How to Succeed with Open Source Software Without Losing Your Shirt

The use of Open Source software has grown dramatically in recent years. Linux is probably the most visible example of this trend. Linux is the fastest growing operating system in the world and in 1999 more Linux was shipped than all of the UNIX products combined, according to an IDC report released in April of 2000. It is also estimated that 75% of all software that runs the internet is Open Source software. There are numerous reasons for the success of Open Source software including:

- Open Source software is inexpensive and the licensing permits unlimited copying
- Much Open Source software is extremely reliable because Open Source software allows for broader peer review and bugs are often fixed faster because more people have access to the source code
- Open Source software allows the user to change the software if needed
- Open Source software is evolving very quickly because of the ease of redistribution and some implementations have become very widespread (62% of all web servers are based on Apache)
- Open Source software is not limited to a particular hardware platform
- In 1999, companies like HP, IBM, SGI and others began to support Linux and other Open Source software. This provided respected suppliers of support around the world and a commitment to a future.

How can companies participate in this Open Source revolution beyond simply using the software? And what are your obligations when you use Open Source software?

First, understand the various licenses are important. A good reference is a collection of articles called Open Sources, published by O'Reilly. An electronic version can be found at the O'Reilly web site, http://www.oreilly.com/catalog/opensources/book/toc.html. Here are some of the highlights for Open Source software definition from the Open Source Initiative started by Eric Raymond and Bruce Perens.

An Open Source Software License must include:

- Source code must be available to licensee without significant charge
- License does not restrict licensee from royalty free redistribution
- The right to make modifications & allow redistribution under the same terms

There are a number of licenses that are compatible with this definition and yet some are more restrictive than others are. For example, Apache, the most popular web server in the world, is available under the so-called Berkeley style license. This license is extremely unrestrictive. It allows you to do anything you want with the software including copy, make changes, redistribute the software and even use the software and your changes in a proprietary product. You have no obligation to release the source code for your changes. However, you must give credit to the copyright owners of the original code.

The most widely used license is the GNU General Public License (often simply referred to as the GPL), that was created by Richard Stallman of the Free Software Foundation. The intent of this license is to ensure that the software remains available for everyone. A copy of the GPL is available at www.fsf.org. The GPL allows the user to make unlimited copies, change the source and redistribute the software. However it also requires that if you distribute/sell this software, you must make the source code available upon request to the people to whom you distribute/sell the software - including source code for any modifications you include in your distribution. These modifications must be licensed under the terms of the GPL. Thus, if you integrate your code tightly with software licensed under the GPL, your code becomes open source as well. This is one way that you **can lose your shirt** making a business with Open Source software. So how do you avoid this?

The main criteria:

1) Do not include any GPL code in your program (including by linking GPL object code into your executables) and interact with the GPL code only through well-defined interfaces.
2) You can link with software which is released under the closely related GNU Lesser (or Library) General Public License (LGPL, which permits static linking of LGPL software into proprietary software).

In general this is pretty easy, simply running on top of the Linux operating system for example is perfectly fine. Your software is simply using the standard Linux system calls or libraries when it runs.

How to Make Money with Open Source Software

- **Sell hardware that supports Linux or other Open Source software (e.g. HP Visualize Workstations, Kayaks, & NetServers)**
- **Sell proprietary software ported to Linux (e.g. OpenMail for Linux)**
- **Sell value added for Open Source software (configured, certified, customized, extended; e.g. Red Hat, TurboLinux, SuSE, Caldera)**
- **Sell bundled HW & SW as specialized client and server appliances (e.g. HP Jet Direct 4000 print server, Cobalt web servers)**
- **Sell electronic e-services based on Open Source software (e.g. Collab.net web and development hosting services)**
- **Sell traditional services for Linux & Open Source (support, education, consulting, custom engineering)**
- **Operate portals on an ad revenue model related to Open Source (e.g. Slashdot)**

HP participates in the Open Source market place in almost all of the above business and has contributed to the Open Source community in numerous areas. Significant software which HP has released includes kernel modules for Linux on IA-64, all of the initial e-speak code, significant enhancements to SAMBA, a large model graphics modeling software and numerous ports of Open Source software to the HP 9000. You can get more information on HP's Linux and Open Source strategy and software contributions at www.hp.com/linux.

Participating in the Open Source Process as an IT Organization

These are most of the key ways to build a business around Open Source software. However there are benefits to participating in the Open Source community as an end user IT department as well. Here are a few examples:

- Identifying and reporting a bug in Open Source software is really no different than what most end user organizations do today. However, because the source code is available to you, you also have the opportunity to fix these bugs yourself. This can result in faster changes being available for your organization and everyone who uses the software. Since this often requires interacting with outside developer, this process also helps expand your network of resources.

- This process of identifying and fixing bugs can be an excellent training ground for IT professionals. In some ways, it can be considered ongoing education. Also, by working in the relevant Open Source projects, you will be aware of enhancements and fixes by others that may be useful.

- Contributing internally developed software as Open Source can be valuable in exchange for ongoing maintenance by a community of users. CISCO did this with software that they developed for managing printing in their offices around the world. The software had been developed by two people at CISCO and leveraged a file and print application called SAMBA. By contributing this software to the SAMBA project, CISCO ensured that future capabilities of SAMBA would also benefit CISCO's worldwide operations, even if the two original developers left the company.

- As the base of high quality Open Source software grows, the ease and practicality of building on top of Open Source software increases. Yahoo builds on top of the BSD operating system, which is Open Source. Yahoo also contributes many significant changes back to the community that maintains the base software. In this way Yahoo extends it's IT resources and allows it to focus on its value add. But the Open Source software model also allows Yahoo to fix problems and work on enhancement that it

considers most critical. It provides a new paradigm where end users can be more active in moving the technology forward.

Things to Consider When Releasing Software to Open Source

Prepare the Code

This first thing is to make sure that there is no intellectual property that the company wants to retain. Assuming you have passed this hurdle, there are several things that must be done to prepare the code itself. The code should be “cleaned up”. This would include adding remarks to help communicate the purpose of a section and removing remarks that are inappropriate including personal remarks or unnecessary company specific references. If this is a large piece of code and the code is confusing or poorly written, this may also hamper its acceptance by the core group of Open Source maintainers.

You also need to make sure that your company has the right to distribute the software under an Open Source license. This includes reviewing the code to make sure that there is no third party software included in the software that you are releasing. Often this will be reflected in third party copyright notices in the source code itself.

Pick the Right License

If you are contributing code to a specific project, like the GNOME desktop project, the software license must be compatible and usually it needs to be identical to the other software in the project to get incorporated into the main source code tree. Otherwise, your choice of license should be driven by your objectives: Open Source licenses differ in a variety of ways.

Copyright Notice

You will want to put a copyright notice in the code. The copyright notice identifies who is providing the code and also gives notice as to ownership. Although you release software under a particular Open Source license, you can still retain ownership rights. These rights include the right to sue someone who does not use the software according to the license. As the owner you can also retain the right to use the software for other proprietary uses (of course, this right is not useful if the code is only usable as a modification to GPL code).

Work with the Community

Although anyone can make and distribute their own modifications, write access to the main tree for an Open Source project is accessible only by select individuals that have a proven record of contribution to the project. It is this inner circle that along with the other members of the community define the rules for contributing software and decide which enhancements and even bug fixes get into the official tree (a tree that is "official" only because it is widely regarded as the primary version). The significance of this process is that the larger the modification to the code, the more important a collaborative process of working with the main developers is to getting your modifications included.

The bottom line is that you can not assume that your additions will be incorporated into the main tree. If work has already proceeded in an area that you have made changes there may be conflicts with the contributions. This makes it very important to develop an ongoing relationship with the community if you anticipate releasing software as Open Source and want to get the benefits of integrating your changes into the main tree.

Sometimes the established practice of those maintaining the official source tree includes assigning ownership rights over to the current guardians/owners. For example, gcc is owned by the Free Software Foundation (FSF) and ownership of all code in the “official” tree must be assigned to the FSF before being accepted.

If a company has a separate piece of software that they want to release as Open Source, they need to be prepared to build/attract a group of interested developers in maintaining and moving forward this Open Source application. HP did this with e-speak. This software was born out of HP Labs and provides the glue for creating standards based e-services in a dynamic way. HP realized the benefits that could be had by creating a pervasive software platform and working with the community and industry to extend the platform. HP is working with Collab.Net to help create a community of developers and interested corporations who will help move the software forward. This software is released under the GPL and LGPL and you can get the source from www.e-speak.net. There are over 5000 developers who have registered and downloaded the software.

In summary, Open Source software is growing rapidly and the opportunities and benefits of contributing to this growing base of software are very real. Working with the community supporting the core software is key as well as understanding the licensing rules. This is particularly true for larger pieces of software. In addition, larger software requires more extensive clean up of the code and review for third party contributions. A key benefit of open sourcing software is to expand the number of people maintaining and enhancing the software. There are also personal benefits to IT professionals participating in the Open Source community.