software is free and available to the public. Therefore there are no licensing or per seat fees, you can install the same software on multiple machines. It is the licensing and per seat fees which makes acquiring Closed Source Software cost prohibitive to many organizations and businesses. Since this is not the case with Open Source Software, businesses and organizations can now spend more time on Research and Development on newer products, rather than having to spend a bulk of that time having to be concerned with about the “Bottom Line”. This also helps the progress of technology to move forward.

The Disadvantages of Open Source Software

1) Because of the nature of Open Source Software, there is no guarantee that development will actually occur.

Since the advancement of Open Source Software is dependent upon the level of interest, and if there is only a minimal amount of interest, the Open Source Software project could conceivably just “die out” on its own. However, with Closed Source Software, a

vendor can still keep the software development project alive, if it is deemed that ultimately there is market potential.

2) Open Source Software is more prone to problems with regards to Intellectual Property.

Since Open Source Software is made available to the public, and has great levels of interaction among software developers from all over the world, the lines tend to become unclear as to who owns the Intellectual Property of the software, who can hold patents, etc. For example, a software developer could “. . . consider source code not as an executable device, but a mere description of how a device (the computer) executes, and therefore uphold the idea that source code is not by itself (in absence of an executable program) covered by patent law even in countries where software patents are accepted” (SOURCE: 15).

3) Open Source Software may not be compatible with Closed Source Software, and vice versa.

There are proprietary file formats associated with Closed Source Software which cannot be read by Open Source Software, thus making any migration to Open Source Software from Closed Source Software impossible.

4) The resistance towards migration to Open Source Software from Closed Source Software.

Although using Open Source Software seems appealing because of its many advantages, there will always be the probability of human resistance to actually using it. In other words, the psychological phenomenon of the human resistance to change and try something new and different will always be an obstacle in the adoption of Open Source Software.

5) The fear that Open Source Software will be more prone to attacks and threats than Closed Source Software.

Although Open Source Software has proven to be secure and stable, there is always the human fear that since the code is readily available to the public with Open Source Software, it will be much more prone to attacks and threats. While this is always a possibility, Closed Source Software has proven to be just as vulnerable to attacks and threats, given the recent advent of Internet Worms and Viruses targeted towards Windows Operating Systems.

6) There are business reasons for resistance to utilizing Open Source Software:

“*There is no marketing budget to push the product”.

“*There is no funded product development budget. In practice, this means that Open Source software “products” tend to get ease-of-use features and user-oriented documentation significantly later in their lifecycle than commercial products. For example, early Linux distributions attracted justifiable criticism for the

complexity and lack of standardisation of their installation and management processes”.

“*Open Source developers tend to be very passionate about technical issues. Consequently, without a project leader with good people skills an Open Source project can break up in acrimony”.

“*Senior managers in companies are likely to equate “free” with “unreliable””.

“*There is no commercial organisation who you can sue if something goes wrong”.

“*Because I do not pay the software developers I have no control over them”.

“*Because the developers are motivated by recognition rather than money, they are unpredictable, for example they might all rush off and work on a new, more exciting Open Source project.”

SOURCE: 16

“*Resources can’t be scheduled or held to account”.

“*No direct economic return for effort”.

“*Developers may withhold best work for the commercial (profit-making) sector.”

SOURCE: 17

The Trends Towards Using Open Source Software

The Open Source Software that has achieved the greatest market dominance and market share has been Apache, as stated previously in this article. The exact dominance of Apache can be seen in Table 1 (SOURCE: 18). These statistics are as of May, 2005. Obviously, Linux and the SQL databases (PostgreSQL and MySQL) have a long way to go yet before they reach the dominance level of Windows and SQL Server, but as people realize the benefits of using Open Source Software in E-Commerce, the adoption rate will increase. There has also been a trend towards the adoption of Open Source Web Browsers, one browser that is gaining momentum is called “Firefox”.

VENDOR	WEB SERVER	MARKET SHARE
Apache	Apache	69.37%
Microsoft	Internet Information Services	20.54%
Sun Microsystems	Java System Web Server	2.92%
Zeus	Zeus Web Server	0.89%

TABLE 1

Article Summary and Prelude To The Next Article

In summary this article has reviewed:

- The differences between Open Source Software and Closed Source Software;
- The history of the development of Open Source Software, focusing upon:
 - *Linux (Operating System)
 - *Apache (Web Server)
 - *PostgreSQL and MySQL (SQL Databases)
- The advantages and disadvantages of Open Source Software;
- The trends towards the use of Open Source Software.

While it is up to the E-Commerce business owner to decide what types of software and applications they want to use and from whom, one should consider the distinct advantages Open Source Software possesses, namely its availability, security, customization, and portability.

Our next article, entitled “How To Build an E-Commerce Business Using The Tools Of Open Source Software”, will continue with the Open Source Software theme, and focus on how to build an E-Commerce website using the tools that are available in Linux, Apache, and the SQL databases.

SOURCES:

1) http://en.wikipedia.org/wiki/Closed_source_software

- 2) <http://www.daahp.wayne.edu/glossary.html#o>
- 3) http://vertigo.hsrl.rutgers.edu/ug/unix_history.html
- 4) <http://www.elegantinnovations.net/history.html>
- 5) <http://www.linuxgazette.com/node/9721>
- 6) http://www.caps-entreprise.com/en/glossaire_contenu.html
- 7) <http://www.linuxgazette.com/node/9721>
- 8) http://news.netcraft.com/archives/web_server_survey.html
- 9) http://httpd.apache.org/ABOUT_APACHE.html
- 10) http://apache.rcbowen.com/ApacheServer.html#Introduction_to_the_Apache_Serve
- 11) <http://www.webasyst.net/glossary.htm>
- 12) <http://www.sql.org/sql-database/postgresql/manual/history.html>
- 13) http://www.unix.org.ua/oreilly/linux/sql/ch01_04.htm
- 14)
<http://64.233.179.104/search?q=cache:Cmrjc1oPEPEJ:www.rsmart.com/assets/OpenSourceOpenSLearningJuly2004.pdf+advantages+of+open+source+software&hl=en#3>
- 15) http://eu.conecta.it/paper/Perceived_disadvantages_ope.html
- 16)
<http://64.233.179.104/search?q=cache:3PaXYUq2ELEJ:www.oss.gov.za/docs/ossreportv2.pdf+disadvantages+of+open+source+software&hl=en>
- 17) <http://www.wohl.com/wa0132.htm>
- 18) http://news.netcraft.com/archives/web_server_survey.html