

Should Government Endorse, Promote, or Develop Free/Open Source Software Policy?

Remy DeCausemaker

Rochester Institute of Technology

Graduate Science Technology Policy Seminar

Professor Franz Foltz

In this paper, we will explore the question of whether the government has a vested interest in promoting or participating in Free/Open Source Software (FOSS) development from both a research perspective, and a policy perspective. We will explore these two scenarios through case-studies, supporting statements from executive branches and agencies, developer testimony, and a legislative analysis of an active bill on the floor of the NYS Legislature.

Before attempting to delve into the policy implications of FOSS, we must first establish and frame the existing copyright framework that makes Free/Open licensing possible, and the principles of Open Source development that guide the community.

The Open Source Way

There has been codified, a collection of normative behavioral guidelines for hackers who **choose** to participate in Open Source Software development. These guidelines were paraphrased and interpolated within a lecture at TEDxBuffalo entitled *Open Data, Open Government, and The Open Source Way*.

"What is The Open Source Way? It is a philosophy. It is a mindset. It is a production process. It is a state of being, that maximizes the potential of collaborative community development.

According to Opensource.com (2009), it has Five pillars:

1. Open Exchange

We learn more from each other when information is open. Reusing information frees up our cognitive surplus to solve new problems, and not recreate the wheel.

2. Participation

When the barriers of entry are few, and the barriers are low, creation and innovation thrive.

3. Rapid Prototyping

Learn by doing. Release early, release often. Fail early, fail often, find the right answer faster.

4. Meritocracy

In a free marketplace of ideas, everyone has access to the same information. Successful work determines which ideas rise and gather momentum from the community.

5. Community

The more people who look at a problem, the faster and more likely we are to solve it.

These principles help to explain the community-wide benefits of developing software The Open Source Way, the *why*, but not the *how*. There are often misconceptions about how FOSS licensing works, so let us begin by laying some myths to rest;

- Open Source is not "Anti-copyright" or merely a way of designating your creative works to the public domain.
- Open Source is not anti-commercial (we'll explore this further in the the *Argument for Business* Section later in this document)
- Proprietary or Private Companies are in direct zero-sum competition with Open Source Solutions
- Open Source is not based solely on altruism or idealism, it is a pragmatic production decision with economic and business aspects.

In fact, without copyright, Open Source licensing could not exist! From here, we will discuss the basics of copyright law, and offer cursory review of the FOSS licensing spectrum.

Copyright Basics (DeCausemaker, 2011 & 2013)

The constitutional rationale of copyrights is contained in Article I, Section Eight, commonly known as "The Enumerated Powers" of Congress (Constitution 1787, LXI). This subsection below is even reproduced as a forward on a page of it's own in the U.S Copyright office circular on "Copyright Law of the United States" (Circular 92 2011, 3).

*The Congress shall have Power ... To promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the **exclusive** Right to their respective Writings and Discoveries.*

United States Constitution 1787, Article I, Section 8

Copyright in the United States protects "original works of authorship," such as literary, dramatic, musical, artistic and other works (United States Copyright Office, 2008). It grants authors these *Exclusive* rights to their original works:

- To Reproduce, or copy the works;
- To make derivatives, or modify the works;

- To distribute copies of the works to the public, by sale, transfer of ownership, rental, lease, or lending;

and in the case of literary, musical, dramatic, and choreographic works, pantomimes, and motion pictures and other audio visual works

- To perform the works publicly;
- To display the works publicly;

additionally in the case of sound recordings,

- to perform the work publicly by means of a digital audio transmission.

Licensing and Intellectual Property

Exclusivity may be the default state of these copyrights (even for unpublished works) but it is for the author to decide if, when, and how to *grant* these copyrights to others in the form of a license agreement. Authors can choose to license some or all of their rights to third-parties under conditions of the author's choosing .

It is a statement of exactly what the original author permits others to do with their own original work, and any future conditions under which that work should be released. The license decision frames all collaboration that occurs, and the relationship between project stakeholders. For the most part, Free/Open Source licenses fall under two categories; copyleft and permissive.

Copyleft Licenses

(Fontana et al., 2008)

“Copyleft” is a play on the word “copyright”. Whereas copyright law has traditionally been used to withhold permission to copy, modify or distribute software, some licenses instead use copyright law to require that such permissions be granted.

Copyleft licenses are conditional licenses. One of the conditions you must satisfy before distributing copylefted software is that any changes you make to that software be likewise released under the copylefted license. A copyleft license ensures that all modified versions of your project remain free in the same way. Such licenses are said to keep code “forever free”.

FOSS licenses can have stronger, weaker or no copyleft provisions, but they all share a common goal: creation of a large pool of software that can be combined and built upon to create new works. Copyleft licenses require that those who take material from the common pool give something back as well.

Permissive Licenses

(Fontana et al., 2008)

There are many licenses commonly referred to as “BSD” or “BSD-style” licenses. Most of these licenses differ from each other in only minor ways, which is why they can be grouped together and described generally as “permissive, non-copyleft” licenses.

These licenses are permissive in that they place the bare minimum of restrictions on subsequent development and distribution. Using these licenses is as close to releasing into the public domain as FOSS licenses get ... The advantage of these and similar licenses (i.e. the ISC License) over more restrictive licenses like the GPL is that they are very tolerant of redistribution under a variety of licensing conditions, including under proprietary licenses. For some projects, having their code included in proprietary software is desirable. Many developers believe this may facilitate wide and quick adoption of the technology by both proprietary software distributors and FOSS projects.

It is somewhat common knowledge that works published by the federal government are designated within the public domain ([17 U.S.C. § 101](#)). A common misconception is that this too is true for state level publications outside of edicts of government, which is not the case (US Copyright office, 1984). It is up to each state to designate the license under which publications are made available, and this includes software.

The Argument for Government

Government as Consumer

According to Ben Balter, Executive Office of the President, Presidential Innovation Fellow, in a post on his site *Open Source for Government: A collaborative resource for government employees looking to participate in the open source community* "There are [lots of arguments](#) in favor of adopting an existing open source project over its proprietary

counterparts. A few of the biggest ones:

- Lower cost - No costly licensing, no vendor lock in
- Nimble - Often better at remaining responsive to emerging trends
- Open - Open standards encourage interoperability. You own your own data; no proprietary formats
- Fast fixes - The distributed model enables rapid detection and patching of bugs
- Customizable - You own the code. Make it do what you want "

Government as Producer

In the recently released web resource Digital Government

: Building a 21st Century Platform to Better Serve the American People, the Whitehouse has outlined a strategy that is built upon four principles or approaches; Information-centric, Shared Platform, Customer-Centric, and Security and Privacy. Below is the excerpt regarding the Shared Platform principle, which specifically mentions Open Source development as a priority:

Shared Platform

To make the most use of our resources and “innovate with less”, we need to share more effectively, both within the government and with the public. We also need to share capacities to build the systems and processes that support our efforts, and be smart about creating new tools, applications, systems, websites and domains. Ultimately, a shared platform approach to developing and delivering digital services and managing data not only helps accelerate the adoption of new technologies, but also lowers costs and reduces duplication. To do so, we need to rapidly disseminate lessons learned from early adopters, leverage existing services and contracts, build for multiple use cases at once, use common standards and architectures, participate in open source communities, leverage public crowdsourcing, and launch shared government-wide solutions and contract vehicles.^{[20](#)}

The most useful argument from a producer's standpoint is the crowdsourcing aspect of an open platform. The Consumer Financial Protection Bureau, after putting their first ever repository on Github.com, an online code forge, began to receive their first community

contributions to the code base within **days** (Balter, 2012, 2nd). Though the initial contributions were mostly trivial edits, they paved the way for continued development and engagement. The CFPB now shows a network of 19 members, with as many as eight contributors submitting code to some CFPB repositories, which are part of infrastructure and tools currently deployed in production for the agency. The same is true for the NYS Senate, and other agencies listed at GovGit.info, and informational site developed for and presented at the NYS CapitolCamp Unconference and Hackathon (Charde, 2012).

The Argument for Business

Pre-competitive Collaboration

Weber characterizes the early history of the Free Software Movement thoroughly in his extensive and authoritative research published in *The Success of Open Source*. Early computer scientists and researchers did not have the resources or tools available to build software components from scratch in a reasonable amount of time. The norm for research and development in computer programming in those days was to share code and results with peers, who would then modify and redistribute their results back to their colleagues. This according to Weber is what is now referred to in business jargon as "pre-competitive collaboration" (Weber, 2004). Fields such as Genomics (Olsen and Berger, 2011) and Biopharmaceuticals (Cohen 2010, 5) are undergoing a similar phase of market development today, and benefit from a variety of emergent "Open Innovation Networks" described by Cohen.

In 2004, *The Economist* published an article entitled "*An Open-source Shot in the*

Arm?" where the magazine describes two areas in which open source development could have huge impacts in the pre-competitive market enabled by the Open Sourcing of the original Human Genome research.

"The first is that of non-patentable compounds and drugs whose patents have expired. These receive very little attention from researchers, because there would be no way to protect (and so profit from) any discovery that was made about their effectiveness. To give an oft-quoted example, if aspirin cured cancer, no company would bother to do the trials to prove it, or go through the rigmarole of regulatory approval, since it could not patent the discovery.

The second area where open source might be able to help would be in developing treatments for diseases that afflict small numbers of people, such as Parkinson's disease, or are found mainly in poor countries, such as malaria. In such cases, there simply is not a large enough market of paying customers to justify the enormous expense of developing a new drug. America's Orphan Drug Act, which provides financial incentives to develop drugs for small numbers of patients, is one approach. But there is still plenty of room for improvement—which is where the open-source approach might have a valuable role to play."

Outside of pre-competitive market advantages, there have been identified by the Open Source Initiative a collection of Business Case Studies (OSI, 2008). Among individual cases from contractors and academics alike, there are listed four established business models for making money with Open Source:

Support Sellers (otherwise known as "Give Away the Recipe, Open A Restaurant"):

In this model, you (effectively) give away the software product, but sell distribution, branding, and after-sale service. This is what (for example) Red Hat does.

Loss Leader:

In this model, you give away open-source as a loss-leader and market positioner for closed software. This is what Netscape is doing (who would eventually become Mozilla)

Widget Frosting:

In this model, a hardware company (for which software is a necessary adjunct but

strictly a cost rather than profit center) goes open-source in order to get better drivers and interface tools cheaper. Silicon Graphics, for example, supports and ships Samba.

Accessorizing:

Selling accessories - books, compatible hardware, complete systems with open-source software pre-installed. It's easy to trivialize this (open-source T-shirts, coffee mugs, Linux penguin dolls) but at least the books and hardware underly some clear successes: O'Reilly Associates, and SSC are among them.

The Argument for Private Citizens

Why Open Licenses?

For the average citizen, there are both social and economic reasons for forgoing the traditional exclusive copyrights in lieu of more inclusive licensing strategies. According to Weber, licenses act as the practical manifestation of a social structure that underlies the open source process (Weber, 2004). Weber lists aspects of this social structure as follows:

Open source licensing schemes generally try to create a social structure that:

1. Empowers users by ensuring access to the source code.
2. Passes a large portion of the rights regarding use of the code to the user rather than reserving them for the author. In fact, the major right the author as copyright holder keeps is enforcement of the license terms. The user gains the rights to copy and redistribute, use, modify for personal use, and redistribute modified versions of the software.
3. Constrains users from putting restrictions on other users (present and future) in ways that would defeat the original goals.

(Weber, 2004)

Weber then further explains on the following page:

The principle of collaboration at work here is clear: Do nothing to complicate or slow down the widespread distribution and use of open source software. On the contrary, do everything you can to accelerate it by making open source software maximally attractive to users. This is intellectual property to be sure, but it is a concept of property configured around the right and responsibility to distribute, not to exclude (Weber 2004, 86).

Non-Rival Goods & Marginal Costs (DeCausemaker, 2011)

Rival goods are goods that confer competitive advantage obtained through ownership. I have it, you don't. Competition is intrinsic to a tangible good, and rivalry and competition for the transfer of these goods is what creates markets. If I have a beer, and I transfer ownership of it to you, I no longer have a beer.

Most tangible goods are rivalrous by the nature of being composed of a finite amount of physical resources. In the case of our beer example, even if I share the beer, the amount of beer you drink is exactly equal to the amount of beer I cannot drink. A non-rival good is one that offers no advantage of ownership, due to equal access. My ability to use a public road offers me no advantage over you, if you too, have access to use the road as you wish.

Enter digital goods. Unlike beer, the cost of the resources that comprise a digital good are roughly equivalent to the cost of disk space, which has been steadily declining as the efficiency of hardware manufacturers and "cloud" service providers has increased. To brew a beer, the cost of production includes the direct costs of the ingredients (water, wheat, barley, hops, sugar) and the indirect costs of labor, distribution, marketing, and organizational overhead. For a digital good, the production of a work is essentially a "one time" cost, and after the initial creation of a work, the marginal cost of reproduction is again, a function of the cost of hard drive space. This cost of reproduction is asymptotically approaching zero.

Take an intangible digital good, such as an image of the beer from our previous example. I can exercise my legal rights to reproduce and distribute a *copy* of that image, and transfer that copy to your hard drive, without depriving myself of the original image. There is no comparative advantage in owning this image any longer, because you too can possess a

copy of this image without impacting my ability to possess it. We can both own it, the cost of reproducing it is nearly zero, and the good ceases to be a rival good.

This crude and laborious example characterizes the concepts of rival and non-rival goods in both tangible and intangible form. A digital good does not have to be an image, but can be any digital representation or work. As Weber so eloquently surmises, the GNU/Linux Operating System is a non-rival and non-excludable good. Anyone can download a copy of GNU/Linux along with its source code for free, which means it is truly non-excludable. And because it is a digital product that can be replicated infinitely at zero cost, it is truly non-rival (Weber, 2004). This same benefit is extended to public and private sector stakeholders alike; *there is no exception made for private industry.*

Case Study

SELinux: Open Source Public/Private Partnership Exemplar in Information Security (DeCausemaker, 2011)

In 2011, as part of a multi-volume textbook series entitled *World of Cybercrime*, the SELinux project was researched and profiled. The story outlined below is taken as an excerpt from this currently unpublished manuscript, and included here as a partial case study.

Research & Policy

Based upon the policy recommendations promulgated in "The Orange Book" or DoDD 5200.28-STD; published as part of the Department of Defense *Rainbow Series*. "The Orange Book" laid out evaluation criteria for trusted computing systems, grading a computer system on a scale similar to that of a school report card--D, C, B, and A--with A1 being the most trusted and highest security. Each level or grade allowed for information of an increasing level

of sensitivity--Unclassified, Secret, Top Secret, and Confidential--to be stored, retrieved, transferred or otherwise accessed from a particular system. At the time of "The Orange Book's" release, computing technology had not developed enough to fully implement the highest recommendations of A1 and beyond. Though now superseded by other directives, DODD 5200.28-STD was the culmination of early research during the 1970's, and wasn't canceled until October 2002.

FLUX & FLASK

From 1992 to 1993, Researchers from the NSA and Secure Computing® Corporation released a prototype of their Distributed Trusted Operating System (DTOS) project to universities for research. Near the completion of the DTOS project, a new joint effort between the NSA, SCC, and the University of Utah was minted (cs.utah.edu 1999, 2nd). FLASK, or Flux Advanced Security Kernel and it's associated research was the forerunner of SELinux. By applying the lessons learned from the DTOS reports, specifications, policies, and techniques (cs.utah.edu 1998, 1st), researchers from the Computer Science Department at the University of Utah was able to provide better support for dynamic security policies (Smalley 2000, 1). Once the DTOS security architecture was successfully ported to the FLUKE research operating system (cs.utah.edu 1998, 2nd), and OSKit (cs.utah.edu 2002, 4th), there was a need to port the project yet again to a larger developer and user community; the Linux Community.

SELinux is Born

The architecture was thereafter ported and mainstreamed into Linux and other operating systems such as FreeBSD® and Solaris™ (NSA.gov 2009) and SELinux was born.

Security Enhanced Linux (SELinux) is *the* exemplar of the success of an open development project in the context of cyber crime and security. It's story is woven into the very fabric of hacker and information security culture. It's initial release was a collaboration between 4 primary entities: The National Security Agency, Secure Computing® Corporation, Network Associates Laboratories, and the MITRE Corporation (NSA.gov 2004).

Significance of the Model (DeCausmaker, 2011)

This type of public private partnership resulted in one of the most advanced and trusted technologies for securing and protecting classified and privileged information, begotten from policy directives that were unable to be implemented at the time they were promulgated. Through collaborating with private sector companies and academic research laboratories, SELinux was able to solve a very real technical challenge in a highly sensitive environment. This same security technology protects not only government servers, but enterprise business servers, as well as private citizens' machines alike, because the technology was able to be released openly.

It may seem counter-intuitive, at first, that a technology that enhances the security of an operating system can be developed publicly and out in the open, without compromising the integrity of that technology. The benefits of open development, when it comes to security, can be explained by the concepts of ductility and brittleness.

Professor Steven Bollevin of Columbia University and Randy Bush of AT&T research released a draft in 2002 entitled *Security Through Obscurity Considered Dangerous*. Obscuring, or "Hiding security vulnerabilities in algorithms, software, and/or hardware **decreases the likelihood they will be repaired and increases the likelihood that they can**

and will be exploited by evil-doers." Bellovin quotes late 19th century Dutch linguist and cryptographer Auguste Kerckhoffs' *La Cryptographie Militaire*:

"Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvenient tomber entre les mains de l'ennemi." (Roughly, "the system must not require secrecy and can fall into the hands of the enemy without causing trouble.")

(Bellovin and Bush, 2002).

This quote, commonly known as the second principle of cipher design *long* precedes the invention of the computer. In his September 2002 piece for *The Atlantic*, Charles Mann interviews security expert and hacker extraordinaire, Bruce Schneier. Schneier also beckons to Kerckhoffs' six principles, while adding a modern and holistic interpretation:

"The algorithms—the scrambling methods—are openly revealed; the only secret is the key. . . . Kerckhoffs's principle applies beyond codes and ciphers to security systems in general: every secret creates a potential failure point. Secrecy, in other words, is a prime cause of brittleness—and therefore something likely to make a system prone to catastrophic collapse. Conversely, openness provides ductility."

The "ductility" Schneier describes was enumerated by Bellovin and Bush in their ISC draft (1995) seven months prior Schneier's piece in the *Atlantic*:

Open Discussion Encourages Better Security

Open discussion and analysis of algorithms exposes weaknesses not thought of by the original authors, and thereby leads to better and more secure algorithms.

Revealing Vulnerabilities is Useful

Revealing and discussing vulnerabilities in hardware and software products allows the users to protect themselves, and encourages general protection and repair strategies.

The Culture of Sharing

In parts of the hacker subculture, information is currency. That is, by disclosing vulnerabilities or by providing exploit code, the purveyor gains status. As a consequence, knowledge of security holes tends to spread rapidly. By contrast, when security professionals withhold such information from the community, the broader community does not have an opportunity to find solutions. In extreme cases, such as that described in [Bell95], the result can be that the bad guys know about the problem long before most defenders do.

Legislative Analysis

In researching this topic, this author had come across Open Source specific policy currently active on the floor of the New York State Legislature. S0173 (and it's same-as bill in the Assembly, A03158) is "AN ACT to amend the tax law, in relation to providing a tax credit to individuals for up to two hundred dollars of expenses related to the development and posting of an open source or free license program" (NYSenate.gov, 2013). It would amend section 606 of the NY Tax Law, to allow for an individual developer to claim a credit twenty-percent of their out-of-pocket expenses (up to \$200) related to the development of software released under an OSI or FSF approved license. An individual may claim this credit over multiple years, so long as the aggregate benefit does not exceed the \$200 limit.

The Process

To obtain this legislative information, information had to be gathered from a number of services and databases. Below is a step-by-step workflow:

1. Use NYSenate.gov Open Legislation Service to locate current and past bill versions.
2. Proceed to New York State Assembly Website to download an original copy of bill, along with summary, actions, vote, memo, and bill text, and save a .pdf copy of the webpage.
3. Utilize the PDFMiner Python Library to convert the human readable printer-friendly bill text, and full bill information webpage .pdf versions to a machine readable .txt version.
4. Use the vimdiff utility to create a color-coded comparison of each version of the senate and assembly bills in a side-by-side, line-by-line comparison. (See appendices)
5. Search the Joint Commission on Public Ethics Lobbying Databases for each bill number.
6. Save a .pdf each client/firm landing page for each bill.
7. Save each client/firm retainer agreement for each bill.
8. Sift through each bi-monthly report using a web browser, searching for individual bill numbers.
9. Download, in some cases, all bi-monthly reports, or all bi-monthly reports pertaining specifically to the bill numbers in question.

10. Utilize general internet searches to find any online discussion or forums relating to each bill, outside of the JCOPE or OpenLeg databases.
11. Download from TransparencyData.org, all of the Federal Election Commission Campaign Finance Records, as well as the state-level records from National Money in State Politics database.
12. (FUTURE) parse the database of campaign contributions for
 1. instances of keywords such as 'Open Source,' 'Free Software,' and others
 2. each individual bill number
 3. each retained firm
 4. each hiring client
 5. each sponsoring member's name
 6. each sponsoring member's FilerID
 7. each sponsoring member's zipcode,
 8. all of the above methods, cross referenced with each of the listed fields

The Results

Through running the vimdiff utility (see appendices), it was easy to visualize how the bill had changed over time, since it was originally introduced in 2009. Due to the color-coding, areas with red highlight were changed, and areas of blue are portions that didn't exist until later versions. Sections outlined in purple are entire sections that had changed, with red outline being changed portions, and grey text being text that stayed the same. The parts of the bill (in both the Assembly and Senate) that actually changed were:

1. the year, session, and bill number,
2. the sponsors
3. and the placeholder for the subsection of the tax code the bill would be amending.
4. the Document ID number used internally within the legislature

Discussion and Commentary

In doing general searching across the internet for conversations and commentary about the legislation, there was both staunch support and opposition from within the Open Source community. The biggest issues that were raised had to do not with what constituted a Free/Open Source license, but determining the value of the software. It was a concern of the community that anyone could license, even the most trivial pieces of code, and receive the full tax credit, though that contribution added little or no value to the commons of FOSS licensed code.

I too, at first, was skeptical. Since reading further into the bill, I believe it was crafted in such a way that if someone were to release "useless" code, they would not be able to repeatedly claim that same piece of code each year. This is something the framers of the bill were making an effort to address.

Let us return to the question of "useless" code. As the state does not currently have a department or agency (that this author is aware of) who's job is to audit open code bases, there is likely a deficit in talent. Not necessarily a deficit in expertise (as is evidenced by GovGit.info, there are some very talented developers within the ranks of NYS Government) but a deficit in available hours and manpower. Determining what code is 'valuable' is a difficult task. Quantitative measures of value do not always translate to direct value. Just because a project has many lines of code, does not make it better quality. In fact, the more lines of code there are, the more likely it is there are bugs, unexpected behavior, or other points of failure contained within the code.

Objective measurement tools, such as linters and style checkers, as well as testing suites and frameworks do exist for many programming languages. Even so, tests can be circumvented or manipulated to show passing results, particularly if the judgment standards are made public (as they should be.)

Most qualitative measures are very subjective, and require a human to visually audit a code base. Each language would likely need its own experts and expert tools, and even then within each language there are many layers of operability and disciplines that could be subdivided. Though it would be a massive undertaking, it is technologically feasible to build the types of expert tools to help audit code submitted for tax credit.

Taking a step further back, the point of copyrights is to 'promote the useful arts and sciences.' Who are we to say that a brand new programmer, who is honestly trying to learn the craft, should be precluded from making their way into the field because they are not already skilled in the art? The only way to become skilled in the art, is to cultivate skills and relationships with mentors through actually writing code and solving problems. This would essentially ensure that tax credit were available only to the most elite developers, who are likely already employed within the industry, or are employers within the industry. These developers likely have access already to existing tax benefits associated with the cost of software development such as "depreciation of capital equipment, expensing of other out-of-pocket costs, and ... tax credit for R&D spending" (Irons and Malamud, 2006). The goal is to provide these incentives to an under-served programming population, that is hobbyists, students, and other continuing education professionals. Putting a 'value' qualification dilutes or even negates possible benefits of this incentive to the class of developer this is meant to primarily benefit.

Because the code will be publicly available, there will be a certain amount of public scrutiny and pressure from more experienced developers. Cultural pressure can enforce moral or social norms outside of legislative pressure, which is likely a more genuine and effective of a motivator to new developers than the \$200 credit. The value in open source comes from the community, not the artifact of the code. Amoral behavior will preclude lazy or malicious developers from access to the upper-echelon mentors, who are the true arbiters of norms, which will further encourage proper and useful development practices.

Should Government Be Involved?

Irons' and Malamud's report from the Center for American Progress is not just a representative study that was stumbled upon during the course of researching this paper; it was specifically cited in the bill memo A08831 under the heading "JUSTIFICATION":

*"Open Source" refers to software offered for free online download along with its instruction set ("source code"). The New York State Assembly currently uses Open Source programs such as Mozilla for email, Firefox for web browsing, and WebCal for electronic calendars. The Assembly's use of this software has led to **significant cost savings for the Assembly's operating budget**. Once an Open Source program is downloaded with its source code, anyone can view the instructions, modify the program, and redistribute it with the same freedoms. **A 2006 report authored by John Irons and Carl Malamud from the Center for American Progress details how Open Source software provides a high social value amongst the group, but a relatively low economic incentive for individual developers.***

*By allowing open access to the technology's source code, Open Source software **enhances a broader dissemination of knowledge and ideas**. Although large corporations such as IBM develop and use Open Source programs, **most Open Source software is written by individuals volunteering from home.***

*Currently, corporations and self-employed individuals who create Open Source technology are eligible for a tax deduction for their software development costs. **The vast majority of Open Source software creators (individual volunteers), however, do not receive this tax credit.** This legislation creates a similar tax credit incentive for individuals who currently have no means to deduct these expenses. **The tax credit would apply only to out-of-pocket expenses such as Internet hosting and depreciated cost of capital expenses such as computers and education.***

*The increased incentive for individuals would attract more Open Source developers to New York State, resulting in job creation and a reinvigoration of our State's technology sector. **Additionally, an influx of Open Source software would lower operating costs for larger agencies (such as the Assembly), by providing usable software without charge."***

Taking the bolded and highlighted sections of the bill memo justification

(emphasis my own), we can pull out the pieces of the argument for why it is imperative

for the government to support this type of legislation:

1. It translates to significant cost savings for the legislature's operating budget, proven by previous Open Source Software adoption.
2. Open Source Software provides high social value to the group, but low economic incentive for individual developers.
3. Open Source Software enhances a broader dissemination of knowledge and ideas.
4. Most Open Source Software is written by individual volunteers from home, with the vast majority not having a comparable tax incentive to corporations and self-employed individuals for out-of-pocket expenses like internet hosting and depreciated cost of capital expenses like computers and education.
5. Increased tax incentives would attract more Open Source developers to New York State resulting in job creation and a reinvigoration of the State's technology sector.
6. An influx of Open Source Software would lower operating costs for large agencies, by providing useful software without charge.

These arguments, though somewhat narrowly pertaining to this particular legislation in the context of New York State, are still applicable at the federal and international level. The Center for American Progress report was in fact the broader scoped justification from which the New York specific arguments were derived.

From an international perspective, there are a number of State-Sponsored Linux

Distributions (Wikipedia, 2013), which include nation-specific content, some even implementing their own standards for things like encryption and network security. If other nations such as China, North Korea, and others have taken to developing their own software, even their own Operating Systems, then this is likely an area into which our government would want to be involved, at the very least, informed.

As mentioned in the case-study on SELinux, the wider a community base, and the more thoroughly tested a piece of software is, the more ductile and responsive it can become. The best way to stress test and ensure vulnerabilities are dealt with, is to engage with as large and as expert of a community as possible.

The last argument to be made for government promotion of Open Source development through policy is based in economics. Irons and Malamud make the case for the Positive Externalities that are begotten from Open Source development, and the knowledge spillover that occurs when code is made publicly available (Irons and Malamud, 2006):

"The textbook example of an externality is the case of pollution. Consider a company that creates a product for its customers, but that creates pollution in the process. If the pollution negatively impacts those living around the factories, then the product is said to create negative externalities. Pollution abating policy might then increase over-all economic welfare.

Positive externalities can also arise in economic transactions. Such is the case when a bee-keeper's hive helps to pollinate a neighboring farmer's field. **Externalities typically lead to a situation in which the costs (or benefits) of an economic transaction to individuals differ from the broader social costs (or benefits) to the broader community. In the case of positive externalities--in which others benefit from a private transaction--the private market will tend to under-provide the good, and the government subsidies can increase overall economic efficiency.**

Consider now the case of open source software. An individual developer can, if she chooses, keep her software code private, and charge for that product. In that case the private gains would accumulate (in some proportion) to the developer and purchaser (the end users). There are not necessarily any positive externalities in this transaction.

Suppose the code were, instead, open source. The benefits from that piece of software

would then accumulate to (a greater number of) end users AND it would also benefit other developers. Users would also have a potential of even greater gain with open source, since they would have the ability to refine or enhance the product themselves..

Importantly, other developers benefit from the open-source code by

1. learning from the code,
2. incorporating their product or components into their work, and/or
3. by expanding/modifying the original code.

Open source development can thus be thought of as disseminating knowledge and ideas in addition to providing a specific product to market. And as before, these benefits accumulate to those who are not part of the initial transaction.

The positive spillover of knowledge to a broader community creates an opportunity for public policy to improve upon the private outcome. By providing incentives for open-source development (either through R&D credits for private companies, or through tax credits for individual developers) the government would be increasing economic efficiency and potentially enhancing economic growth."

This positive spillover of knowledge to the broader community can lead to massive positive network effects, due to the extended area-of-effect provided by the internet. By offering affordable software, and free access to information, this can lead to positive impacts for both 1st and 2nd degree parties, but more so for the 3rd and 4th degree. Education has been recognized as a public good that all of society has an interest in promoting, even if they are not the ones directly in school. It helps to raise the standard of living for everyone, when more people are educated. This is why everyone helps pay school taxes (even those without children) and educational institutions can be receive a 501(c)3 designation, and be granted a tax-exempt status. Open source software provides applied vocational education opportunities that supplement the public education system, and help to bridge the digital divide. Providing affordable software increases the digital standard of living for everyone, and allowing everyone to contribute to the FOSS ecosystem allows us, as a global society, to maintain and improve that standard on an ongoing basis.

Sources

1. Balter, Benjamin, J. (2012). "Open Source for Government. A collaborative resource for government employees looking to participate in the open source community." (Reviewed online 2/2/2013 at <http://ben.balter.com/open-source-for-government/>)
2. Balter, Benjamin, J. (2012). [CFPB Accepts First Citizen-Submitted Code on Behalf of Federal Government](#). (Reviewed online 2/1/2013 at <http://ben.balter.com/2012/04/15/cfpb-accepts-first-citizen-submitted-pull-request-on-behalf-of-federal-government/>).
3. Bellovin, Steven, M. and Bush, Randy. 2002. *Security Through Obscurity Considered Dangerous* New Jersey: The Internet Society
 1. [Bell95] S..M. Bellovin, "Using the Domain Name System for System Break-Ins", Proc. Fifth Usenix Security Symposium, 1995.
4. Charde, Luke. (2012). "Government on Github." (Reviewed online 11/17/2012 at <http://govgit.info/on-github.html>)
5. Cohen, Neal. 2010. *Open Innovation Networks: An Imperative for Breakthrough Therapies* Washington D.C.: Institute of Medicine of the National Academies
6. Copyright Office, United States. (1984). *Compendium II of Copyright Office Practices: Chapter 206.03*. Washington D.C.: U.S. Copyright Office.
7. Copyright Office, United States. (2011). *Circular 92: Copyright Law of the United States and Related Laws Contained in Title 17 of the United States Code*. Washington D.C.: Library of Congress.
8. DeCausemaker, J. R. (2011). *World of Cybercrime: The Open Source Chapter*. Unpublished manuscript .
9. DeCausemaker, J. R. (2013). Github.com - RIT [GradSciTechPolSem-NYS-FOSS-Bills](#). (Initially created 2/7/2013 at <https://github.com/decause/ritgradscitechpolsem-nys-foss-bills>).
10. Department of Computer Science, University of Utah (1998). "DTOS Technical Reports". (Reviewed online 11/20/2011 at <http://www.cs.utah.edu/flux/fluke/html/dtos/HTML/technical-docs.html>.)
11. Department of Computer Science, University of Utah. (1999) "Fluke: Flux μ -kernel Environment". (Reviewed online at <http://www.cs.utah.edu/flux/fluke/html/index.html>).

12. Department of Computer Science, University of Utah. (2002). "The OSKit Project". (Reviewed online 11/20/2011 at <http://www.cs.utah.edu/flux/oskit.>)
13. Department of Computer Science, University of Utah. (2011). "The Flux research Group". (Reviewed online 11/20/2011 at <http://www.cs.utah.edu/flux>).
14. Department of Defense, United States. (1985). *DEPARTMENT OF DEFENSE TRUSTED COMPUTER SYSTEM EVALUATION CRITERIA* Washington, D.C.: National Computer Security Center.
15. Economist, The. (2004). "An Open-Source Shot in the Arm?" (Reviewed online 2/1/2013 at <http://www.economist.com/node/2724420>)
16. Fontana, Richard, Kuhn, Bradley, M., Moglen, Eben, Norwood, Matthew, Ravicher, Danial, B., Sandler, Karen, Vasile, James, Williamson, Aaron. 2008. *A Legal Issues Primer for Open Source and Free Software Projects*. New York: Software Freedom Law Center.
17. House of Representatives, United States. (1787). *Constitution of the United States of America*. Washington D.C.: U.S. House of Representatives
18. House of Representatives, United States. (2002). "United States Code. Title 17 - Copyrights." (Retrieved online 2/1/2013 at http://uscode.house.gov/download/title_17.shtml)
19. Irons, John and Malamud, Carl. (2006). *An Open Source Tax Credit: Proposal and Economic Analysis*. Washington D.C.: The Center for American Progress
20. National Security Agency. (2009). "Contributors to SELinux." (Reviewed online 11/20/2011 at <http://www.nsa.gov/research/selinux/contrib.shtml>).
21. New York State Senate. (2013). New York Senate Open Legislation - S0173-2013. (Reviewed online 1/11/2013 at <http://open.nysenate.gov/legislation/bill/S173-2013>)
22. Office of Management and Budget, Executive Office of the President (2012). Digital Government: Building a 21st Century Platform to Better Serve the American People. (Reviewed online 2/1/2013 at <http://www.whitehouse.gov/sites/default/files/omb/egov/digital-government/digital-government.html>).
23. Open Source Initiative (2008). "Open Source Case for Business:Advocacy." (Reviewed online 2/2/2013 at http://opensource.org/advocacy/case_for_business.php).
24. Opensource.com. (2009). "Open Source Way - Open Source Principles." (Reviewed online 2/1/2013 at <http://opensource.com/open-source-way>).

25. Smalley, Steven. (2000). "Flask: Flux Advanced Security Kernel." (Reviewed online 11/20/2011 at <http://www.cs.utah.edu/flux/fluke/html/flask.html>).
26. Tischler, Linda. (2002). "He Struck Gold on the Net (Really)" (Reviewed online 2/1/2013 at <http://www.fastcompany.com/44917/he-struck-gold-net-really>)
27. Weber, Steven. (2004). *The Success of Open Source*. Massachusetts: Harvard University Press.
28. Welsh, Peter. Open Source and the Power of Community (2012). (Reviewed online 2/1/2013 at <http://www.whitehouse.gov/blog/2012/11/20/open-source-and-power-community>).
29. Wikipedia.org. (2013). Category: State Sponsored Linux Distributions. (Reviewed online 2/12/13 at https://en.wikipedia.org/wiki/Category:State-sponsored_Linux_distributions).

STATE OF NEW YORK

§ 173

2013-2014 Regular Sessions

IN SENATE

(PREFILED)

January 9, 2013

Introduced by Sen. SQUARON -- read twice and ordered printed, and when printed to be committed to the Committee on Investigations and Government Operations.

AN ACT to amend the tax law, in relation to providing a tax credit to individuals for up to two hundred dollars of expenses related to the development and posting of an open source or free license program

THE PEOPLE OF THE STATE OF NEW YORK, REPRESENTED IN SENATE AND ASSEMBLY, DO ENACT AS FOLLOWS:

Section 1. Section 606 of the tax law is amended by adding a new subsection (aa) to read as follows:

(aa) OPEN SOURCE OR FREE LICENSE CREDIT. (1) A TAXPAYER SHALL BE ALLOWED A CREDIT, TO BE COMPUTED AS PROVIDED IN THIS SUBSECTION, AGAINST THE TAX IMPOSED PURSUANT TO SECTION SIX HUNDRED ONE OF THIS PART, THE AMOUNT OF CREDIT SHALL EQUAL UP TO TWENTY PERCENT OF THE TOTAL OUT-OF-POCKET EXPENSES OF THE TAXPAYER USED IN THE DEVELOPMENT OF A PROGRAM THAT IS PROVIDED TO THE PUBLIC UNDER AN OPEN SOURCE OR FREE SOFTWARE LICENSE, UP TO A MAXIMUM OF TWO HUNDRED DOLLARS.

(2) FOR THE PURPOSES OF THIS SUBSECTION, A PROGRAM SHALL QUALIFY FOR THE CREDIT PROVIDED BY THIS SUBSECTION IF THE CODE FOR SUCH PROGRAM HAS BEEN RELEASED UNDER AN OPEN SOURCE LICENSE RECOGNIZED BY THE OPEN SOURCE INITIATIVE, OR HAS BEEN RELEASED UNDER A FREE SOFTWARE LICENSE RECOGNIZED BY THE FREE SOFTWARE FOUNDATION.

(3) THIS SUBSECTION SHALL APPLY TO ANY INDIVIDUAL, FOR ANY TAXABLE YEAR ONLY IF SUCH INDIVIDUAL ELECTS TO HAVE THIS SECTION APPLY FOR SUCH TAXABLE YEAR. AN ELECTION TO HAVE THIS SECTION APPLY MAY NOT BE MADE FOR ANY TAXABLE YEAR IF SUCH ELECTION IS IN EFFECT WITH RESPECT TO SUCH INDIVIDUAL FOR ANY OTHER TAXABLE YEAR AND PERTAINING TO THE SAME PROGRAM OR ANY PORTION THEREOF.

(4) IN NO EVENT SHALL THE AMOUNT OF THE CREDIT PROVIDED BY THIS SUBSECTION EXCEED THE TAXPAYER'S TAX FOR THE TAXABLE YEAR. HOWEVER, IF

EXPLANATION--Matter in ITALICS (underscored) is new; matter in brackets [] is old law to be omitted.

§ 173 2 LBO 0298-01-3

1. ANY TAXABLE YEAR RESULTS IN SUCH EXCESS AMOUNT, ANY AMOUNT OF CREDIT NOT DEDUCTIBLE IN SUCH TAXABLE YEAR MAY BE CARRIED OVER TO THE FOLLOWING YEAR OR YEARS AND MAY BE DEDUCTED FROM THE TAXPAYER'S TAX FOR SUCH YEAR OR YEARS.

§ 2. This act shall take effect immediately and shall apply to taxable years beginning on and after the first of January next succeeding the date on which it shall have become a law.

STATE OF NEW YORK

§ 4046

2009-2010 Regular Sessions

IN SENATE

APRIL 8, 2009

Introduced by Sens. SQUARON, C. JOHNSON, STACHOWSKI -- read twice and ordered printed, and when printed to be committed to the Committee on Investigations and Government Operations.

AN ACT to amend the tax law, in relation to providing a tax credit to individuals for up to two hundred dollars of expenses related to the development and posting of an open source or free license program

THE PEOPLE OF THE STATE OF NEW YORK, REPRESENTED IN SENATE AND ASSEMBLY, DO ENACT AS FOLLOWS:

Section 1. Section 606 of the tax law is amended by adding a new subsection (aq) to read as follows:

(aq) OPEN SOURCE OR FREE LICENSE CREDIT. (1) A TAXPAYER SHALL BE ALLOWED A CREDIT, TO BE COMPUTED AS PROVIDED IN THIS SUBSECTION, AGAINST THE TAX IMPOSED PURSUANT TO SECTION SIX HUNDRED ONE OF THIS PART, THE AMOUNT OF CREDIT SHALL EQUAL UP TO TWENTY PERCENT OF THE TOTAL OUT-OF-POCKET EXPENSES OF THE TAXPAYER USED IN THE DEVELOPMENT OF A PROGRAM THAT IS PROVIDED TO THE PUBLIC UNDER AN OPEN SOURCE OR FREE SOFTWARE LICENSE, UP TO A MAXIMUM OF TWO HUNDRED DOLLARS.

(2) FOR THE PURPOSES OF THIS SUBSECTION, A PROGRAM SHALL QUALIFY FOR THE CREDIT PROVIDED BY THIS SUBSECTION IF THE CODE FOR SUCH PROGRAM HAS BEEN RELEASED UNDER AN OPEN SOURCE LICENSE RECOGNIZED BY THE OPEN SOURCE INITIATIVE, OR HAS BEEN RELEASED UNDER A FREE SOFTWARE LICENSE RECOGNIZED BY THE FREE SOFTWARE FOUNDATION.

(3) THIS SUBSECTION SHALL APPLY TO ANY INDIVIDUAL, FOR ANY TAXABLE YEAR ONLY IF SUCH INDIVIDUAL ELECTS TO HAVE THIS SECTION APPLY FOR SUCH TAXABLE YEAR. AN ELECTION TO HAVE THIS SECTION APPLY MAY NOT BE MADE FOR ANY TAXABLE YEAR IF SUCH ELECTION IS IN EFFECT WITH RESPECT TO SUCH INDIVIDUAL FOR ANY OTHER TAXABLE YEAR AND PERTAINING TO THE SAME PROGRAM OR ANY PORTION THEREOF.

(4) IN NO EVENT SHALL THE AMOUNT OF THE CREDIT PROVIDED BY THIS SUBSECTION EXCEED THE TAXPAYER'S TAX FOR THE TAXABLE YEAR. HOWEVER, IF

EXPLANATION--Matter in ITALICS (underscored) is new; matter in brackets [] is old law to be omitted.

§ 4046 2 LBO09314-02-9

1. ANY TAXABLE YEAR RESULTS IN SUCH EXCESS AMOUNT, ANY AMOUNT OF CREDIT NOT DEDUCTIBLE IN SUCH TAXABLE YEAR MAY BE CARRIED OVER TO THE FOLLOWING YEAR OR YEARS AND MAY BE DEDUCTED FROM THE TAXPAYER'S TAX FOR SUCH YEAR OR YEARS.

§ 2. This act shall take effect immediately and shall apply to taxable years beginning on and after the first of January next succeeding the date on which it shall have become a law.

1	STATE OF NEW YORK	1	STATE OF NEW YORK	1	STATE OF NEW YORK
2		2		2	
3		3		3	
4	3158	4	8882	4	8831
5	2013-2014 Regular Sessions	5	2011-2012 Regular Sessions	5	
6	IN ASSEMBLY	6	IN ASSEMBLY	6	IN ASSEMBLY
7		7		7	
8		8		8	
9	January 23, 2013	9	October 27, 2011	9	(REPEALED)
10		10		10	January 4, 2012
11		11		11	
12		12		12	
13	Introduced by N. of A. QUART, JAFFEE, NEPPON -- Multi-Sponsored by -- N. of A. BOTLAND -- read once and referred to the Committee on Ways and Means	13	Introduced by N. of A. KELLER -- read once and referred to the Committee on Ways and Means	13	Introduced by N. of A. QUART -- read once and referred to the Committee on Ways and Means
14		14		14	
15		15		15	
16	AN ACT to amend the tax law, in relation to providing a tax credit to individuals for up to two hundred dollars of expenses related to the development and posting of an open source or free license program	16	AN ACT to amend the tax law, in relation to providing a tax credit to individuals for up to two hundred dollars of expenses related to the development and posting of an open source or free license program	16	AN ACT to amend the tax law, in relation to providing a tax credit to individuals for up to two hundred dollars of expenses related to the development and posting of an open source or free license program
17		17		17	
18		18		18	
19		19		19	
20	THE PEOPLE OF THE STATE OF NEW YORK, REPRESENTED IN SENATE AND ASSEMBLY, DO ENACT AS FOLLOWS:	20	THE PEOPLE OF THE STATE OF NEW YORK, REPRESENTED IN SENATE AND ASSEMBLY, DO ENACT AS FOLLOWS:	20	THE PEOPLE OF THE STATE OF NEW YORK, REPRESENTED IN SENATE AND ASSEMBLY, DO ENACT AS FOLLOWS:
21		21		21	
22		22		22	
23		23		23	
24	1. Section 606 of the tax law is amended by adding a new	24	1. Section 606 of the tax law is amended by adding a new	24	1. Section 606 of the tax law is amended by adding a new
25	subsection aa to read as follows:	25	subsection tt to read as follows:	25	subsection tt to read as follows:
26	3. (AA) OPEN SOURCE OR FREE LICENSE CREDIT. (1) A TAPAYER SHALL BE	26	3. (TT) OPEN SOURCE OR FREE LICENSE CREDIT. (1) A TAPAYER SHALL BE	26	3. (TT) OPEN SOURCE OR FREE LICENSE CREDIT. (1) A TAPAYER SHALL BE
27	ALLOWED A CREDIT, TO BE COMPUTED AS PROVIDED IN THIS SUBSECTION, AGAINST	27	ALLOWED A CREDIT, TO BE COMPUTED AS PROVIDED IN THIS SUBSECTION, AGAINST	27	ALLOWED A CREDIT, TO BE COMPUTED AS PROVIDED IN THIS SUBSECTION, AGAINST
28	THE TAX IMPOSED PURSUANT TO SECTION SIX HUNDRED ONE OF THIS PART. THE	28	THE TAX IMPOSED PURSUANT TO SECTION SIX HUNDRED ONE OF THIS PART. THE	28	THE TAX IMPOSED PURSUANT TO SECTION SIX HUNDRED ONE OF THIS PART. THE
29	AMOUNT OF CREDIT SHALL EQUAL UP TO TWENTY PERCENT OF THE TOTAL	29	AMOUNT OF CREDIT SHALL EQUAL UP TO TWENTY PERCENT OF THE TOTAL	29	AMOUNT OF CREDIT SHALL EQUAL UP TO TWENTY PERCENT OF THE TOTAL
30	OUT-OF-POCKET EXPENSES OF THE TAPAYER USED IN THE DEVELOPMENT OF A	30	OUT-OF-POCKET EXPENSES OF THE TAPAYER USED IN THE DEVELOPMENT OF A	30	OUT-OF-POCKET EXPENSES OF THE TAPAYER USED IN THE DEVELOPMENT OF A
31	PROGRAM THAT IS PROVIDED TO THE PUBLIC UNDER AN OPEN SOURCE OR FREE	31	PROGRAM THAT IS PROVIDED TO THE PUBLIC UNDER AN OPEN SOURCE OR FREE	31	PROGRAM THAT IS PROVIDED TO THE PUBLIC UNDER AN OPEN SOURCE OR FREE
32	SOFTWARE LICENSE, UP TO A MAXIMUM OF TWO HUNDRED DOLLARS.	32	SOFTWARE LICENSE, UP TO A MAXIMUM OF TWO HUNDRED DOLLARS.	32	SOFTWARE LICENSE, UP TO A MAXIMUM OF TWO HUNDRED DOLLARS.
33	(2) FOR THE PURPOSES OF THIS SUBSECTION, A PROGRAM SHALL QUALIFY FOR	33	(2) FOR THE PURPOSES OF THIS SUBSECTION, A PROGRAM SHALL QUALIFY FOR	33	(2) FOR THE PURPOSES OF THIS SUBSECTION, A PROGRAM SHALL QUALIFY FOR
34	THE CREDIT PROVIDED BY THIS SUBSECTION IF THE CODE FOR SUCH PROGRAM HAS	34	THE CREDIT PROVIDED BY THIS SUBSECTION IF THE CODE FOR SUCH PROGRAM HAS	34	THE CREDIT PROVIDED BY THIS SUBSECTION IF THE CODE FOR SUCH PROGRAM HAS
35	BEEN RELEASED UNDER AN OPEN SOURCE LICENSE RECOGNIZED BY THE OPEN SOURCE	35	BEEN RELEASED UNDER AN OPEN SOURCE LICENSE RECOGNIZED BY THE OPEN SOURCE	35	BEEN RELEASED UNDER AN OPEN SOURCE LICENSE RECOGNIZED BY THE OPEN SOURCE
36	INITIATIVE, OR HAS BEEN RELEASED UNDER A FREE SOFTWARE LICENSE RECOGNIZED BY THE FREE SOFTWARE FOUNDATION.	36	INITIATIVE, OR HAS BEEN RELEASED UNDER A FREE SOFTWARE LICENSE RECOGNIZED BY THE FREE SOFTWARE FOUNDATION.	36	INITIATIVE, OR HAS BEEN RELEASED UNDER A FREE SOFTWARE LICENSE RECOGNIZED BY THE FREE SOFTWARE FOUNDATION.
37	(3) THIS SUBSECTION SHALL APPLY TO ANY INDIVIDUAL, FOR ANY TAXABLE YEAR	37	(3) THIS SUBSECTION SHALL APPLY TO ANY INDIVIDUAL, FOR ANY TAXABLE YEAR	37	(3) THIS SUBSECTION SHALL APPLY TO ANY INDIVIDUAL, FOR ANY TAXABLE YEAR
38	ONLY IF SUCH INDIVIDUAL ELECTS TO HAVE THIS SECTION APPLY FOR SUCH TAXABLE YEAR. AN ELECTION TO HAVE THIS SECTION APPLY MAY NOT BE MADE FOR ANY	38	ONLY IF SUCH INDIVIDUAL ELECTS TO HAVE THIS SECTION APPLY FOR SUCH TAXABLE YEAR. AN ELECTION TO HAVE THIS SECTION APPLY MAY NOT BE MADE FOR ANY	38	ONLY IF SUCH INDIVIDUAL ELECTS TO HAVE THIS SECTION APPLY FOR SUCH TAXABLE YEAR. AN ELECTION TO HAVE THIS SECTION APPLY MAY NOT BE MADE FOR ANY
39	TAXABLE YEAR IF SUCH ELECTION IS IN EFFECT WITH RESPECT TO SUCH INDIVIDUAL FOR ANY OTHER TAXABLE YEAR AND PERTAINING TO THE SAME PROGRAM OR ANY	39	TAXABLE YEAR IF SUCH ELECTION IS IN EFFECT WITH RESPECT TO SUCH INDIVIDUAL FOR ANY OTHER TAXABLE YEAR AND PERTAINING TO THE SAME PROGRAM OR ANY	39	TAXABLE YEAR IF SUCH ELECTION IS IN EFFECT WITH RESPECT TO SUCH INDIVIDUAL FOR ANY OTHER TAXABLE YEAR AND PERTAINING TO THE SAME PROGRAM OR ANY
40	PORTION THEREOF.	40	PORTION THEREOF.	40	PORTION THEREOF.
41	(4) IN NO EVENT SHALL THE AMOUNT OF THE CREDIT PROVIDED BY THIS	41	(4) IN NO EVENT SHALL THE AMOUNT OF THE CREDIT PROVIDED BY THIS	41	(4) IN NO EVENT SHALL THE AMOUNT OF THE CREDIT PROVIDED BY THIS
42	SUBSECTION EXCEED THE TAPAYER'S TAX FOR THE TAXABLE YEAR. HOWEVER, IF	42	SUBSECTION EXCEED THE TAPAYER'S TAX FOR THE TAXABLE YEAR. HOWEVER, IF	42	SUBSECTION EXCEED THE TAPAYER'S TAX FOR THE TAXABLE YEAR. HOWEVER, IF
43	THE AMOUNT OF CREDIT OTHERWISE ALLOWABLE PURSUANT TO THIS SUBSECTION FOR	43	THE AMOUNT OF CREDIT OTHERWISE ALLOWABLE PURSUANT TO THIS SUBSECTION FOR	43	THE AMOUNT OF CREDIT OTHERWISE ALLOWABLE PURSUANT TO THIS SUBSECTION FOR
44	ANY TAXABLE YEAR RESULTS IN SUCH EXCESS AMOUNT, ANY AMOUNT OF CREDIT NOT	44	ANY TAXABLE YEAR RESULTS IN SUCH EXCESS AMOUNT, ANY AMOUNT OF CREDIT NOT	44	ANY TAXABLE YEAR RESULTS IN SUCH EXCESS AMOUNT, ANY AMOUNT OF CREDIT NOT
45	DEDUCTIBLE IN SUCH TAXABLE YEAR MAY BE CARRIED OVER TO THE FOLLOWING	45	DEDUCTIBLE IN SUCH TAXABLE YEAR MAY BE CARRIED OVER TO THE FOLLOWING	45	DEDUCTIBLE IN SUCH TAXABLE YEAR MAY BE CARRIED OVER TO THE FOLLOWING
46	YEAR OR YEARS AND MAY BE DEDUCTED FROM THE TAPAYER'S TAX FOR SUCH YEAR OR YEARS.	46	YEAR OR YEARS AND MAY BE DEDUCTED FROM THE TAPAYER'S TAX FOR SUCH YEAR OR YEARS.	46	YEAR OR YEARS AND MAY BE DEDUCTED FROM THE TAPAYER'S TAX FOR SUCH YEAR OR YEARS.
47	S 2. This act shall take effect immediately and shall apply to taxable years beginning on and after the first of January next succeeding the date on which it shall have become a law.	47	S 2. This act shall take effect immediately and shall apply to taxable years beginning on and after the first of January next succeeding the date on which it shall have become a law.	47	S 2. This act shall take effect immediately and shall apply to taxable years beginning on and after the first of January next succeeding the date on which it shall have become a law.
48		48		48	
49	EXPLANATION--Matter in ITALICS (underscored) is new, matter in brackets [] is old law to be omitted.	49	EXPLANATION--Matter in ITALICS (underscored) is new, matter in brackets [] is old law to be omitted.	49	EXPLANATION--Matter in ITALICS (underscored) is new, matter in brackets [] is old law to be omitted.
50		50		50	
51	A. 3158 2 LEO9029-01-3	51	A. 8882 2 LEO13487-01-1	51	A. 8831 2 LEO0164-03-1
52		52		52	
53	1. ANY TAXABLE YEAR RESULTS IN SUCH EXCESS AMOUNT, ANY AMOUNT OF CREDIT NOT DEDUCTIBLE IN SUCH TAXABLE YEAR MAY BE CARRIED OVER TO THE FOLLOWING YEAR OR YEARS AND MAY BE DEDUCTED FROM THE TAPAYER'S TAX FOR SUCH YEAR OR YEARS.	53	1. DEDUCTIBLE IN SUCH TAXABLE YEAR MAY BE CARRIED OVER TO THE FOLLOWING YEAR OR YEARS AND MAY BE DEDUCTED FROM THE TAPAYER'S TAX FOR SUCH YEAR OR YEARS.	53	1. YEAR OR YEARS AND MAY BE DEDUCTED FROM THE TAPAYER'S TAX FOR SUCH YEAR OR YEARS.
54		54		54	
55		55		55	
56		56		56	
57		57		57	
58		58		58	
59		59		59	
60		60		60	