

THE VALUE OF OPEN SOURCE SOFTWARE: THE CREATION OF VALUE LEADING TO  
MARKET TRANSFERENCE

A thesis submitted to the faculty of  
San Francisco State University  
In partial fulfillment of the  
requirements for the  
degree

Master of Business Administration

by

Julie R. Blaustein

San Francisco, California

December 2003

Copyright by  
Julie R. Blaustein  
2003

## CERTIFICATION OF APPROVAL

I certify that I have read THE VALUE OF OPEN SOURCE SOFTWARE: THE CREATION OF VALUE LEADING TO MARKET TRANSFERENCE by Julie Blaustein, and that in my opinion this work meets the criteria for approving a research project submitted in partial fulfillment of requirements for the Master of Business Administration degree at San Francisco State University.

--signed--

---

Sameer Verma  
Assistant Professor of Information Systems

--signed--

---

Richard Castaldi  
Professor of Marketing



## Table of Contents

Introduction.....	1
Definition of Open Source Software.....	1
The Cathedral and the Bazaar.....	2
Licensing Issues.....	2
Multiplicity of Licenses.....	2
The Dual Purpose of Licenses.....	4
Open Source Software.....	4
The Champion of Linux: Linus Torvalds.....	4
The Origins of the Internet.....	5
Open Sourcing Netscape.....	6
The Mozilla Organization.....	6
The Competitive Landscape of the Browser Industry.....	7
The Driving Forces.....	7
The Five Forces of the Browser Industry.....	8
About The Rivalry.....	9
Bargaining Powers of Users.....	10
The Threat of New Entrants.....	10
The Bargaining Power of Suppliers.....	11
Substitution of Products.....	11
Opportunities.....	12
Threats.....	13
Value Chain Analysis.....	14
The Value Chain.....	15
Mozilla Foundation and the Value Chain.....	15
Inbound Logistics:.....	16
Operations.....	18
Outbound Logistics.....	21
Marketing and Sales.....	23
After Sales Services.....	25
Recommendations.....	26
The Commercial Value of Open Source Software.....	26
Sales & Marketing.....	28
Solicitation of Contributions.....	29
Growth of Market Share.....	29
Appendices.....	31
Appendix 1.....	32
Appendix 2.....	38
Appendix 3.....	40
Appendix 4.....	48
Appendix 5.....	49
Appendix 6.....	50
Appendix 7.....	59
Appendix 8.....	63
References.....	67

## LIST OF FIGURES

Table	Page
1. Five Forces of the Browser Industry.....	12
2. The Value Chain.....	15
3. Inbound Logistics.....	16
4. Operations.....	18
5. Outbound Logistics.....	21
6. Marketing and Sales.....	23
7. After Sales Services.....	25

## **Introduction**

This study is about open source software and whether it has commercial value in the marketplace. The topic has recently become of major interest in both the public and the private sectors with the onset of security threats and potential economic savings. An overview of the Open Source Movement and its impact on licensing and its influence on how business is conducted will be discussed. The driving forces and their impact on the browser industry is examined and its influence on the success of Mozilla.org, the largest open source project established. A close look at the value chain of the newly formed Mozilla Foundation is provided to detail any gaps found. Finally, recommendations are provided as to how the Mozilla Foundation can sustain and build its market share to compete in the marketplace and continue to pass on commercial value to other companies through its open source code.

## **Definition of Open Source Software**

The Open Source Movement started back in the 1970's with people who truly believed in the creed that one should do what one wishes to improve a product, charge for it if one likes, but share the underlying source code that has been added. (Raymond, 1999) The movement spread quickly with the emergence of the Internet in the 1990's and the rapid adoption of Linux, a free version of UNIX known for its stability and its core software programs that run corporate computer systems and networks. Programmers from around the world contribute freely to open source software development for the ego gratification and the peer recognition that it provides. The Open Source Movement is a model for software development and distribution that allows one to get both the executable version of a software program and the source code-text files that contain the programming language statements to perform various operations within the program. (Cervone, 2003) Open Source Software is not a new idea as its traditions go back to the beginnings of the Internet thirty years ago, but it has only recently

brought together the technical and the market forces that have drawn out its full potential.

(Raymond, 1999)

### **The Cathedral and the Bazaar**

Eric Raymond, a hacker of free software, gave a speech in May of 1997 called "*The Cathedral and the Bazaar*" that questioned the commercialization and the monopolistic ways of the software industry and more importantly, caused an explosion of interest in open source software. He gathered a group of other advocates for free software and formed the Open Source Initiative (OSI). (Sandred, 2001) Its importance was its ability to market the concept of Open Source, which they coined, and to create a series of guidelines to describe software that qualified as open source. The OSI has dedicated itself as managing and promoting the Open Source Definition that mandates:

Licenses shall not restrict any party from selling or giving away software...shall not require a royalty and the program must include source code, and allow distribution in source code as well as compiled form...

Its main focus is to promote this evolutionary process of open source software to produce better software than the traditional closed model, in which only a very few programmers can see the source and everybody else must blindly use an opaque block of bits. (OSI, 2003d)

### **Licensing Issues**

#### **Multiplicity of Licenses**

The multiplicity of licenses with various degrees of freedom and objectives points to a growing experimentation with new business ideas aimed at building commercial operations on the traditional "community" foundations of open source. According to Danese Cooper, The Open Source Community Liaison Officer at Sun Microsystems, there are licenses that range from the extreme right, to the middle and to the extreme left. To the extreme right are the conservative, copyrighting, and proprietary licensing. To the extreme left is the GNU licensing

known as the GPL Public Licensing, originated by Richard Stallman who declared it as “Copyleft” as its reversed from the name Copyright to show that open source software is not taking away the user’s freedom. It requires developers to provide other users the freedom to access and change source code and their enhancements when they distribute the programs and other applications derived from it. (Cooper, 2003) (See Appendix 1) Richard Stallman, founder of The Free Software Foundation, has become well known for advocating his belief that it is wrong for software patents to provide a 20-year lifecycle in addition to championing the term “Think free speech, not free beer.” (Sam, 2002)

Licenses found in the middle include the more permissive licenses such as the Berkley Software Distribution license (BSD) (See Appendix 2) and the Mozilla Public License (MozPL) (OSI, 2003c) (See Appendix 3) The Netscape Public License (NPL) was created before the MozPL as a compromise promoting free source development by commercial enterprises and protecting free source developers. The NPL was very much needed in order for Netscape to Open Source as the GPL was inapplicable for the commercial software developers who were required, if any new code goes into original code, to then be covered under the GPL. The problem with the GPL is it prohibits the addition of any restrictions or further permissions and insists that when you distribute code covered under its terms, it must be complete. Through the creation of the NPL, Netscape hoped to avoid breaking its contracts with its existing customer when it open sourced its core browser code. Revisions to the NPL that were created through a public process in which draft licenses were published and comments solicited on the Internet resulted in the final MozPL 1.0. As a result, all source code and modified code of Netscape was to be released under the NPL while any new code was to be released under MozPL or any other compatible license. [Sandred, 2001 #3 5]

## **The Dual Purpose of Licenses**

The licensing methods that are used by businesses influence the business models used by the software developers to market their products. Therefore, there is a dual purpose to licenses such that business models and software licensing methods are interlinked. Software developers have the availability to use a number of licensing methods when utilizing the open source approach. The real value of open source licensing is the guarantee of freedom that the licensing terms provide. Its important to the open source community to maintain this licensing as it ensures that source code will remain free and available for others to build upon it and to create new versions in the future. (Sayeed, Verma, & Bora, 2002) There are over 47 OSS licenses that have been approved by the Open Source Initiative (OSI, 2003a) yet only 5 or 6 of them are widely used with the GPL license dominating by 82% most open source projects. (Sayeed et al., 2002)

## **Open Source Software**

### **The Champion of Linux: Linus Torvalds**

By the 1990s, all the major components for a free operating system had been found or written except for the core element, the kernel. Linus Torvalds of Finland, best known for what has become known as Linux, developed the kernel and provided it for free via the Internet enabling anyone to use it for free. Combining Linux with the almost-complete GNU system of Richard Stallman resulted in the first truly open source operating system: a Linux-based GNU system, which became known as GNU/Linux. (FSF, 2003)

Through the essays that Eric Raymond published in the *Cathedral and the Bazaar* in 1999, the Open Source Movement found its new champion and preferred style of development in Linus Torvalds. Eric Raymond speaks of Torvalds style of development below:

Linus Toad's style of development—release early and often, delegate everything you can, be open to the point of promiscuity—came as a surprise. No quiet, reverent cathedral-building here - rather, the Linux community

seemed to resemble a great babbling bazaar of differing agendas and approaches...out of which a coherent and stable system could seemingly emerge only by a succession of miracles. (Raymond, 1999)

Raymond recognized that in order for there to be accomplishments in the Open Source Movement, the way that the work was done had to be changed. Raymond saw Torvalds' style as one where hackers, those who know the ins and outs of computers, needed to be constantly awarded by the prospect of having an ego-satisfying piece of the action that would be satisfied through the improvement in their work. What became known as Linus' Law, "*Given enough eyeballs, all bugs are shallow*" is what has made the idea of open source software achievable. This statement suggests that as open source software is peer reviewed by many through access of the Internet, bugs tend to be identified faster, fixed at a similar pace and results in quality assurance that make the software usable and of quality for commercial use. (Raymond, 1999)

### **The Origins of the Internet**

The Web came into focus to the world in 1991 when Tim Berners-Lee came up with a way to organize the information found on the Web by specifying the Web's underlying protocols; the sets of rules or standards that let computers communicate over the Internet and by building its basic tools. Marc Andreessen, a 21-year-old University student, working at the National Center for Supercomputing Applications (NCSA) in Champaign, IL is considered responsible for its commercial success that drew millions to the Web. He is credited with providing the first major web browser that handled graphics, enabled information to be displayed on the Web and made the Internet easy to use. The project known as Mosaic was posted on the Internet in beta form, free of charge, in March of 1993. When Andreessen met entrepreneur Jim Clark, formerly of Silicon Graphics, he immediately agreed to join him to form Mosaic Communications, before it was renamed Netscape for legal reasons. (Levy, 2003)

## **Open Sourcing Netscape**

In March of 1998, Netscape executives announced plans to publish the source code of its navigator web browser in the hopes of enlisting hacker support in its future. They were influenced by the writings of Eric Raymond whose first speech was in May of 1997 but not published until 1998. Netscape was also reacting to the competitive landscape dominated by Microsoft and planned to counter Microsoft's threat to its own market share. Quoted from Netscape's Press Release at the time of the release:

Netscape Communications Corporation (NASDAQ:NSCP) today announced the creation of mozilla.org, a dedicated team within Netscape with an associated Web site that will promote, foster and guide open dialog and development of Netscape's client source code." (Website, 1998)

In January of 1998 the first build of Communicator 5.0 offered its source code freely to the world making it available for modification and redistribution. Netscape believed that harnessing the creative abilities of all Internet developers would be an unprecedented way to advance the features and quality of the Netscape Communicator for all customers and that it would be a weapon to gain market share over its competitors. Mozilla, the code name for Netscape's first browser owes its name to a play on the lizard creature Godzilla that was supposed to annihilate Mosaic the software, thus resulting in *Mozilla*. At a later time Mozilla the lizard later became an engineering mascot at Netscape. (Reid, 1997) (See Appendix 4)

The following is found if one inputs "about:Mozilla" in any browser address field:

And the beast shall come forth surrounded by a roiling cloud of vengeance. The house of the unbelievers shall be razed and they shall be scorched to the earth. Their tags shall blink until the end of days. *from The Book of Mozilla, 12:10*

## **The Mozilla Organization**

The Mozilla Organization was formed initially to keep track of all the code and to provide a "Benevolent Dictator" to manage the process of incorporating new code into the code base of Communicator code releases. This code included bug fixes and feature additions

programmed by Netscape developers, as well as all the open-source developers on the Internet. It was important to Netscape in the beginning to create this organization to keep the releases limited and to ensure that there was not chaos in the process. It is one of the largest open source projects that have been created to date and is available to anyone with access to the Internet. (Hughes, 1999) Mozilla has remained important to the open source community as it provides choice to users and innovation of technology. Mitchell Baker, President of the Mozilla Foundation states in a posting on Mozilla.org the goal of the Mozilla Foundation:

The goal of the Mozilla project is to promote innovation and enable the creation of standards-compliant client technology to help keep content on the web open. The key to open content is not any particular browser application, but openness, standards compliance and cross-platform technologies. The more people who use browsers based on open, standards-compliant technologies, the better the chances we will all enjoy viable choices in the way we conduct digital transactions. (Baker, 2003a)

## **The Competitive Landscape of the Browser Industry**

### **The Driving Forces**

Industry and competitive conditions change because forces are constantly in motion that creates incentives for change. The most dominant forces are known as driving forces as they have the most influence on what kinds of changes will take place in the industry's structure and competitive environment. (Porter, 1985) Jim Rapoza, an analyst of browsers with ZDNet from the time when Mosaic was vying with Spyglass (the forebears of Mozilla and IE) notes the driving forces behind the success of the dominant browser in the marketplace results:

As companies build more new Web applications and Web services, they will be strongly attracted to a state-of-the-art browser that powerfully interact with the applications and services they are creating. Commercial Web sites, many of which are seeing a resurgence in their business, will also want to leverage new technologies that only up-to-date and innovative browsers support.

The *Critical Driving Force* behind the large-scale downloading of browsers results from the impact of the Internet and its technologies that became mainstream to the public in the 1990's.

Additional driving forces that strongly influence the browser industry include:

*Innovation:* The Technology Industry is measured in terms of how fast it creates new and useful technology. Just as Moore's Law stated that the number of transistors on a given chip can be doubled every two years and has become the guiding principle of progress in electronics and computing (Webopedia, 2003) innovation is the impetus that pushes for new features expected to be faster and even more innovative than the previous version.

*Speed:* The Internet drives companies to innovate 5 or even 10 times faster than most technological innovations. (Reid, 1997) In terms of browser innovations, web pages continue to become larger and in order for browsers to compete they need to download information as quickly as possible.

*Perception:* End users must believe that they are using the most recent and most up-to-date, feature laden, powerful browser version available in order for them to download it.

### **The Five Forces of the Browser Industry**

The "five forces" is a framework used to analyze the nature of competition and the potential profitability of an industry. Michael Porter of the Harvard Business School has created a framework to apply this analysis. Below is a discussion of the competitive landscape of the browser industry. The old rules of the game for traditional business forms are no longer valid due to the Internet, which is fast and unpredictable and constantly changing. Competing on *Internet Time* is about moving rapidly to new products and markets, becoming flexible in strategy, structure and operations and exploiting all points of leverage for competitive advantage. (Cusumano & Yoffie, 1998) In the following section about the five forces, each of Michael Porter's five forces will be analyzed and presented for the Browser Industry, its impact on the Mozilla Foundation and specifically on the Mozilla Browser.

## **About The Rivalry**

The Mozilla Foundation is facing strong competition within the browser Market. The Market share is not evenly distributed with Microsoft having captured 95.4% of the market with all other leading browsers maintaining less than 5% and Mozilla having only 1.6% but growing. (Hicks, 2003) In spite of Microsoft being the leading rival, its plans to keep its latest browser, Explorer 6.x as its last standalone version of the Web browser and to have all future versions embedded as features in new versions of Windows provides opportunity for competitors to gain market share. It is expected to be more than three years before a new version of Windows achieves any significant adoption; allowing competitors to gain market share significantly. (Rapoza, 2003a)

The Internet creates an extremely competitive stage where rivals must be flexible and willing to innovate rapidly and with superiority in order to succeed. (Cusumano & Yoffie, 1998) The barriers to exit and the barriers to enter are difficult due to users' brand loyalty.

Customer loyalty is strong in the browser market. Users become comfortable with the features of a particular browser such as its bookmarks and are reluctant to switch to competitor browsers. In addition, most users are not browser savvy and are unfamiliar with the benefits of upgraded browser features, choosing not to switch as a result. (S.R., 2002)

Another aspect creating strong rivalry within the browser market is the battle between the Open Source Movement that is championed by the users or corporations in protest, mostly against Microsoft, proprietary software. (Waters, 2003a) The Mozilla Foundation also has a competitive advantage of utilizing a virtual army of debuggers that outnumber the number of paid Quality Assurance (QA) testing personnel that Microsoft, a closed source company, employs. (Lerner & Tirole, 2002)

## **Bargaining Powers of Users**

Users have strong bargaining power within the Browser Market. Users are brand loyal as they become familiar and dependent on certain features of its browsers, resulting in high switching costs due to reluctance to switch brands and the complexity of the process of switching. It is relatively easy for users to gain Information about competitor products, especially through the Internet. It is relatively easy to learn about improved features found in company's latest versions that could easily lead them to switch. As browsers are free, pricing is not an influencer in user's decision as to which browser they will choose. (Cusumano & Yoffie, 1998)

Users or companies can easily backward integrate through use of the open source software. Users can create their own browsers based on the Mozilla code and the gecko engine, which is robust, proven and available to anyone for free. The gecko engine is designed to support open Internet standards such as HTML, XML, JavaScript and many others and to provide a layout engine that takes content and formats the information (such as Cascading Style Sheets, hard-code HTML tags, etc.) and displays the formatted content on the screen. Any organization or company not wanting to independently develop their own browsers can take advantage of the open source Mozilla.org code and the gecko engine. (Mozilla.org, 2003a)

## **The Threat of New Entrants**

The threat of new entrants entering the Browser Market is weak as there is strong brand loyalty that makes it difficult for other competitors to enter the industry. It is also difficult for competitors to gain market share due to Microsoft's IE's huge lead at 95.4%. (Hicks, 2003)

Although there is opportunity for new entrants it is limited to a few niche players. There is growing resentment against Microsoft due to the Justice Department's case against it. People are now seeking more choices for browsers in the market place. Users are concerned about using Microsoft products due to viruses that infect their desktops such as in September of

2003 when Microsoft was forced repair flaws found in its Windows operating system and at the same time failed to deal with other virus related problems. (Waters, 2003b).

### **The Bargaining Power of Suppliers**

The Bargaining Power of Suppliers is strong as it is the browser companies that are creators and thus the ones in control of the source code. Microsoft has the strongest bargaining power as it has the largest market share at 94.5% (Hicks, 2003) and it keeps the source code proprietary. As Microsoft is a monopoly it is able to exercise its strength with its vendors by dictating bundling of the IE browser through their vendor's products. (Stellin, 2002) Browser companies have little bargaining leverage with users, as there is no cost involved in the development of the product. Mozilla has the little bargaining leverage as it makes its code available to its users through open source. (Cusumano & Yoffie, 1998)

### **Substitution of Products**

There is a threat of substitution of superior products in place of the original Mozilla products due to the Mozilla code being open sourced. Competitors and savvy users are enabled to build browsers on top of the original code. There are a number of vendors that use the Mozilla platform, the Mozilla browser, the source code, or Mozilla development tools to create applications and other products. The categories that they fall into include Mozilla Application Suite, Mozilla/Gecko Embedded Applications, Bugzilla and Other Development Tools.

(Mozilla.org, 2003a)

The Five Forces of the Browser Industry			
Force	Strong	Weak	Comments
Rivalry	x		-Leading Rival MS has 95.4% market share -Barriers to enter difficult
Bargaining Power	x		-Switching Costs difficult -Strong rivalry between proprietary and open source
Threat of New Entrants		x	-Strong brand loyalty -Opportunity as users want more choice
Bargaining Power of Suppliers	x		-Suppliers are the creators of code giving company control -Microsoft has monopoly thus ability to control vendors
Substitution of Products	x		-Source code is open allowing for others to build their own substitutes easily

Figure 1 (Rapoza, 2003b), (S.R., 2002), (Cusumano & Yoffie, 1998), (Lerner & Tirole, 2002)

The following *Opportunities* and *Threats* in the Browser Market have been identified as a result of the Five Forces Analysis discussed above.

### **Opportunities**

There is growing interest from foreign governments to purchase software, almost 17 billion in 2002 with an expectation of it growing 9% a year for the next five years as estimated by the market research IDC. This is a crucial market that at this time Microsoft only controls a relatively small part leaving it available for alternative opportunities. There are a number of governments ranging from Peru, India, China, U.K., Germany and France seriously analyzing Open Source procurement policies. (Stanco, 2003) The Spanish government is in the process of acquiring over 80,000 Linux desktops for its education department saving Spain over 20 million euros a year. When the Brazilian government choosing to use open source software, as the Brazilian government has, they chose the Mozilla browser along with it. Japan has recently declared that it will collaborate with China and South Korea to develop an

open source alternative to Microsoft providing an additional opportunity for Mozilla to gain market share. ("Open Source Software: Microsoft at the Power Point," 2003)

Due to the increased security risks found in proprietary software such as Microsoft, Open Source has become increasingly popular as an alternative. Concerns for security have become more prevalent with the influx of viruses such as the Blaster and SoBig, which have in turn increased the interest in OSS. An advantage of OSS is that it allows programmers to root out bugs from code. It is also attractive as software can be tailored to the user's specific needs and the user can choose upgrades when they want them, not when the vendor dictates them. ("Open Source Software: Microsoft at the Power Point," 2003)

Browsers have improved greatly in the past few years mainly due to Linux and the Open Source model offering competitors the opportunity to gain market share based on the success of their marketing, promotion of their superior features and the willingness of users to download and experiment with new browsers. (Asaravala, 2003)

### **Threats**

Mozilla is threatened by Microsoft's market share of 95.4% which makes it difficult for them to gain much needed market share. (Hicks, 2003) It is also threatened by its lack of brand name recognition in the market place that hinders its growth.

The amount of revenue generated within the Mozilla Foundation is minimal. They are a non-profit organization dependent on the contributions of corporations and individuals to provide funding. This may not be sufficient enough to support the Mozilla Foundation in the long run. There is the question as to whether companies will continue to provide contributions in the future and if so, for how long.

The Launching of the Mozilla Foundation:

In this paper a particular OSS project, the Mozilla Browser 1.5 developed by the Mozilla Foundation will be discussed through an analysis of its Value Chain to determine if indeed OSS is capable of providing commercial value to the market. Mozilla started back in 1998 when Netscape decided to provide its source code to the public. It was then that the Mozilla Open Source project was founded. It was initially created by Netscape to promote the Mozilla browser and e-mail software. (Wagner, 2003) AOL acquired Netscape in late 1998. As part of a lawsuit settlement with Microsoft in May of 2003, AOL gave into Microsoft in the browser space when it negotiated a seven-year, royalty-free license to use IE with its AOL client software. AOL laid off about 50 software developers working on browser development including Mozilla and Netscape.

The Mozilla open source project of Netscape announced the creation of the Mozilla Foundation on July 17, 2003. Several of AOL's software developers were maintained for the foundation, which also received a \$2 million donation over two years from AOL. Mitch Kapor, an advocate of Open Source Software and the founder of Lotus Development donated \$300,000. (Wagner, 2003) Additional funding came from corporate supporters including \$150,000 contributed by IBM along with expectations for more from enterprise type of companies. Further contributions are received through the Mozilla.org website where donations are welcomed.

### **Value Chain Analysis**

The Value Chain is the set of activities through which a product or service is created and delivered to its customers. Information technology has resulted in a determined influence on the

value chain as every activity involves the creation, processing and communication of information. The Internet thus has the special advantage of linking one activity with other activities and creating instantaneous or real-time data making that data widely available, both within and outside the company. (Porter, 2001) Once activities are defined in an organization, an analysis of each step found in the value chain can identify both costs incurred and value created. Emerging new models that result from new technologies such as the Internet have reshaped the global business landscape and are redefining power. (Applegate, Austin, & McFarlan, 2002) A close look at the value chain, as diagramed below includes the following primary activities.

### **The Value Chain**

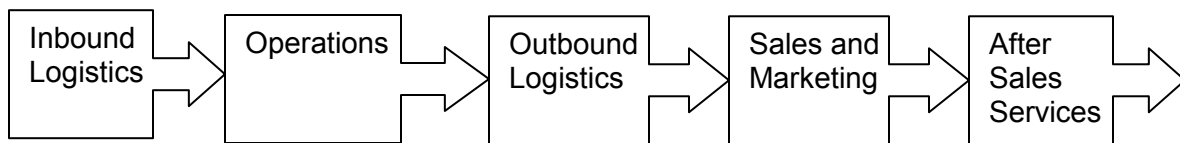


Figure 2

### **Mozilla Foundation and the Value Chain**

By examining the Value Chain of the Mozilla Foundation we can determine how well each segment performs and where there is room for improvement in those areas. The Internet influences the newly formed Mozilla Community and it works within the framework of Open Source Software (OSS). OSS involves software developers at many different locations and organizations sharing code to develop and refine software programs. (Lerner & Tirole, 2002) The product developed by the Mozilla Foundation that is being closely looked at in this study and found on the Mozilla.org website is the latest browser known as Mozilla 1.5. For convenience, Mozilla 1.5 will be referred to as the Mozilla browser in this study.

## Inbound Logistics:

Inbound Logistics	
Organizational Structure	Organization of virtual leadership consisting of Staff, Associate members and virtual contributors. Most authority is delegated to the 88-module owner and test teams, but mozilla.org reserves the right to determine module ownership and to resolve conflicts.
The Roadmap	Anyone can submit a problem report or request an enhancement, but Mozilla.org controls the direction of the project as defined by the Roadmap

Figure 3 (Audris, 2003), (Eich & Hyatt, 2003)

The first stage of the value chain, Inbound Logistics, found within the framework of the Internet consists of the activities involved in getting orders over the web captured, processed, picked, shipped, delivered and invoiced. These activities include real time integrated scheduling, shipping, warehouse management, demand management, planning and dissemination of inventory data across the company and for its suppliers. (Bowles, 2003), (Porter, 2001) Inbound logistics for the Mozilla browser do not involve many of those activities listed as the browser product is offered to the public for free and all of its activities result from the contributions of an open source organizational structure consisting of a network of virtual contributors from individuals and organizations around the world. Its main inbound logistic activities involve the virtual community of the Mozilla Foundation and its defining of projects through a *roadmap*.

The organization of the Mozilla Foundation is organized virtually with 10 paid staff members at this time with the hopes to increase that amount and additional associate members who make up a consensus-based type of authority. Two members are the ultimate decision makers within the organization. Mitchell Baker, known as the Chief Lizard Wrangler, who is the ultimate decision maker for The Mozilla Foundation regarding any non-technical issues while Brendan Eich, creator of the scripting language JavaScript, ultimately decides technical issue along with any module ownership issues. Although each staff member has their expertise,

they all manage to work with one another and on a variety of projects mainly virtually. Anyone who is a contributor to the Mozilla project has access to any of the staff members. (Baker, 2003b)(See Appendix 5)

Successful projects depend on the organization's ability to break the projects into distinct components known as modules. At the Mozilla.org there are 88 module owners responsible for project development such as fielding bug reports, enhancement requests, patch submissions and so on. The owner of a module is expected to facilitate good development, as defined by the developer community and is ultimately in control of the direction of the project. (Mozilla.org, 2003e) The Mozilla Foundation ultimately reserves the right to determine module ownership and to resolve conflicts. (Audris, 2003) Brendan Eich retains authority over all module owners and is responsible for both the architecture and the technical direction of the Mozilla browser.

Mozilla.org, as an open source project, requires the management of a virtual team made up of contributors from around the world and this requires unique demands including expertise knowledge and wisdom from its leaders. (Sandred, 2001) It also warrants leadership that provides vision, attracts other programmers and keeps the project together to prevent it from forking or being abandoned. (Lerner & Tirole, 2002) Talented programmers who are considered to be the best in their areas of expertise and who will take ownership seriously are sought as leaders as is detailed in Mozilla.org's most current roadmap: (See Appendix 6).

There is no substitute for leadership by an "application czar" Therefore we will promote strong ownership by relieving vigilant owners, on a case-by-case basis, of mandatory super-review requirements, for modules that the super-reviewers deem sufficiently well-owned. And where ownership is weak or missing, we will take steps to find an owner, or absent any candidates, to reduce our dependencies on the under-owned or un-owned code. (Eich & Hyatt, 2003)

The Mozilla browser is an Open Source Project and it is essential for Mozilla.org to maintain an organizational structure that enables their products to be of superior quality and to be completed in a timely manner. Structure begins with the formation of a *software roadmap* that provide the direction it should take to build its projects and have them available on its website. The roadmap determines the success of a project as it includes the decisions and their rationales along with reflecting unforeseen and unexpected developments as they emerge. The roadmap consists of *software milestones* that are included in the roadmap and determine whether or not the project will meet the anticipated schedule along with rules of development, discussion and rational for its development plans. Mozilla.org maintains not only its current but also its two previous roadmaps from its beginnings in October of 1998. (See Appendix 7 & 8)

## Operations

Operations	
Development Work through the Bugzilla Database tool	Developers use the Bugzilla database tool to find problems or enhancements on which to work. Bugzilla can be used to request help on a particular change, to submit their code, to keep a to-do list as well as to prioritize, schedule and track dependencies
Binary Versions of Code Released Daily for Milestone Testing	Development, based on what is referred to as milestone releases, act as periodic checking points on projects and add additional infrastructure, features, and bug fixes
Prerelease Testing	Quality Assurance Smoke Screen tests are performed on the daily builds. Six test teams maintain test cases, guidelines, training materials etc. on Mozilla.org website.
Inspections	Two stages of inspections are conducted. One at the module level, and one by a member of the "super reviewer" group who focus on how a patch fits into the Mozilla code base.

Figure 4 (Audris, 2003), (Mozilla.org, 2003a)

Operations in the Value chain within the framework of the Internet refer to the scheduling and the decision making within the company along with working with contract assemblers and suppliers. (Porter, 2001) At The Mozilla Foundation the main operations that take place are

the building of open source projects such as the Mozilla Browser 1.5 resulting from the debugging of the source code, testing and Inspecting of the code for release.

When code is completed it is necessary to debug it and rid it of any flaws in the source code. The way to debug is by discovering the ways in which the Mozilla client software's actual behavior deviates from its expected behavior, normally determined by Internet specifications. Specifications are determined by the Internet Engineering Task Force (IETF) that concerns itself with "near-term" technical problems of the Internet, the World Wide Web Consortium (W3C) that develops interoperable technologies such as specifications, guidelines, software and tools for the Web and through the Mozilla Foundation. (University, 1996)

Compatibility of standards is an important issue to the development of open source software. Standards leave room for differences in interpretation and implementation such as from vendor. Until Mozilla and Netscape 6 (e.g. HTML 4.0, CSS1, CSSP, W3C DOM level 1) there was no other browser that had such high degree of support of many of the existing standards. The practical approach for building applications is to study the standards, study each vendor's implementation, find the common functionality, and build on top of that. (Krock & Ermakov, 2000)

The Bugzilla database, a tool maintained by the Mozilla Foundation, allows individuals or groups of developers to report bugs and assigns these bugs to the appropriate module owners. This is an example of a class of programs called "Defect Tracking Systems." Most commercial defect-tracking software vendors charged enormous licensing fees, and Bugzilla has become a favorite of the open-source crowd with its genesis in the open-source browser project, Mozilla. It is now the de-facto standard defect-tracking system against which all others are measured. There are over 110,000 bug reports in Bugzilla with between 100 and 300 bugs reported daily. There are over 38,000 Bugzilla accounts. (Dotzler, 2003)

Developers also utilize Bugzilla to keep a to-do list as well as to prioritize, schedule and track dependencies. (Mozilla.org, 2003b)

Testing binary code of the Mozilla browser is produced daily for testing purposes only and is made available to volunteer programmers to find problems or to provide their own enhancements. Binary code is source code (logic) that has been compiled into a binary format meeting a specific set of CPU instructions (i.e. 32 bit X86, 64 bit X86, 32 bit PowerPC, 32 bit Linux, 32 bit .NET, J2SE JVM). Binary code is source code (logic) that has been compiled into the machine level language of bits and bytes. (Edwards, 2003) This source code is made available for testing to both developers and non-developers before the product is complete in what is known as alpha and later, beta code. (Hughes, 1999) An “accidental product” results at times in the process of open sourcing, enabling the binary code to be modified and fixed thus becoming a product of unintended consequences.

Development in the Mozilla.org is based on what is referred to as milestone releases, acts as periodic checking point for projects and provides opportunity at that time to add additional infrastructure, features, and bug fixes. These volunteer developers and programmers from around the world download more than 2,500 download of daily verification builds, approaching over 200,000 downloads a day. There is crash data that results from more than 8,000 Milestone users in a peak day and about 70,000 users across the entire Milestone cycle. (Dotzler, 2003)

Quality Assurance Smoke Screen tests on the daily builds performed by six teams assigned to various parts of the product. They maintain test cases, guidelines, training materials etc. on the Mozilla.org website. There are minimal sets of test cases, which are required to pass/work in order to bless a Mozilla build as stable and permit developers to continue work.

Upon completion of testing, two stages of inspections are conducted. One at the module level, and one by a member of the “super reviewer” group who focus on how a patch fits into the Mozilla code base and makes all the major decisions. Module owners must approve all changes in their modules.

### **Outbound Logistics**

Outbound Logistics	
Downloading of product over the Internet	Use of the Internet’s FTP protocol that enables users to download product, users receive product
Capital Investments in Open Source Projects using the Mozilla Code	Companies such as Sun Microsystems, IBM, and Red Hat that release existing proprietary code and create some governance structure for the resulting open source process

Figure 5 (Lerner & Tirole, 2002), (Reid, 1997)

Outbound Logistics in the Value Chain within the framework of the Internet consists of the transactions that take place over the Internet, automated customer agreements and its delivery status. (Porter, 2001) The Mozilla Foundation’s primary outbound activities consist of the downloading of its browser and other products. As the Browser and other products are available for free, customer agreements and delivery status is not necessary. Additional ways that Mozilla product is made available to the public is through companies that develop and use the Mozilla source code in their own projects and either they keep their projects open source or they commercialize the source code for their own products.

The Internet has a set of commands that enable computers to implement specific operations across a network and they are known as protocols. FTP or File Transfer Protocol allows Users download programs and other files from distant machines. (Reid, 1997) The Mozilla Foundation takes advantage of the Internet’s efficiency, ease and speed due to the Internet’s FTP protocol and offer distribution of its Mozilla browser along with other software packages for free over the Internet. Other products that they distribute include the following products:

- Mozilla: the Mozilla Internet application suite.

- Firebird: a redesign of the Mozilla browser.
- Camino: a native browser build for Mac OS X.
- Calendar: the calendar client.
- LDAP: software development kits including c-sdk, java-sdk and perldap
- Grendel: a Java mail client.
- JavaScript: JavaScript Reference (JSRef) implementation and Rhino (JavaScript in Java) implementation.
- CCK: Localization Kits for Netscape Communicator.
- Messaging SDK: Netscape Messaging Access SDK
- Bugzilla: Web based bug-tracking system.
- NSS: Network Security Services

(Mozilla.org, 2003a)

Increased distribution and larger market share results from other company's development and use of the Mozilla source code in their own projects. The availability of embedding the modular and the standard's compliant Gecko in web applications is attractive to companies. The Gecko engine is the heart of a browser that controls how a Web page appears on a screen. One such company that has chosen to do that is Sun Microsystems. Sun Microsystems' involvement in Open Source has a long history back to the founders who were involved in Open Source Software from the time when they worked on their own open source projects at Stanford University. Sun Microsystems has provided the Netscape browser to its customers from the time of its first releases and continues today with its support of the Mozilla Foundation. Sun supports the Mozilla Foundation through contributed code that is developed by about 20 dedicated engineers (previously there were 50) based in Beijing and through contributions such as the \$300,000 contributed upon the creation of the Mozilla Foundation in July of 2003. (Sasaki, 2003)

It is necessary for a company such as Sun Microsystems to integrate, polish, test and optimize the open source code in order to integrate it as part of their own products. Sun has customized Mozilla 1.2.1 for its own Solaris Operating System, which includes Improved

navigation and filtering, and Integrated enterprise features. Sun has also contributed key technologies to the Netscape Gecko and ensures the seamless integration with the Java platform. According to Curtis Sasaki, Vice President of Desktop Solutions at Sun Microsystems, companies see Open Source as providing choice and through its development of it they are not locking anyone into a single vendor. He also feels that the individual developers who work on Open Source may have their own agenda and may view Open Source as more of a community. It is thus important for companies such as IBM HP, Red Hat and others that develop open source code to support the Open Source Movement and projects to promote choice. (Sasaki, 2003) OSS is important enough to Sun Microsystems that it maintains its relationship with the Mozilla Foundation with the employment of a Community Liaison Officer to oversee its relationship.

### Marketing and Sales

Marketing and Sales	
Marketing of Mozilla	All advertising and promotions of Mozilla.org are made through the website
Sales	Distribution of product through website. Product available through website. Merchandise also offered.

Figure 6 (Mozilla.org, 2003f)

The value chain functions of Marketing and Sales on the Internet are expected to be online, real time and with an emphasis on advertising. The expectations for being online include channel sales including websites and marketplaces and for real time inside and outside access to customers, products information, dynamic pricing, inventory availability, online submissions of quotes and order entry. (Porter, 2001)

Mozilla.org until recently existed as a division within Netscape and then under the parent company of AOL/Time Warner. They previously had the marketing resources available to them to work with product managers in addition to promoting their products with both national

and international large advertising campaigns. As a non-profit organization now, with the emphasis on product development, sales and marketing appear to not have the emphasis they had on them while at a commercial company.

The Mozilla Foundation has evolved into a loosely structured virtual organization that is driven by engineers. The functions of marketing and sales are not highly evolved. Nor is there any formal function established for Public Relations or Advertising. (Cooper, 2003) Mozilla.org utilized their website to provide information and to advertise its products via their website. Through Mozilla.org on its first page known as a home page, important information is promoted through content and links to other areas within the website. They do not conduct any forms of advertisement outside of the information provided on their website. A new product that is currently being developed into a product available for downloading shortly is the Mozilla Firebird, known as a “best of breed” browser that is cross platform available. The only marketing consists of a “why” document with detailed information about the updates and features included and “sales” though downloads will be made available through a new product site. (Mozilla.org, 2003d)

Products developed through the Mozilla Foundation are all available at no cost. Recently, The Mozilla Foundation enabled users with the opportunity to purchase their products through a CD that is available for \$3.95 on their website. It includes Mozilla 1.5, Mozilla Firebird .7, Mozilla Thunderbird .3, Camino .7, Bugzilla, and the Mozilla Code. (Mozilla.org, 2003h) They are also offering merchandising products such as the Mozilla T-Rex T-shirt that has recently become available for \$12.95 - \$15.95

## After Sales Services

After Sales Support	
End User Support	Mozilla. Org has outsourced end user support through an outside vendor DecisionOne for \$39.95 per incident
Customer Information and Contacts	Information is made available to customers through the Mozill.org website such as through “Mozilla at a Glance” page or through Newsgroups and e-zines such as Mozillazine.com

Figure 7 (DecisionOne, 2003) (Mozilla.org, 2003f)

After Sales Services in the Value Chain within the framework of the Internet may consist of online support through email response management, billing integration, cd-browse, chat, customer self-service via websites and real time field service with access to customer account review. (Porter, 2001) The Mozilla Foundation provides timely customer information through its website through a number of online resources as listed below.

*Mozilla at a Glance* is the primary document found within Mozilla.org that provides numerous links to learn about the website and information on how to reach the leaders of the organization. Email addresses for all their current staff members are available to anyone wanting a correspondence with the organizers of Mozilla.org. (Mozilla.org, 2003f) Numerous newsgroups are also maintained on the website that provide additional means of communications for Mozilla Users. Websites also work closely with Mozilla providing content about Mozilla such as Mozillazine.com that provided updates and news throughout the week and Blogzilla, which is a weblog that cover news, tips, info, links, and hacks related to the Mozilla project. (Mozilla.org, 2003c)

There is limited live support to users that download a product from the Mozilla.org website as they do not have the resources. Support has always been available to users for free through newsgroups or through information found on the web such as Newzilla, Mozilla1.5 FAQ, Mozilla.org end user documentation, *Mozilla Getting Started Guide*, *Gunnar’s Unofficial*

*Mozilla Help Site* etc. In fact, this informal support that is available for free through the newsgroups and documentation is seen as especially helpful. *The Mozilla Champions* provides end user support through newsgroups and is made up of a team of volunteers through invitation only.

When one contributes to the community such as Eric Krock, former Group Product Manager for Netscape's Gecko who has filed numerous Bugzilla reports and fixes for bugs, there is tremendous support and the capability to see the status of the reports. There is quality support offered for free by Mozilla.org unlike Microsoft for its IE browser. Many users though are not contributors and are not sophisticated Internet users and are in need of end user support. (Krock, 2003) In attempts to reach out to a larger market, the Mozilla Foundation recently has outsourced end user support for Mozilla 1.5 and above through DecisionOne for \$39.95 per incident. This company is the vendor that Netscape used in the past and they are based out of Philadelphia. (DecisionOne, 2003) There are plans to provide email support at a later date but at this time all support is made available through DecisionOne.

Recommendations

### **The Commercial Value of Open Source Software**

The commercial value of OSS can be found in the study conducted at Carnegie Mellon University consisting of a comparison of the OSS development of Apache, an open source web server and the development of Mozilla, the browser. It was concluded that Open Source Management could be successful for commercial development of software products.

The particular way that the core team in Apache (and, we assume, many other OSS projects) is formed may be one of the keys to their success. .... We believe that for some kinds of software, in particular those where developers are also highly knowledgeable users, it would be worth experimenting, in a commercial environment, with OSS-style "open" work assignments. (Audris, 2003)

Open source software has gained a reputation for reliability, efficiency and functionality that has surprised many people in the software engineering world. The Internet has facilitated the coordination of volunteer developers around the world that has produced open source solutions that have become the market leaders in their sector. (Lerner & Tirole, 2002)

However most of the users of these applications are relatively technically sophisticated while the average desktop user is using standard commercial proprietary software.

There is great opportunity for OSS to become more readily available than proprietary software in the future. The main problem holding back OSS growth, as concluded in the paper by David M. Nichols, is that the involvement of most of the users of applications are technically sophisticated while the average desktop user is not very sophisticated and is lacking in basic computer skills. It was concluded in the study software developers in the future need to design better for the usability of OSS projects. It was also determined that if typical users are included in the development process of OSS projects, this can be accomplished. (Nichols & Twidale, 2003)

OSS provided by the Mozilla Foundation through Mozilla.org is available to be downloaded for free via the Internet, minimizing its potential for commercial value. To create value it may be that they need to be more commercial and to do this they may need to make end user support available; a tremendous cost to a non-profit organization. Although the potential to create commercial value is accessible to the Mozilla Foundation, they ought to continue to remain the home of Mozilla.org whose purpose is to act as the virtual meeting place for the Mozilla code and to provide a central point of contact and community for those interested in using or improving the source code. Thus, through the continued existence of the Mozilla Foundation, commercial value is made available and transferred to other companies, who as for-profits, can take full advantage of its open source code for commercial purposes.

Below are recommendations to enable the Mozilla Foundation to better reach the market and build up their market share to its hoped for target of up to 20% ensuring its continued survival

### **Sales & Marketing**

Develop an Open Source Community Field Sales force that would act to provide information about products before and after sales. This would be an extension of Mozilla.org open development group that codes its software. It would be organized in a similar manner that the modules are organized with a designated leader. It would provide an opportunity for those that participate to have peer recognitions for their contribution to the community. Mozilla products are available for free and the focus should be on growing market share.

- Those that participate in the volunteer Field Sales would be invited based on their previous commitment and contribution to the community such as their involvement in newsgroups and projects
- Modules similar to the ones for development would be the organizational structure and would be divided up into the targeted selling areas including education, government, corporate and end users. Include a module that would focus on public relations, an area lacking in the value chain at this time.
- Ongoing online training of products would be made available to participants on Mozilla.org.
- Target users that are not as Internet sophisticated as has been the case in the past. Consider those users when developing new software
- Recruit volunteers seeking real hands on experience from marketing organizations such as the Marketing Association of America and Universities
- Opportunities to meet face-to-face with supporters and those seeking to convert. This would be encouraged at events such as Linux World held annually in San Francisco and other relevant seminars

## **Solicitation of Contributions**

Contributions from users and corporations are the main source of revenue for the Mozilla Foundation as they are a non-profit organization. To sustain the Mozilla Foundation there should be solicitation campaigns targeted at the online community at large.

- Target companies for contributions that have worked with and utilized the Mozilla browser, the source code, or Mozilla development tools to create applications and other products and fall
- Work closely with the W3C, OSAF, OSI and other organization that have a number of members that support open source and use it in their products
- Suggest that anytime a user downloads any product that they contribute in order to sustain the community
- Establish a module of volunteers that will be responsible for fund raising campaigns to raise money for the Mozilla Foundation

## **Growth of Market Share**

To sustain the Mozilla Foundation its ultimate goal is to increase adoption as mentioned by Mitchell Baker, President of the Mozilla Foundation, to 5% to 20% of the browser market.

(Baker, 2003b) To do this the Mozilla Foundation should consider:

- Increase Brand recognition by requiring companies that include Mozilla applications in their products to include the Mozilla logo; similar to the Intel logo included on products that are built with its chip
- Enter new markets such as government and enterprises with the dedication that has been established with the end users. Establish modules similar to the ones established for engineering to pursuit these new markets.
- Educate the Internet community about Open Source Software. Recruit a group of volunteers to represent the Mozilla Foundation and educate those in schools,

at seminars and conferences about Mozilla. Establish the important perception through education that Mozilla is the fastest, easy to use and most feature-laden browser to choose from in the marketplace.

## Appendices

## Appendix 1

### GNU GENERAL PUBLIC LICENSE (FSF, 1991)

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so

that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three

years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the

original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will

be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## Appendix 2

### The BSD License (OSI, 2003b)

The following is a BSD license template. To generate your own license, change the values of OWNER, ORGANIZATION and YEAR from their original values as given here, and substitute your own.

*Note:* The advertising clause in the license appearing on BSD Unix files was officially [rescinded](#) by the Director of the Office of Technology Licensing of the University of California on July 22 1999. He states that clause 3 is "hereby deleted in its entirety."

Note the new BSD license is thus equivalent to the [MIT License](#), except for the no-endorsement final clause.

<OWNER> = Regents of the University of California  
<ORGANIZATION> = University of California, Berkeley  
<YEAR> = 1998

In the original BSD license, both occurrences of the phrase "COPYRIGHT HOLDERS AND CONTRIBUTORS" in the disclaimer read "REGENTS AND CONTRIBUTORS".

Here is the license template:

Copyright (c) <YEAR>, <OWNER>  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR

PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE

## Appendix 3

### The MOZILLA PUBLIC LICENSE (OSI, 2003c)

#### Version 1.1

##### 1. Definitions.

**1.0.1. "Commercial Use"** means distribution or otherwise making the Covered Code available to a third party.

**1.1. "Contributor"** means each entity that creates or contributes to the creation of Modifications.

**1.2. "Contributor Version"** means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

**1.3. "Covered Code"** means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof.

**1.4. "Electronic Distribution Mechanism"** means a mechanism generally accepted in the software development community for the electronic transfer of data.

**1.5. "Executable"** means Covered Code in any form other than Source Code.

**1.6. "Initial Developer"** means the individual or entity identified as the Initial Developer in the Source Code notice required by **Exhibit A**.

**1.7. "Larger Work"** means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

**1.8. "License"** means this document.

**1.8.1. "Licensable"** means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

**1.9. "Modifications"** means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

**A.** Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

**B.** Any new file that contains any part of the Original Code or previous Modifications.

**1.10. "Original Code"** means Source Code of computer software code which is described in the Source Code notice required by **Exhibit A** as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

**1.10.1. "Patent Claims"** means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

**1.11. "Source Code"** means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

**1.12. "You" (or "Your")** means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

## **2. Source Code License.**

### **2.1. The Initial Developer Grant.**

The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and

(b) under Patents Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof).

(c) the licenses granted in this Section 2.1(a) and (b) are effective on the date Initial Developer first distributes Original Code under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: 1) for code that You delete from the Original Code; 2) separate from the Original Code; or 3) for infringements caused by: i) the modification of the Original Code or ii) the combination of the Original Code with other software or devices.

### **2.2. Contributor Grant.**

Subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code and/or as part of a Larger Work; and

**(b)** under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: 1) Modifications made by that Contributor (or portions thereof); and 2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

**(c)** the licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first makes Commercial Use of the Covered Code.

**(d)** Notwithstanding Section 2.2(b) above, no patent license is granted: 1) for any code that Contributor has deleted from the Contributor Version; 2) separate from the Contributor Version; 3) for infringements caused by: i) third party modifications of Contributor Version or ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or 4) under Patent Claims infringed by Covered Code in the absence of Modifications made by that Contributor.

### **3. Distribution Obligations.**

#### **3.1. Application of License.**

The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

#### **3.2. Availability of Source Code.**

Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License either on the same media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available via Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party.

#### **3.3. Description of Modifications.**

You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

#### **3.4. Intellectual Property Matters**

**(a) Third Party Claims.**

If Contributor has knowledge that a license under a third party's intellectual property rights is required to exercise the rights granted by such Contributor under Sections 2.1 or 2.2, Contributor must include a text file with the Source Code distribution titled "LEGAL" which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If Contributor obtains such knowledge after the Modification is made available as described in Section 3.2, Contributor shall promptly modify the LEGAL file in all copies Contributor makes available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

**(b) Contributor APIs.**

If Contributor's Modifications include an application programming interface and Contributor has knowledge of patent licenses which are reasonably necessary to implement that API, Contributor must also include this information in the LEGAL file.

**(c) Representations.**

Contributor represents that, except as disclosed pursuant to Section 3.4(a) above, Contributor believes that Contributor's Modifications are Contributor's original creation(s) and/or Contributor has sufficient rights to grant the rights conveyed by this License.

**3.5. Required Notices.**

You must duplicate the notice in **Exhibit A** in each file of the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice. If You created one or more Modification(s) You may add your name as a Contributor to the notice described in **Exhibit A**. You must also duplicate this License in any documentation for the Source Code where You describe recipients' rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

**3.6. Distribution of Executable Versions.**

You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code, and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the obligations of Section 3.2. The notice must be conspicuously included in any notice in an Executable version, related documentation or collateral in which You describe recipients' rights relating to the Covered Code. You may distribute the Executable

version of Covered Code or ownership rights under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

### **3.7. Larger Works.**

You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

## **4. Inability to Comply Due to Statute or Regulation.**

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

## **5. Application of this License.**

This License applies to code to which the Initial Developer has attached the notice in **Exhibit A** and to related Covered Code.

## **6. Versions of the License.**

### **6.1. New Versions.**

Netscape Communications Corporation ("Netscape") may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

### **6.2. Effect of New Versions.**

Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License published by Netscape. No one other than Netscape has the right to modify the terms applicable to Covered Code created under this License.

### **6.3. Derivative Works.**

If You create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), You must (a) rename Your license so that the phrases "Mozilla", "MOZILLAPL", "MOZPL", "Netscape", "MPL", "NPL" or any confusingly similar phrase do not appear in your license (except to note that your license differs from this License) and (b) otherwise make it clear that Your version of the license contains terms which differ

from the Mozilla Public License and Netscape Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in **Exhibit A** shall not of themselves be deemed to be modifications of this License.)

## **7. DISCLAIMER OF WARRANTY.**

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

## **8. TERMINATION.**

**8.1.** This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

**8.2.** If You initiate litigation by asserting a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You file such action is referred to as "Participant") alleging that:

**(a)** such Participant's Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You either: (i) agree in writing to pay Participant a mutually agreeable reasonable royalty for Your past and future use of Modifications made by such Participant, or (ii) withdraw Your litigation claim with respect to the Contributor Version against such Participant. If within 60 days of notice, a reasonable royalty and payment arrangement are not mutually agreed upon in writing by the parties or the litigation claim is not withdrawn, the rights granted by Participant to You under Sections 2.1 and/or 2.2 automatically terminate at the expiration of the 60 day notice period specified above.

**(b)** any software, hardware, or device, other than such Participant's Contributor Version, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.1(b) and 2.2(b) are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant.

**8.3.** If You assert a patent infringement claim against Participant alleging that such Participant's Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

**8.4.** In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

## **9. LIMITATION OF LIABILITY.**

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

## **10. U.S. GOVERNMENT END USERS.**

The Covered Code is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

## **11. MISCELLANEOUS.**

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions. With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in the United States of America, any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California, with venue lying in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which

provides that the language of a contract shall be construed against the drafter shall not apply to this License.

**12. RESPONSIBILITY FOR CLAIMS.**

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

**13. MULTIPLE-LICENSED CODE.**

Initial Developer may designate portions of the Covered Code as "Multiple-Licensed". "Multiple-Licensed" means that the Initial Developer permits you to utilize portions of the Covered Code under Your choice of the MPL or the alternative licenses, if any, specified by the Initial Developer in the file described in Exhibit A.

**EXHIBIT A -Mozilla Public License.**

``The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.mozilla.org/MPL/>

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is \_\_\_\_\_.

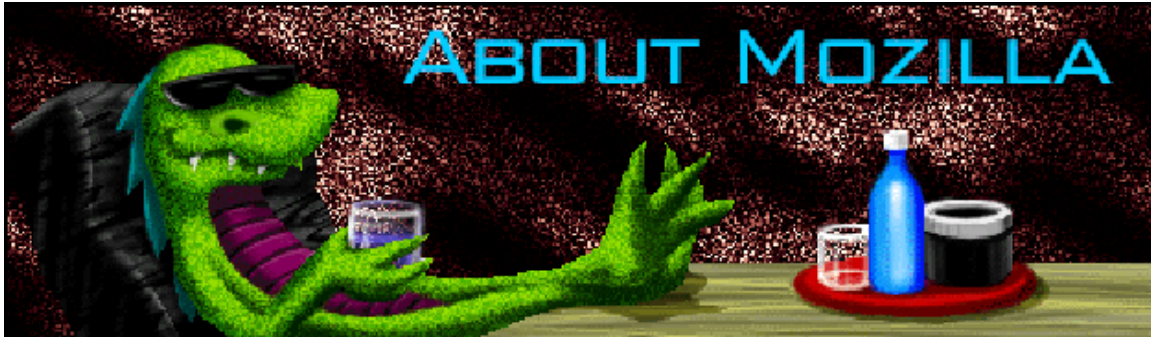
The Initial Developer of the Original Code is \_\_\_\_\_. Portions created by \_\_\_\_\_ are Copyright (C) \_\_\_\_\_. All Rights Reserved.

Contributor(s): \_\_\_\_\_.

Alternatively, the contents of this file may be used under the terms of the \_\_\_\_\_ license (the "[ ] License"), in which case the provisions of [ ] License are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the [ ] License and not to allow others to use your version of this file under the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the [ ] License. If you do not delete the provisions above, a recipient may use your version of this file under either the MPL or the [ ] License."

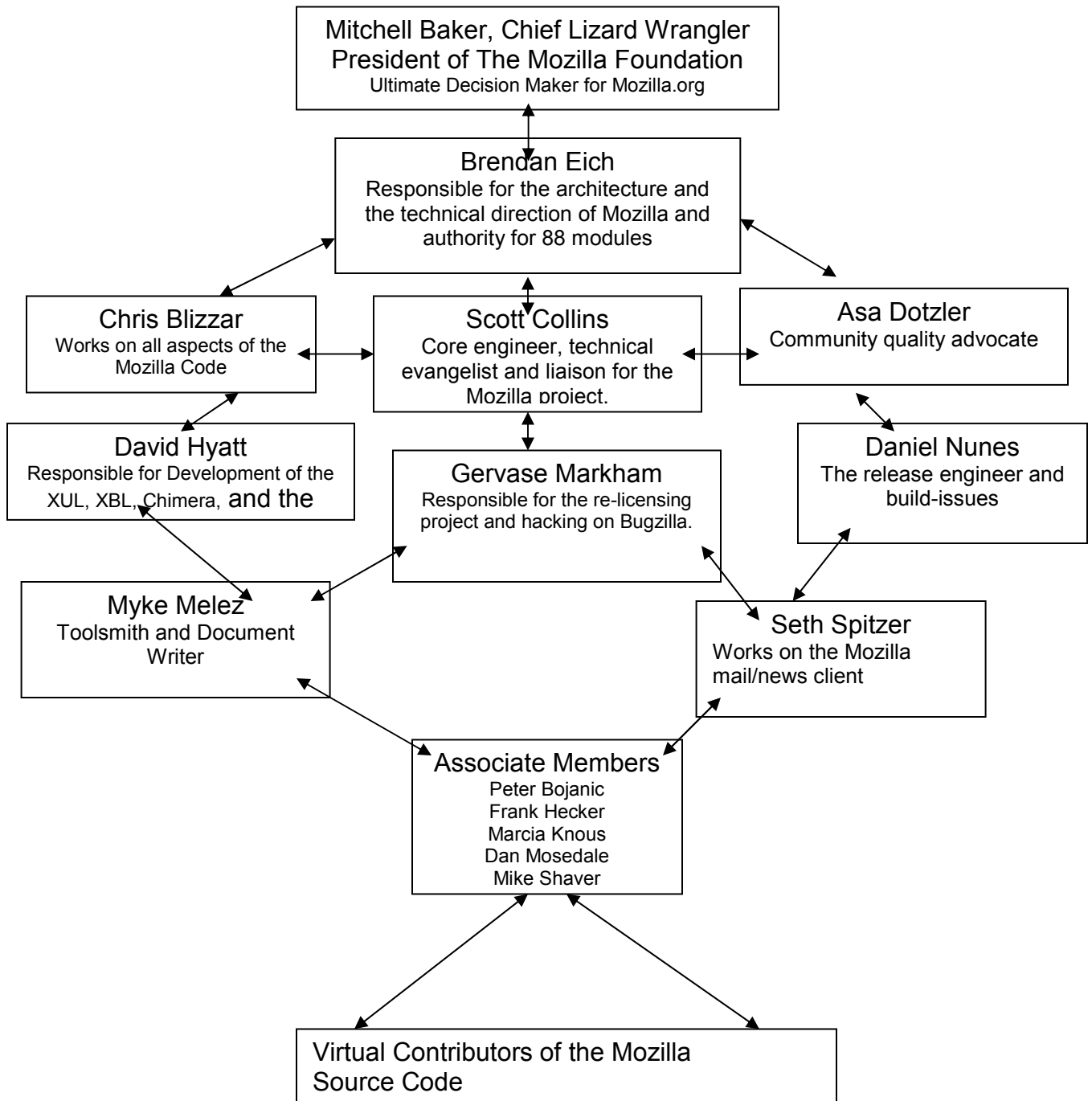
[NOTE: The text of this Exhibit A may differ slightly from the text of the notices in the Source Code files of the Original Code. You should use the text of this Exhibit A rather than the text found in the Original Code Source Code for Your Modifications.]

**Appendix 4**  
(Mozh, 2003)  
**Netscape's Engineering Mascot The Mozilla Lizard**



**Appendix 5**  
(Mozilla.org, 2003g)

**The Virtual Community of The Mozilla Foundation**



## Appendix 6

(Eich & Hyatt, 2003)

### Current Mozilla Development Roadmap

Welcome to the Mozilla development roadmap. This is the third major roadmap revision since the [original roadmap](#) that set Mozilla on a new course for standards compliance, modularity, and portability in 1998. The [previous roadmap](#) documented milestones and rules of development through Mozilla 1.3, and contains links to older roadmaps.

We have come a long way. We have achieved a [Mozilla 1.0](#) milestone that satisfies the criteria put forth in the Mozilla 1.0 [manifesto](#), giving the community and the wider world a high-quality release, and a stable branch for conservative development and derivative product releases. See the [Mozilla Hall of Fame](#) for a list of Mozilla-based projects and products that benefited from 1.0.

Since 1.0, on the main-line "trunk" development path continuing up to 1.3, we have improved on the footprint and performance work begun during the 0.9.x milestones, fixing many longstanding flaws in the standards-compliant, modular, and portable codebase begun in 1998. We've seen the emergence of great features such as tabbed browsing, popup blocking, and <http://paulgraham.com>-inspired Bayesian spam filtering, along with faster, more focused browser projects such as [Mozilla Firebird](#) (formerly Phoenix) and [Camino](#) (formerly Chimera).

#### a new roadmap

But incremental development of the kind we've had since 1.0 is not enough for a healthy open source project. It's clear to us that Mozilla needs a new roadmap, one that charts a path to an even better future. Below we will propose a new application architecture based on the [Gecko](#) Runtime Environment ([GRE](#)), which can be shared between separate application processes. Before discussing the rationales and trade-offs, here are the implications and key elements:

1. Switch Mozilla's default browser component from the [XPFE](#)-based Navigator to the standalone [Mozilla Firebird](#) browser. **Note: the standalone browser's user interface is defined entirely using XUL. So in preferring it, we are not deprecating XUL. We are demonstrating how XUL is a sound basis for fast, cross-platform applications such as Mozilla Firebird.**
2. Develop further the standalone mail companion application to Mozilla Firebird based on the new XUL toolkit used by Firebird, codenamed [Mozilla Thunderbird](#). **Note: the new toolkit is a compatible reimplementaion of the XPFE toolkit, with added features such as customizable toolbars. We are not starting a new C++ GUI toolkit, we are simply moving to the next generation XUL toolkit.**
3. Deliver a Mozilla 1.4 milestone that can replace the 1.0 branch as the stable development path, then move on to make riskier changes during 1.5 and 1.6. The major changes after 1.4 involve switching to Mozilla Firebird and Thunderbird, and working aggressively on the next two items.
4. Fix crucial [Gecko](#) layout architecture bugs, paving the way for a more maintainable, performant, and extensible future.

5. Continue the move away from an ownership model involving a large cloud of hackers with unlimited CVS access, to a model, more common in the open source world, of vigorously defended modules with strong leadership and clear delegation, *a la* NSPR, JavaScript, Gecko in recent major milestones, and Phoenix.

The reasoning behind these new roadmap elements comes down to preferring quality over quantity. We must do less, but better, and with sound extension mechanisms, so that (for example) the community does not fight over user interface pigeon-holes such as the main menu items.

Another, non-UI, extensibility example: Gecko needs to support emerging XML and related standards, some experimentally and conditionally, without everyone having to hack into [nsCSSFrameConstructor.cpp](#) (and that enormous file should be eliminated or greatly simplified by incremental change, during 1.5, 1.6, and probably beyond). It should be possible, using good primitives (HTML and XUL widgets, SVG) and styling and composition tools (CSS, XBL) to avoid an explosion of new C++ code and rendering object bloat.

## **discussion**

When Netscape released its original browser codebase as open source on March 31, 1998, the expectation, product plans, checkin rights, and indeed, code structure, all militated toward a continuation of the big suite of applications (browser, mail, editor) first heavily promoted as "Communicator" in the Netscape 4 days. This "swiss army knife" approach continued when we reset Mozilla around Gecko, XPFE, and [scriptable XPCOM interfaces](#) in late 1998. Along the way, of course, Mozilla has received very generous support from Netscape, mainly in the form of paid contributors and infrastructure.

The result has been a rich and complex application suite with a large set of back-end modules for the user-interface front ends. Many very good things have come from this ambitious integration, including [XUL](#) and [XBL](#).

Yet, if the goal were merely to "rewrite the browser", XUL would have been a false economy. True, as [intended](#), it allowed us to write a cross-platform application front end once, instead of writing native-OS-toolkit-based front ends for at least three platforms. But we ended up spending at least as many people and as much time on the various applications in the suite, and on integrating those application components, as we would have spent developing native browser-only front ends and one browser back end.

This critique supposes that Mozilla and major contributors such as Netscape might have been content to develop only a browser, and not a mail user agent, news reader, HTML editor, and application framework. Even in hindsight, it seems unlikely that "just a browser" would have sufficed. Beyond the worthy cross-platform mail and composer applications it enabled, XUL has been a huge win (a "true economy") for customizers, localizers, distributors, and portable application developers. Without making it our primary focus, we've developed a fairly high quality, web-oriented cross-platform application framework: Mozilla-the-platform.

Nevertheless, ignoring the valuable user-interface itch-scratching, and considering only the minimal set of goals for each application in Mozilla-the-application-suite, the cost of

integration has been high. Unintegrated applications tend to be faster to load, smaller on average in dynamic memory consumption, and more robust and crashproof.

Another example of the high cost of app-suite integration is the inherently overloaded and complicated user interface (just one example out of too many: the *File / New* sub-menu). The target audience of the suite was never clear, and seemed to shift back and forth with prevailing business- and voluntary-contributor-driven winds. [Hyatt's blog](#) is an effective summary of the case against this approach. Simply put: great applications cannot be managed as common land, with whoever is most motivated in a particular area, or just the last to check in, determining the piecemeal look and feel of the application.

Gecko also suffered from over-reach. Not because too many integrated applications were built on top of it -- those helped shake out many design and implementation bugs -- but because it was naively quite "modular" without actually being easy to extend. At the same time, parts of Gecko are still [baroque](#) due to early design limitations and an accumulation of code to patch around core problems.

### summary rationale

In short, and in the same order as the roadmap element list above, the reasons for this new plan are:

1. Mozilla Firebird is simply smaller, faster, and [better](#) -- especially better *not* because it has every conflicting feature wanted by each segment of the Mozilla community, but because it has a strong "add-on" extension mechanism. We recognize that different users need many different features; such demand is legitimate on its face. Attempting to "hardwire" all these features to the integrated application suite is not legitimate; it's neither technically nor socially scalable.
2. What's good for the browser is good for the mail application, too. Mozilla's integrated mail has many fine features, but it suffers from too many integration points with the other apps, and it remains a complicated front end maintained by too few people, most of whom have different day jobs now.
3. The 1.0 branch is almost a year old. It's time to move from 1.0 to 1.4 for mozilla.org-blessed stable development and product releases, to get all the stability, performance, and security fixes made on the trunk since 1.0 into the hands of distributors and users. Many distributors have plans to make this migration. This migration frees the trunk to make more aggressive changes during 1.5 and 1.6, but still with the incremental daily build discipline, and the quarterly alpha/beta/final milestone testing feedback loops.
4. Gecko stalwarts are leading an effort to fix those layout architecture bugs and design flaws that cannot be treated by patching symptoms. Those bugs stand in the way of major improvements in maintainability, footprint, performance, and extensibility. Just by reducing source code complexity, Gecko stands to become much easier to maintain, faster, and about as small in dynamic footprint, yet significantly smaller in code footprint.
5. The *faux*-egalitarian model of CVS access and pan-tree hacking that evolved from the earliest days of Mozilla is coming to an end. Many of the original hackers have moved on, leaving unowned and under-owned modules behind. The combination of over-reach, turnover, and legacy CVS access grants has led mozilla.org to institute [code](#)

[review requirements](#) beyond those required by the relevant module owner (if there is an owner).

### **about ownership...**

The last point is controversial. Let's dwell on it for a moment, and try to clarify it by exclusion.

- [Super-review](#) is generally a fine thing, and two levels of review, one from a domain expert and another from a strong generalist, often pay off for scary changes (big or small). We are not discouraging it where it does pay.
- We do not mean to disparage the valuable pan-tree cleanup and build system hacking done often by only a few dedicated volunteers -- that kind of work should continue.
- Nor do we mean to insist that owners are always right, or that they have "tenure" and cannot be replaced.

Having knocked down all those straw men, the important point that remains standing is this: it is almost always better to have a competent owner who rules decisively, than to have no owner and live in a state of indecision (N.B.: a committee of more than one or two is not an effective owner). This point is especially true for top-down application design and policy setting, particularly for user-interface design. For coherent UI within an application, there is no substitute for leadership by an "application czar". For cross-application consistency where it is needed, we expect such czars to communicate, cooperate, and consolidate things such as common default keybindings.

It is time for Mozilla to "return to normalcy": great software is originated by one or a few hackers building up and leading a larger team of people who test, clean up, extend, and grow to join or replace the first few. Code review, like testing, is an auditing procedure that cannot make excellent code from mediocre input.

Therefore we will promote strong ownership by relieving vigilant owners, on a case-by-case basis, of mandatory super-review requirements, for modules that the super-reviewers deem sufficiently well-owned. And where ownership is weak or missing, we will take steps to find an owner, or absent any candidates, to reduce our dependencies on the under-owned or unowned code. If we can "limp" along for a while without an owner for some crucial module, we will do so -- but experience has shown that almost always leads to a much more serious condition than a limp.

### **what all this does not mean**

In the same vein of clarifying by counter-example, here is a list of more things that we are *not* proposing:

- We are not trying to stop volunteers and companies from developing the XPFE-based browser. Several companies have shipped and will ship products based on this venerable component of the application suite, and on the entire suite. However, we intend to focus on switching to separate applications based on the new [application architecture](#). Therefore it is quite possible that the XPFE-based browser may bit-rot fairly quickly, so that the 1.4 branch contains its only working form. If enough

contributors sign up to keep the XPFE-based browser working, mozilla.org will consider supporting that browser on the trunk beyond 1.4. However, it's not clear that we will have all the tinderbox and other resources needed to keep two different toolkit-based browser applications well-tested. **We request that those companies and organizations shipping products based on Mozilla-the-application-suite let us know their plans in light of this new proposed roadmap.**

- We are **not** deprecating XUL in favor of front ends based on native GUI toolkits. Nor are we deprecating Camino, Mozilla's Gecko-based browser that has a native [OS X](#) front end. Both approaches have their wins, and their loyal fans. The Mozilla community has embraced both approaches (even producing a version of Phoenix for OS X that tests well).
- Therefore, in switching browsers, we are not dropping XUL on the Mac. We aim to ensure that Mozilla's cross-platform applications and toolkit remain both cross-platform and viable as applications that people actually use. And we need the same kind of embedded Gecko test coverage on the Mac that we get on other platforms. So, when we switch the default-built browser to Mozilla Firebird, we will provide daily and milestone builds of it for OS X.
- Several crucial tools integrated with the XPFE-based browser, the [DOM inspector](#) and [Venkman](#) (the JavaScript Debugger), must be supported in the new, standalone browser, as add-ons.
- The other integrated components of the Mozilla application suite, [Calendar](#), [Chatzilla](#), and [Composer](#) (the HTML editor application), are not going away, either. We're not sure yet how they'll evolve -- whether they'll become standalone toolkit applications (and if so, based on which XUL toolkit), or popular add-ons to Mozilla Firebird (if so, they will need to use its new XUL toolkit). But we're committed to supporting them to the fullest extent required by their owners, including providing daily and milestone builds of them for community testing and feedback.

## application architecture

Let's begin the new application architecture proposal by recapitulating the relevant facts and defining some terms.

There are currently two major classes of applications being built using Mozilla's technology. The first class of applications are *embedding* applications. They use Mozilla's layout engine, but write their own code for user interface features that exists outside of the laid-out HTML or XML document view. The second class of applications are *toolkit* applications. They are built on top of Mozilla itself and designed to be cross-platform. The user interface elements are defined in XUL and rendered by Gecko itself.

Both classes of applications will be able to make use of the Gecko Runtime Environment ([GRE](#)) to enable the sharing of a single installation of Gecko. Applications may even share profiles, although the inter-process communication work to support sharing profiles among applications running in separate processes is not done yet (as of 1.4alpha).

Toolkit applications will also support extensions, specially designed add-ons that can be layered on top of the core application to provide additional functionality. In the case of the

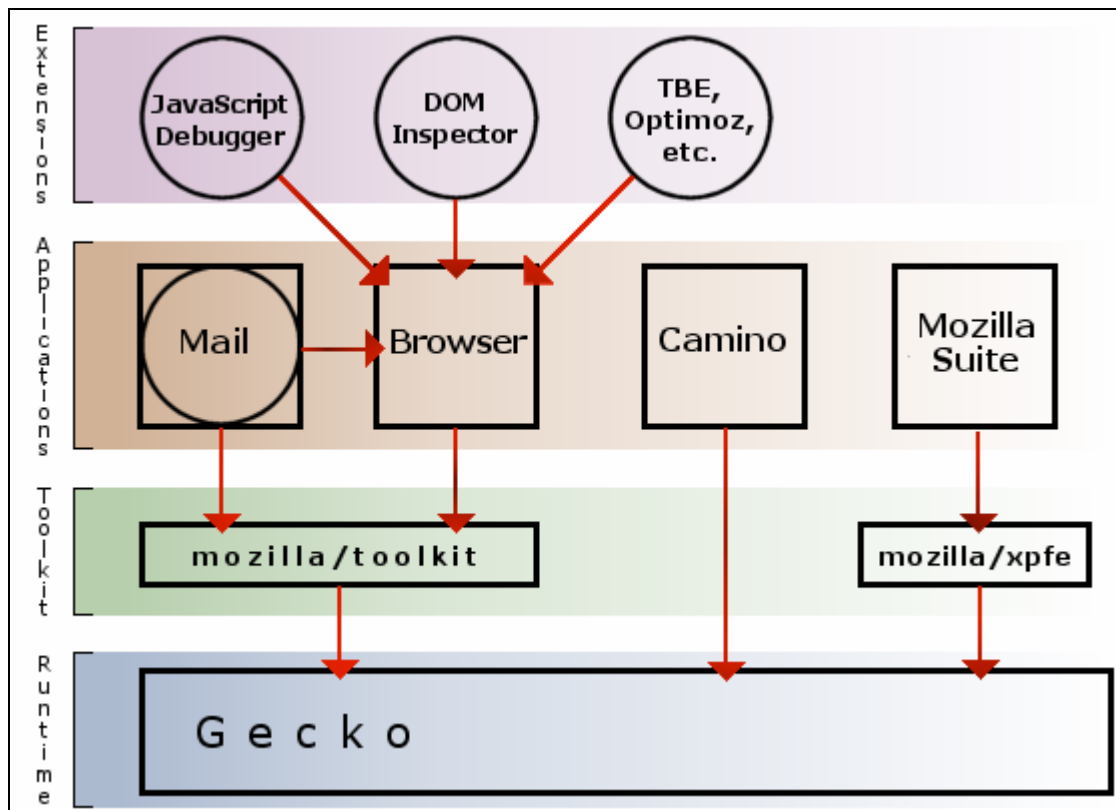
browser application, examples of such add-ons include mouse gestures, the site navigation toolbar, the DOM inspector, and the JavaScript debugger.

In addition, we propose that certain toolkit applications should themselves be extensions, meaning that they can be built both as standalone applications and as add-ons installed into other applications. An example of such an application is Thunderbird, a.k.a. Mozilla Mail, which will be capable of either running as a standalone application or being installed directly into Mozilla Firebird, a.k.a. the Mozilla Browser, as an extension.

(We expect many fans of the current, integrated browser/mail application to demand such an add-on, and we will work to provide it before switching the default-built browser.)

The idea is to move from the over-integrated application suite to simpler toolkit applications, to remove more advanced functionality from the default configurations, but to provide robust tools for building your own browser by layering those extensions that you want to use on top of the base. In an attempt to avoid an explosion of unique builds that have to be supported by mozilla.org, we will likely ship with all of the popular extensions installed but disabled, so that they can be easily turned on by those who wish to use them, and uninstalled by those who don't.

A picture should make this architecture clear.



This picture shows Mail as both an Application (inside a square) and an Extension (the circle with an arrow pointing to the Browser application that Mail extends). The pure extensions, which are never applications as well, are shown at the top.

## to do

This roadmap is a proposal. We are pointing in a direction toward which we think the Mozilla project should move. To get from where we are today (1.4alpha) to that better place, at least the following things need to be done:

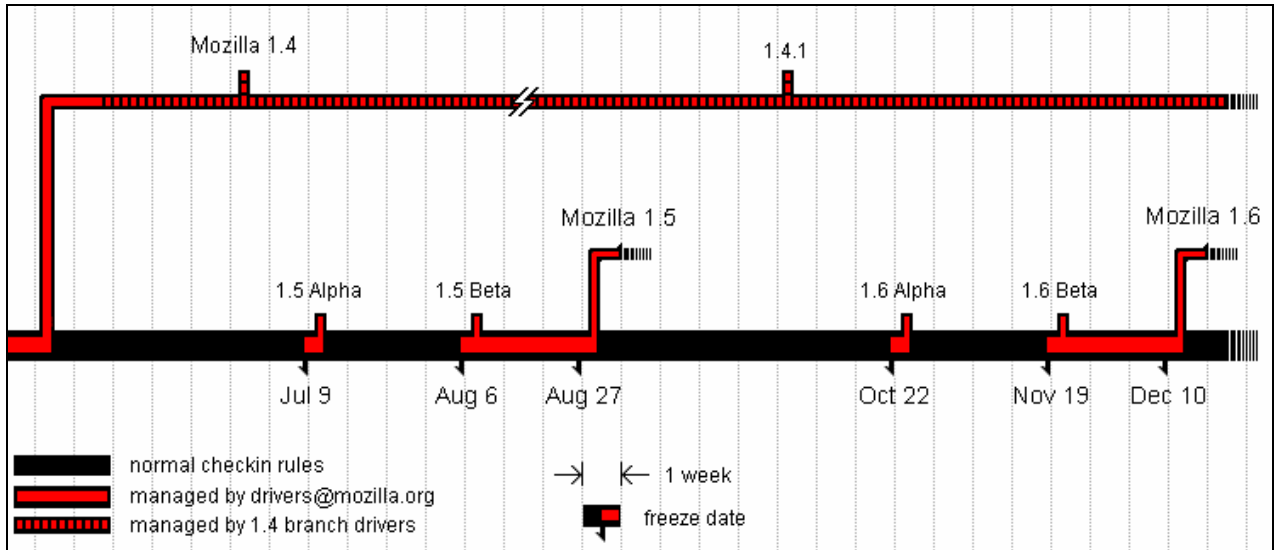
- Eliminate the [MOZ\\_PHOENIX](#) #ifdefs added to files built by both browsers.
- Un-fork any files forked by Phoenix, where the XPFE side of the fork has since seen significant bugfixes not made to the Phoenix side. Whether the solution is to move such files to a shared location, or to merge fixes but continue maintaining both "tines" of the fork, or to deprecate and obsolete the XPFE tine, we can't say in general. But something needs to be done, and soon (fixes will continue to be made on the XPFE tine during 1.4).
- Ensure that small build options (low-level JS debugger support, MathML, rich text editing) that may have been excised from Phoenix previously are configured into the default build.
- Create or procure existing add-ons for:
  - DOM inspector
  - Venkman
  - The site navigation toolbar
  - Mail integration (<ctrl-M> for message compose, etc.)
- Develop Thunderbird based on the new XUL toolkit.
- Provide enough tinderbox machines to cover both old and new browser builds, during the transition period.

Again, we're only pointing the way here. The detailed plan of attack should be developed in the newsgroups and via Bugzilla. It's clear now that we will not be able to switch to Mozilla Firebird by the Mozilla 1.5 final milestone. Instead, we expect Mozilla 1.5 to coincide with Mozilla Firebird 0.7. But we intend to implement the new application architecture in the next several milestones, till most of the community is won over to the new apps. We invite your comments, preferably in the [mozilla.seamonkey](#) newsgroup.

## milestone schedule

As the previous roadmap [documented](#), the Mozilla project has been following a quarterly milestone plan that emphasizes regular delivery of stable new-feature releases, ideally with risky changes pushed into an "alpha" minor milestone, followed by stabilization during a "beta" period, then a shorter freeze during which only "stop-ship" bugs are found and fixed.

Many in the community have asked for a longer alpha period. But with too long an alpha, we fail to collect *and act on* feedback including automatic crash reports from a sufficiently large group of testers. Still, we're convinced that a longer alpha than beta makes sense, so we have moved one week from beta to alpha. Here is the updated milestone schedule:



Tabulating the milestones to show the proposed dates, with trunk freeze, branch creation, and milestone release dates distinguished from one another (the next milestone's start date is the previous one's branch date), yields:

milestone	start	freeze	branch	ideal release	actual release
1.4	25-Apr-2003	14-May-2003	16-May-2003	21-May-2003	30-Jun-2003
1.5alpha	16-May-2003	09-Jul-2003	11-Jul-2003	11-Jul-2003	22-Jul-2003
1.5beta	11-Jul-2003	06-Aug-2003	n/a	08-Aug-2003	27-Aug-2003
1.5	08-Aug-2003	27-Aug-2003	29-Aug-2003	03-Sep-2003	15-Oct-2003
1.6alpha	10-Sep-2003	22-Oct-2003	n/a	24-Oct-2003	?
1.6beta	24-Oct-2003	19-Nov-2003	n/a	21-Nov-2003	?
1.6	21-Nov-2003	10-Dec-2003	12-Dec-2003	19-Dec-2003	?

As always, the milestone freeze time is 11:59 P.M. Pacific Time (see the [tinderbox](#) page for notices) on a Tuesday.

If you are planning a Mozilla-based product release whose schedule does not jibe well with the above milestones, we welcome your [feedback](#) (we will keep confidential information to ourselves, and will take appropriate safeguards as necessary).

### how you can help

**C and C++ hackers:** tinderbox now measures [code footprint](#), so let's start working to reduce it rather than increase it. Resist the all-too-common tendency to add more code. Try to remove code, simplify over-complicated code, undo premature optimizations, and purge gratuitous use of XPCOM and its not-quite-XPCOM precursors. If you have to add a new feature, make sure it's bundled in the right library, which may mean adding a new library. **All additions to modules linked into the [minimal embedding browser builds](#) must be approved by [drivers](#).**

**Mozilla embedders:** we need your input to Bugzilla, marking bugs with `embed`, `footprint`, `mlk`, `perf`, and other relevant keywords. Use Bugzilla's new request-tracking capabilities to

nominate bugs as blocking upcoming milestones (e.g., `blocking1.4b?`), and please comment in the bug with a convincing argument for why you require the fix by that milestone. [Mozilla project management](#) will help ensure that bugs are assigned to hackers who can target their fixes so as to satisfy as many milestone-keyword nominations as possible.

**Bug assignees**, and especially helpers who can offload `Futured` or untargeted bugs from their nominal assignees and fix those bugs: please make a pass at targeting your assigned bugs across the `mozilla1.5` and `mozilla1.6` milestones, using the criteria application and layout re-architecture for 1.5 and 1.6 as your guides. Please try to offload bugs to helpers, pinging [drivers@mozilla.org](mailto:drivers@mozilla.org) if you cannot find anyone to help fix `futured` or untargeted bugs that you believe should be fixed soon.

**Community members**: please use Bugzilla's milestone nomination feature wisely, and do not change the Target Milestone of anyone else's bug without the assignee's consent. You may of course vote for bugs also to help inform prioritization (but remember that patching beats voting). Finally, please help keep advocacy and "me-too" comments out of the bug system.

## project management

To drive developers looking to help toward bugs needing assistance in a timely fashion, to moderate risk, and to aid commercial projects based on Mozilla in managing their product releases, mozilla.org has created a group of project managers, [drivers@mozilla.org](mailto:drivers@mozilla.org).

The current drivers are:

- David Baron, [dbaron@dbaron.org](mailto:dbaron@dbaron.org)
- Chris Blizzard, [blizzard@redhat.com](mailto:blizzard@redhat.com)
- Scott Collins, [scc@mozilla.org](mailto:scc@mozilla.org)
- Asa Dotzler, [asa@mozilla.org](mailto:asa@mozilla.org)
- Brendan Eich, [brendan@mozilla.org](mailto:brendan@mozilla.org)
- Chris Hofmann, [chofmann@mozilla.org](mailto:chofmann@mozilla.org)
- Randell Jesup, [rjesup@wgate.com](mailto:rjesup@wgate.com)
- Michael Kaply, [mkaply@us.ibm.com](mailto:mkaply@us.ibm.com)
- Robert O'Callahan, [roc+moz@cs.cmu.edu](mailto:roc+moz@cs.cmu.edu)
- Tim Rowley, [tor@cs.brown.edu](mailto:tor@cs.brown.edu)
- Mike Shaver, [shaver@mozilla.org](mailto:shaver@mozilla.org)
- Seth Spitzer, [sspitzer@mozilla.org](mailto:sspitzer@mozilla.org)
- Dan Veditz, [dveditz@cruzio.com](mailto:dveditz@cruzio.com)

## Appendix 7

### 2nd Version of Mozilla Development Roadmap (Eich, 2002)

Welcome to the Mozilla development roadmap. This document briefly describes where the Mozilla project has been, and then details where it is going. It proposes key "road rules" and a release schedule for ongoing Mozilla-the-browser source milestone releases, from which anyone can build commercial and other products. It also hints at how Mozilla-the-platform should evolve, again from an operational or "release process" point of view.

#### background

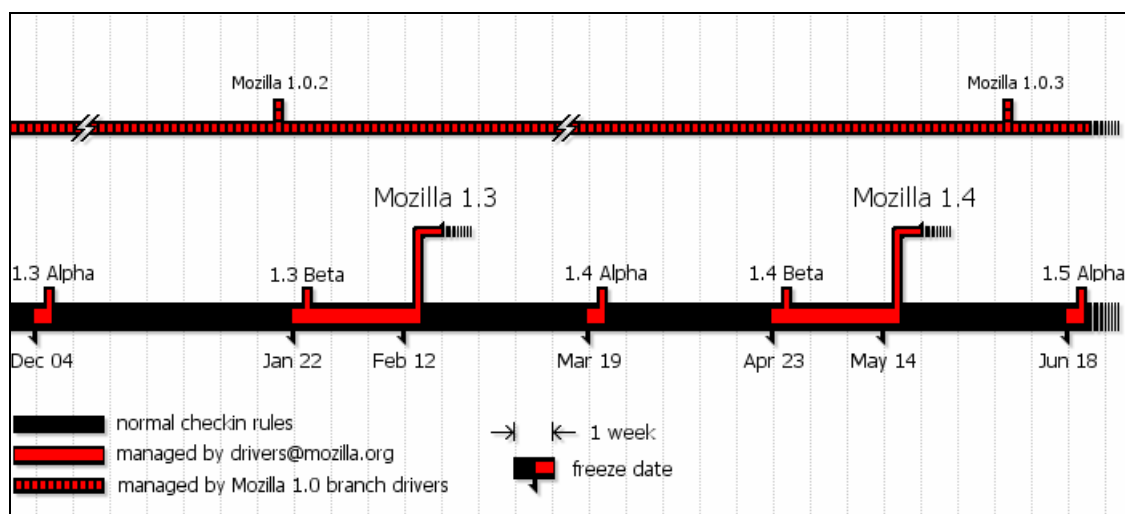
The [original roadmap](#) recorded the momentous decision in October 1998 to reset the Mozilla project around the [new layout engine](#) (now called Gecko), a cross-platform front end ([XPFE](#); now several [XP Apps](#) built on an [XP Toolkit](#)), and a scriptable components architecture ([XPCOM](#) and [XPConnect](#)).

The [previous roadmap](#) charted Mozilla's path through the release of [Netscape 6](#) and beyond, toward the goal of releasing a Mozilla 1.0 milestone. In that update, I wrote "Mozilla needs performance, stability, and correctness" and not any particular new feature. Just before 2001 began, I wrote that useful and relevant (defined by the community) extensions are always welcome, provided that they don't have a high opportunity cost in terms of contributors who otherwise could and would have helped hack on 1.0. But by the fall of 2001, as noted in the Mozilla 1.0 [manifesto](#), the opportunity costs of features and extensions had grown to the point where such "non-1.0" work jeopardized a 1.0 milestone that fit into any achievable schedule.

This roadmap update reflects the milestone schedule for the summer of 2002, now that [Mozilla 1.0](#) has been released.

#### tree management

Here is a depiction of the latest thinking on tree management:



In a departure from the five-week milestone cycle of 2001, each major milestone takes one quarter of a year or thirteen weeks to be delivered, and starts with a five-week "alpha" milestone, during which destabilizing development should be done. This is followed by a five-week "beta" milestone period during which less risky bug fixes, in particular followup and cleanup of the alpha phase work that tend to restabilize the codebase, should be targeted. Finally, a two to three week "done" period, where only driver-approved changes can be checked in, completes the milestone. There is enough slack to allow a floating holiday week per quarter, which we hope will help avoid the "time and people shortage" experienced last December, during 0.9.8.

Each milestone ends with a release to gather automatic crash reports ("talkback"). We believe that this feedback must be gathered and acted upon at least every five or six weeks, based on our several years' experience. We hope that alpha and beta milestones need no more than a few days' tree closure to prepare for release. As we do not intend to promote alpha and beta releases for uses other than testing, we will ship those releases according to the predetermined schedule, and let the chips fall where they may. Again, the once-per-quarter major release (e.g., 1.1) is the product-worthy branch-point.

Thus we hope to match the rhythm of the Mozilla community, which seems to have resulted in an odd-even pattern of "alpha-ness" and "beta-ness" observed since at least 0.9.1, along with a protracted freeze and branch effort before enough bugs were wrung out to release.

Finally, we are rationalizing our release numbering to count branches, with the leading 1 for the trunk version. Thus the 1.0.1 branch-of-a-branch sketched in the picture above, although at a date yet to be determined. The long-lived 1.0 branch should host conservative development of bug fixes that may also be appropriate for the trunk, per the Mozilla 1.0 manifesto's [rationale](#),

Interested parties should collaborate, with support from [drivers@mozilla.org](mailto:drivers@mozilla.org), in developing conservative fixes for critical bugs in this branch. Anyone who wants a baseline for development that will work with the public APIs of Mozilla 1.0 is free to develop against the 1.0 branch.

Tabulating the milestones to show the proposed dates, with trunk freeze, branch creation, and milestone release dates distinguished from one another (the next milestone's start date is the previous one's branch date), yields:

milestone	start	freeze	branch	ideal release	actual release
1.1.alpha	09-Apr-2002	05-Jun-2002	n/a	07-Jun-2002	11-Jun-2002
1.1.beta	07-Jun-2002	10-Jul-2002	n/a	12-July-2002	22-July-2002
1.1	12-Jul-2002	31-Jul-2002	02-Aug-2002	09-Aug-2002	26-Aug-2002
1.2.alpha	02-August-2002	04-Sept-2002	n/a	06-Sept-2002	11-Sept-2002
1.2.beta	06-Sept-2002	09-Oct-2002	n/a	11-Oct-2002	16-Oct-2002
1.2	11-Oct-2002	30-Oct-2002	01-Nov-2002	08-Nov-2002	26-Nov-2002
1.3.alpha	01-Nov-2002	04-Dec-2002	n/a	06-Dec-2002	13-Dec-2002

1.3beta	06-Dec-2002	22-Jan-2003	n/a	24-Jan-2003	10-Feb-2003
1.3	24-Jan-2003	12-Feb-2003	14-Feb-2003	21-Feb-2003	?
1.4alpha	14-Feb-2003	19-Mar-2003	n/a	21-Mar-2003	?

Traditionally, the milestone freeze time is 11:59 P.M. Pacific Time (see the [tinderbox](#) page for notices) on a Tuesday. During the freeze, only bugs deemed "must-fix-for-this-milestone", typically last minute regressions, are fixed on the trunk, as everyone tries to qualify the daily builds as branch-worthy. Release [staff@mozilla.org](mailto:staff@mozilla.org) create the milestone branch, which receives even more intensive QA over the weekend, leading to a release at the earliest by the following Wednesday.

We have tried, and sometimes failed, to avoid slipping any release, to avoid extending the life of a milestone branch and dividing labor poorly between trunk and branch. In spite of past difficulties, we aim to release by the Wednesday following the freeze, and use the "done" mini-milestone to absorb extra work that might cause an earlier slip. So, any not-quite-showstopper bugs unfixed by Wednesday should be targeted at the next Mozilla milestone and fixed ASAP.

[Drivers](#) have proposed these changes to the previous milestone plan in order to use "even more obvious" milestone numbers, to match the odd/even stability pattern and end-game schedule observed since 0.9.1, and to match calendar seasons and business quarters (with only a week's skew from the vernal equinox to when the 1.1alpha trunk should open). If you are planning a Mozilla-based product release whose schedule does not jibe well with the above milestones, we welcome your [feedback](#) (we will keep confidential information to ourselves, and will sign NDAs if necessary).

### how you can help

**Mozilla embedders:** we need your input to Bugzilla, marking bugs with `embed`, `footprint`, `mlk`, `perf`, `mozilla1.0.1`, `mozilla1.1`, and other relevant keywords. [Mozilla project management](#) will help ensure that bugs are assigned to hackers who can target their fixes so as to satisfy as many milestone-keyword nominations as possible.

**Bug assignees,** and especially helpers who can offload `Futured` or untargeted bugs from their nominal assignees and fix those bugs: please make a pass at targeting your assigned bugs across the `mozilla1.1` and `mozilla1.2` milestones, using the criteria in the Mozilla 1.0 manifesto. Please try to offload bugs to helpers, pinging [drivers@mozilla.org](mailto:drivers@mozilla.org) if you cannot find anyone to help fix `futured` or untargeted bugs that you believe should be fixed soon.

**Community members:** please use the `mozilla1.x` milestone-keyword nomination system wisely, and do not change the Target Milestone of anyone else's bug without the assignee's consent. The keyword-proposal/target-milestone-disposal system will continue beyond 1.0 to help inform prioritization by bug assignees. You may of course vote for bugs also to help inform prioritization (but remember that patching beats voting; also, please keep advocacy out of the bug system).

### code review

mozilla.org is continuing to require [code review](#) across the board to approve checkins to the bulk of the Mozilla codebase. The Mozilla Code Reviewers document [distinguishes](#) areas of code that require this so-called "super-review" from those hosted on cvs.mozilla.org that have other policies set by module owners. All code hosted by mozilla.org requires active ownership, and therefore module owner review of changes before a contributor can type 'cvs commit'.

Design review obviously should precede code review! Use the [porkjockeys](#) mailing list (which has a [news gateway](#)) to raise design questions and issues.

## **project management**

To drive developers looking to help toward bugs needing assistance in a timely fashion, to moderate risk, and to aid commercial projects based on Mozilla in managing their product releases, mozilla.org has created a group of project managers, [drivers@mozilla.org](mailto:drivers@mozilla.org).

The current drivers are:

- David Baron, [dbaron@dbaron.org](mailto:dbaron@dbaron.org)
- Chris Blizzard, [blizzard@redhat.com](mailto:blizzard@redhat.com)
- Peter Bojanic, [peterb@oeone.com](mailto:peterb@oeone.com)
- Scott Collins, [scc@mozilla.org](mailto:scc@mozilla.org)
- Asa Dotzler, [asa@mozilla.org](mailto:asa@mozilla.org)
- Brendan Eich, [brendan@mozilla.org](mailto:brendan@mozilla.org)
- Chris Hofmann, [chofmann@netscape.com](mailto:chofmann@netscape.com)
- Randell Jesup, [rjesup@wgate.com](mailto:rjesup@wgate.com)
- Robert O'Callahan, [roc+moz@cs.cmu.edu](mailto:roc+moz@cs.cmu.edu)
- Tim Rowley, [tor@cs.brown.edu](mailto:tor@cs.brown.edu)
- Mike Shaver, [shaver@mozilla.org](mailto:shaver@mozilla.org)
- Jud Valeski, [valeski@netscape.com](mailto:valeski@netscape.com)
- Dan Veditz, [dveditz@cruzio.com](mailto:dveditz@cruzio.com)

I hope, in the not too distant future, to update the roadmap with some thoughts on the future of the Mozilla project after 1.0. Clearly, bug fixes, architectural changes, and new code waved off from mozilla1.0 will want to land during 1.1alpha. Mozilla also needs to support the "Mozilla-the-platform" side of the coin for more applications than the default-built browser/mailnews/editor/chatzilla suite. I hope too that Mozilla technologies and innovators can make significant qualitative improvements (e.g., via something like [Intertwingle](#)) in how we all browse, organize, and cope with information on the web, in our file systems, and locked into application data formats, wherever such data may live. There is life after 1.0 -- stay tuned!

## Appendix 8

### Original development roadmap (Eich, 1998)

Welcome to the Mozilla browser development roadmap. For up-to-the-minute answers to frequently-asked questions, click [here](#).

In my experience, the best maps contain historical notes and sightseeing tips. A software roadmap in particular must keep integrating the present as it passes by, or it soon becomes useless. That kind of map records design decisions and their rationales. More important, such a map reflects unforeseen terrain and unexpected inventions as they emerge, rather than starting from some (unrealistic) fixed architecture and proceeding through mundane implementation.

#### brief history

It has been over five months since I posted to [mozilla.general](#) my [then-current thoughts](#) on the Mozilla browser development schedule. That schedule listed only one large feature, mail/news integration, and put off enumerating the rest until "later, [...] barring a better idea."

Since then, a lot of great work has been done by all mozilla.org developers, fixing warnings and bugs, porting and cleaning up code, improving performance, refining downloadable chrome, setting up the autoconf build system, etc. But the mail/news code from Netscape never arrived. The features most demanded by content developers (incremental layout with reflow, correct CSS, a Level-1 DOM) have been hard to implement on the old codebase. And developers who have wanted to "just add a hook", ala [Emacs](#) and its extension languages, have found the old layout and FE code less hospitable than they would like.

Well, now is later, and we do have a better idea or three.

#### where we're going

It's time to stop banging our heads on the old layout and FE codebase. We've pulled more useful miles out of those vehicles than anyone rightly expected. We now have a great [new layout engine](#) that can view hundreds of top websites. We have a convincing proof-of-concept for an [XPFE](#). Why not use these two new projects in Mozilla, to solve a bunch of longstanding problems? All of the following combined to say that it is time for the Mozilla browser project to use XPFE and NGLayout:

- the judgment of the [module owners](#);
- the fervent wishes of web content authors; and
- my personal judgment as mozilla.org technical bigshot

Not only will many bugs and lacks be resolved, the road ahead can better support a full mail client, user-scriptable hooks, and many other clients and proxy-like modules.

Moving to XPFE and NGLayout means moth-balling all of the old-style FEs, and of course `mozilla/lib/layout` and its friends. Coordinated development will cease right away on the [old layout](#) code, [XFE](#), and all of the other FEs.

Note however that most back-end modules carry forward with XPFE and NGLayout; many already work with NGLayout as-is, while some have been `#ifdef`'ed. While we won't break old layout and FEs intentionally, evolution of all the back-end modules common to old and new layout will tend to rot the old codebase. We've already CVS-tagged the mozilla tree with the label `MozillaSourceClassic_19981026_BASE` to facilitate mothballing and archaeology (there's also a `MozillaSourceClassic_19981026_BRANCH` branch tag for retrograde development).

## re-introduction

This document will serve as a narrative roadmap for mozilla browser development. It won't be a schedule, although schedules should be attempted that refer to the terrain recorded and projected here. It will state rules of the road, or the design principles we'd like to employ, to build on the value of XPFE, NGLayout, and [modularity using XPCOM](#). It will list major items of work. And I'll update it over time to reflect reality.

Nothing (apart from accurate history) in this roadmap should be taken as irrevocable. Although we try to avoid unduly revisiting decisions, we welcome all developer comments and corrections. Let controversy bloom if it needs to. Just give us your evidence and reasons, and we at mozilla.org will correct our course and this map to match the actual terrain.

## design principles

Here are the browser road rules. Some of these points are concrete design decisions more than highfalutin' principles. But we think they reflect sound ideas (for example, autoconf captures feature dependencies better than our old Unix build system's platform macros and consequent `#if defined(linux) || ... tests` in the code).

- External development counts more than convenience or ease-of-habit for internal-to-Netscape developers. The Netscape X-heads, for example, have moved all of their mail usage except for I'm-out-sick-today and any truly-proprietary messages to the [mozilla.unix](#) newsgroup. Likewise with NGLayout hackers and the [mozilla.layout](#) group. So it shall be for all development.
- Fresh mail and news clients will be designed and implemented jointly by Netscape and external developers, using open source methods, and this time with an open source database (which I think is likely to be [Berkeley DB](#)). The new mail/news code will use NGLayout for HTML rendering. It shall use as much XPFE-like cross-platform widgetry as it can; it shall use thin-mail techniques akin to [Ender](#) and [Shack](#).
- As much as possible, UI structure should be implemented using HTML or XML and the NGLayout engine. Style should be expressed using CSS where doing so makes sense, and doesn't require non-standard extensions to CSS. Structure that can be queried, modified, and superposed from remote sources should probably be RDF, expressed in its XML syntax. [Actions](#) should be implemented (or at least, be implementable) via well-defined events and their JavaScript handlers
- Modules will use [XP-IDL and XPCOM](#) to specify and query their programmatic interfaces. That way, all modules are scriptable without requiring all interface creators to write repetitive-yet-tricky script-engine glue by hand. Also, parts of the system may be distributed across thread, process, and machine boundaries using automatically

- generated stubs. This distribution potential is a distant second goal, except where single-threaded code restrictions require us to implement an [apartment](#) threading model.
- Any scripting hooks above and beyond the scriptable-C++ entry points that we get by using XP-IDL must be designed and specified using IDL extensions. The crucial principle here is to make all useful parts of the browser easy to script. What's more, events must be easy to "hook" or handle in a flexible way, say using JavaScript. More on this point will be written in the [XPCOM-Connect document](#).
  - [autoconf](#) will be the one true Unix-hosted build system for the modules you checkout and build *en masse* to make a browser that runs on Unix. Module owners will have to maintain only their module's `Makefile.in` and `configure.in` scripts for Unix, not both of those and `gmake Makefiles`. (This principle does not apply to NSPR, which is a separately-built prerequisite of the browser; it may be that the Berkeley DB follows NSPR into separately-built prerequisite status, eventually.)
  - [X-heads](#) believe the time is right to move from Motif to [GTK+](#), the leading fully-open-source X-based UI toolkit. The general principle here, mentioned above in the mail/news paragraph in connection with Berkeley DB, and of course behind `autoconf` adoption, is that Mozilla must use the best open source solution to solve a well-understood problem.

### major work items

Each of the following tasks constitutes a major milestone. This list isn't even partially ordered, but it should be, to express dependencies. It should be topologically sorted and scheduled, too (later). It shall be refined to include sub-tasks and missing items of work. For now, it's a roughly-ordered brain dump of what I think we'll all be doing in the next few months:

- NGLayout completion. See the [NGLayout Project Page](#) for detailed to-do lists and schedules. In light of the "open source solution" principle, the widget library used by NGLayout needs to be ported from Motif to GTK+.
- The old composer/editor was tightly coupled to old layout. NGLayout has a rich DOM API that perhaps (with more event work) could form the basis for the new editor. In any case, Mozilla must have an HTML composer/editor, and it should be embeddable via an `OBJECT` or `TEXTAREA TYPE="text/html"` tag.
- XPFE design and implementation. There is a very preliminary [set of design documents](#), but they are subject to change and evolution based on the design principles above, and interested developers' input.
- Back-end modules such as RDF and JavaScript are more-or-less ready to go, but their interactions, especially their execution model interactions, need analysis. Also, memory consumption and things like garbage collector scheduling require a "whole system" analysis.
- Mail/news XPFE and modular back-ends need to be designed and implemented, or in a few cases evolved from existing Mozilla code such as `mozilla/lib/libmime`.
- The XP-IDL compiler and IDL specs for all interfaces must be written, tested, and deployed as part of the build system.
- The JavaScript XPCOM-Connect runtime, by which JS can call C++ methods using runtime type information, and convert results to JS types, must be built. The good news is that some of it will resemble LiveConnect. The bad (or good, for hackers who enjoy it) news is that we need some assembly code to implement each platform's `Invoke`

- method, where platform is a pair: (cpu instruction set architecture, compiler-and-cpu-determined calling conventions).
- Conversion of all Unix builds to autoconf. Currently, NGLayout is built by `mozilla/nglayout.mk`. If we follow the existing browser build model, it should be built by the `Makefile` generated from `mozilla/Makefile.in` and `mozilla/configure.in`.
  - For Windows builds, at the least `mozilla/nglayout.mak` and `mozilla/client.mak` need to unify under the latter name. But it seems possible to use autoconf even on Win32, along with cygwin32 tools. Perhaps some ATL-like dependency will prevent it, but it should be given a fair shot.

## References

- Applegate, L. M., Austin, R. D., & McFarlan, W. F. (2002). *Creating Business Advantage in the Information Age*. New York: McGraw-Hill Irwin.
- Asaravala, A. (2003). Surf's Up! There are lots of choices, but which browser is the best? Here is the roundup. *Linux Magazine*.
- Audris, M. (2003). Two Case Studies of Open Source Software Development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology*, 11(3), 309-346.
- Baker, M. (2003a). *Browser Innovation, Gecko and the Mozilla Project*, [Website]. Mozilla.org. Available: <http://www.mozilla.org/browser-innovation.html> [December 8, 2003].
- Baker, M. (2003b). Meeting and conversation with Mitchell Baker. San Francisco.
- Bowles, J. (2003, October 25, 2003). eLogistics: Delivering the Promise of the Internet Economy. *Forbes*.
- Cervone, F. (2003). The Open Source Option. *Library Journal*, 8.
- Cooper, D. (2003). Meeting and Conversation with Danese Cooper. San Francisco.
- Cusumano, M. A., & Yoffie, D. B. (1998). *Competing on Internet Time*. NY: Free Press.
- DecisionOne. (2003). *DecisionOne: Who We Are*. Available: <http://www.decisiononecorporate.com/whoweare.htm> [November 1, 2003].
- Dotzler, A. (2003). *Getting Involved with Mozilla*. The Mozilla.org. Available: <http://mozilla.org/events/lugod/October 21, 2003>].
- Edwards, G. (2003). Email Correspondence with Gary Edwards.
- Eich, B. (1998). *Development Roadmap: Original*, [Website]. Mozilla.org. Available: <http://www.mozilla.org/roadmap/roadmap-26-Oct-1998.html> [December 3, 2003].
- Eich, B. (2002). *Mozilla Development Roadmap: Previous*, [Website]. Mozilla.org. Available: <http://www.mozilla.org/roadmap/roadmap-05-Jun-2002.html> [December 3, 2003].
- Eich, B., & Hyatt, D. (2003). *A New Roadmap*, [Website]. Available: <http://www.mozilla.org/roadmap.html> [October 7, 2003].
- FSF. (1991). *GNU General Public License*. Free Software Foundation. Available: <http://www.gnu.org/licenses/gpl.txt> [December 3, 2003].
- FSF. (2003). *Overview of the GNU Project*. Free Software Foundation. Available: <http://www.gnu.org/September 20, 2003>].

- Hicks, M. (2003, October 13). Mozilla Charts an Independent Course. *eWeek*.
- Hughes, R. (1999). *Mozilla Preview History*. Geek.com. Available: <http://www.geek.com/features/mozilla5/mozilla5.htm> December 3., 2003].
- Krock, E. (2003). Telephone Conversation with Former Group Product Manager for Netscape Gecko.
- Krock, E., & Ermakov, V. (2000). *Introduction to Cross-Browser, Cross-Platform, Backwardly Compatible JavaScript and Dynamic HTML*. Mozilla.org. Available: <http://www.mozilla.org/docs/web-developer/xbdhtml/xbdhtml.html> November 8, 2003].
- Lerner, J., & Tirole, J. (2002). Some Simple Economics of Open Source. *Journal of Industrial Economics*, 50(2).
- Levy, S. (2003). *The Killer Browser*, [Website]. Newsweek. Available: <http://www.msnbc.com/news/899010.asp?cp1=1> October 25, 2003].
- Mozh, V. v. (2003). *The Mozilla Museum*. Mozilla.org. Available: <http://home.snafu.de/tilman/mozilla/> December 1, 2003].
- Mozilla.org. (2003a). *All Mozilla Products*. Available: <http://www.mozilla.org/download.html> October 12, 2003].
- Mozilla.org. (2003b, October 11, 2003). *Bugs*. Available: <http://www.mozilla.org/bugs/>.
- Mozilla.org. (2003c). *Community*. Available: <http://www.mozilla.org/community.html> November 1, 2003].
- Mozilla.org. (2003d). *Firebird*. Available: <http://www.mozilla.org/products/firebird/> October 15, 2003].
- Mozilla.org. (2003e). *Module Owners*. Available: <http://mozilla.org/owners.html> November 5, 2003].
- Mozilla.org. (2003f, November 7, 2003). *Mozilla at a Glance*. Available: <http://www.mozilla.org/mozorg.html>.
- Mozilla.org. (2003g). *Mozilla Roles and Responsibilities*. Available: <http://www.mozilla.org/about/roles.html> November 15, 2003].
- Mozilla.org. (2003h). *Order Mozilla on CD*. Available: <http://store.mozilla.org/> October 21, 2003].
- Nichols, D. M., & Twidale, M. B. (2003). The Usability of Open Source Software. *First Monday*, 8(1).
- Open Source Software: Microsoft at the Power Point. (2003). *The Economist*, p.59.

- OSI. (2003a). *The Approved Licenses*, [Website]. Open Source Initiative. Available: [http://www.opensource.org/site\\_index.php](http://www.opensource.org/site_index.php)September 28, 2003].
- OSI. (2003b). *The BSD License*. Open Source Initiative. Available: <http://www.opensource.org/licenses/bsd-license.php>December 3, 2003].
- OSI. (2003c). *Mozilla Public License 1.1 (MPL1.1)*. Open Source Initiative. Available: <http://www.opensource.org/licenses/mozilla1.1.php>September 28, 2003].
- OSI. (2003d). *The Open Source Definition*, [Website]. Open Source Initiative:. Available: [http://www.opensource.org/docs/definition\\_plain.php](http://www.opensource.org/docs/definition_plain.php)September 28, 2003].
- Porter, M. E. (1985). *Competitive Advantage*. New York: Free Press.
- Porter, M. E. (2001). Strategy and the Internet. *Harvard Business Review*, 63-78.
- Rapoza, J. (2003a). *A Fighting Dragon* (July), [Website]. eWeek: Enterprise News and Reviews. Available: <http://search.epnet.com/direct.asp?an=10380814&db=afh> [20, 30].
- Rapoza, J. (2003b). *Mozilla Plans to Stick to Basics*. eWeek: Enterprise News and Reviews. Available: <http://www.eweek.com/article2/0,4149,1158505,00.asp> [20, 17].
- Raymond, E. S. (1999). *The Cathedral and the Bazaar* ( Vol. Vol. 1). CA: O'Reilly.
- Reid, R. H. (1997). *Arhitects of the Web*. New York: John Wiley & Sons, Inc.
- S.R. (2002, July 7, 2002). Beyond the Broswer Wars. *PC Magazine*, 21, 24.
- Sam, W. (2002). *Free as in Freedom: Richard Stallman's Crusade for Free Software*. CA: O'Reilly.
- Sandred, J. (2001). *Managing Open Source Projects*. New York: John Wiley & Sons, Inc.
- Sasaki, C. (2003). Vice President, Desktop Software Engineering at Sun Microsystems.
- Sayeed, L., Verma, S., & Bora, A. (2002, July, 2002). *Open Source Software Licensing and Business Models*. Paper presented at the Asia-Pacific Decision Sciences Institute, Bangkok, Thailand.
- Stanco, T. (2003). *On Open Source Procurement Policies*, [Website]December 4, 2003].
- Stellin, S. (2002, September 9, 2002). Browser Battles Wind Down. *The New York Times*, pp. 6.
- University, V. P. I. S. (1996, January 16, 1997). *WWW: Beyond the Basics*, [Website]. Available: <http://ei.cs.vt.edu/~wwwbtb/book/index.html>October 24, 2003].
- Wagner, R. (2003). *AOL Spins Off Mozilla Ownership to Independent Foundation*, [Website] [July 17, 2003, September 30,2003].

Waters, R. (2003a, June 18, 2003). The Cult of Free Software Finds Willing Devotees. *Financial Times*, 15.

Waters, R. (2003b, September 11, 2003). Microsoft Virus 'Fix' Not Enough IT Software. *Financial Times*, pp. 29.

Webopedia. (2003). *Moore's Law*. Available: [http://www.webopedia.com/TERM/M/Moores\\_Law.html](http://www.webopedia.com/TERM/M/Moores_Law.html) [September 25, 2003].

Website, N. (1998). *Netscape Announces Mozilla.org, A Dedicated Team and Web Site Supporting Development of Free Client Source Code*, [Website]. Company Press Release. Available: <http://wp.netscape.com/newsref/pr/newsrelease577.html> [September 29, 2003].