

WORKING WITH THE FREE SOFTWARE COMMUNITY TO REPLACE A RADIO BROADCAST AUTOMATION SYSTEM

Federico Grau

Radio Free Asia
Washington DC
2007-01-19

ABSTRACT

Like many radio stations, Radio Free Asia (RFA) has a minimal operating budget and an aging radio broadcast automation system. While their current system is all digital and has served them well, it is reaching the end of its life cycle. For their next generation replacement they searched the Internet for available Free Software components. The most mature project found is the Rivendell radio broadcast automation system authored by Fred Gleason of Salem Radio Labs, and released under the GNU General Public License (GPL).

This paper will start with a review of what Free Software is, how an organization can use it, and how and why to contribute back. In the case of Radio Free Asia, one example is their investment into modifying the Rivendell system to work in a multi-user environment. As part of their deployment, RFA has integrated several Free Software components such as the Linux High Availability (HA) project and the Linux Terminal Server Project (LTSP) to produce a robust working solution.

The paper will also elaborate on the differences in working with a Free Software project versus a commercial solution. For example, when it comes to support, with a commercial solution a problem will be approached by dialing a dedicated support phone number (and possibly waiting for a solution); with a Free Software project more effort and interaction is possible and expected. Avenues of support with Free Software include participating in mailing lists, reading documentation on community wikis (web sites that allow visitors to easily edit the content of the web pages), reviewing the source code, and working with consultants who know the field. In addition to these steps, RFA has helped establish a dedicated Internet Relay Chat (IRC) channel for the project, where users and developers can help support each other in real time.

HISTORY

Radio Free Asia

RFA is a large organization with over 50 studios, a staff of 300, transmitting 36 hours per day in 9 different languages. They are a non-profit organization and therefore have a fixed budget to work with. As a result there is a strong “do it yourself” attitude and an openness to trying new ideas and technologies. The facility has been all digital since its inception in the 1996. RFA has a history of using and working with Free Software since the late 1990s, documenting and sharing their efforts with the community in the Radio Broadcast Open Source System (RBOSS) hosted on the techweb.rfa.org server.

Their current broadcast automation system is a commercial all digital solution that has worked well, but it is almost 10 years old. There are increasing hardware failures and software support is reaching its end of life cycle. Because of the scale of the organization, the project of upgrading the facility is a large and challenging endeavor.

Free Software

The ideas behind Free Software were published by Richard Stallman (RMS) during the early 1980s as he observed software distribution practices changing. Where previously computer source code was commonly distributed along with software, allowing one to study and improve a computer program, this was becoming less and less the case with the onset of the microcomputer. RMS saw the importance in allowing one to see computer code as society and the economy become more dependent on computers.

The key ideas behind Free Software are to ensure freedoms as in liberty, “libre”, or free speech. They do not necessarily guarantee free as in cost, “gratis”, or free beer. Those freedoms are listed in Table 1 below. They

encourage sharing and cooperation and result in continuously improving software as long as a community thrives around it. Software has a fixed amount of effort that is required to produce a quality product; one example is the office suite which was functionally complete around the release of Microsoft® Office version 95. With the mass collaboration possible via the Internet, very high quality results can come from Free Software, some projects more further along than others.

Table 1: Guarantees of Free Software

• to run the program, for any purpose
• to modify the program to suit one's needs (this means access to the source code)
• to redistribute copies, either gratis or for a fee
• to distribute modified versions of the program

Recognizing the importance of Free Software, in 1984 RMS started the GNU's Not Unix (GNU) project to create a complete UNIX-like operating system comprised of Free Software. To this software project he applied the GNU GPL which ensures the guarantees listed in Table 1 above. The GNU project has evolved into one of the key components in many GNU/Linux distributions available today such as Debian, Ubuntu, SUSE, Redhat.

Free Software has grown in popularity with the development of the Internet – both acting as catalysts to help the other grow. The Internet has allowed groups from all over the world to cooperate and produce more software projects. Similarly, the Internet is based on open standards and many portions run on Free Software, which has a large collection of programs that work with those standards. The most popular example in the media today is the Linux kernel, most commonly shipped in GNU/Linux distributions. The number of Free Software projects is in the thousands and growing (there are 42,000 projects registered at freshmeat.net and 139,000 at sourceforge.net).

So what drives one to use and develop Free Software, especially when there is less of a guarantee to profit? Often the motivation is to “scratch an itch”: solving one's own computing problem by writing some software. A good programmer knows how to write code, a better programmer knows how

to reuse code. Using Free Software, there is a growing base of work done by others on which to build upon and a culture that encourages further cooperation.

The result is a work of passion, necessity or both. Unlike commercial ventures, it is not restricted by budgets or marketing deadlines. What ties the community together is not profit, but a common interest and goal (such as broadcasting effectively).

Radio And Free Software

Radio and Free Software are well suited to one another with commonalities in cultures. Hacking, or modifying a tool to better fit ones needs, has been a long term practice in the broadcast arena. When a new piece of equipment would come in, engineers would study the technical manuals and schematics that were standard with pro-audio grade gear. This would enable them to learn how the unit worked and allow them to support it. They might make improvements such as adding buttons to a console, replacing the heads on a tape machine, or adding a remote control. In the 21st century an increasing number of the tools in the broadcast field are virtual and software based. Consoles are now Internet Protocol (IP) based; recorders and editors are digital software programs instead of physical machines. The lines between engineer, system administrator, and software developer are blurring. Using Free Software there is opportunity for modern radio broadcast engineers to see and hack the code of the tools they use to better suit them. The source code is the ultimate technical manual and schematic defining how a software system works.

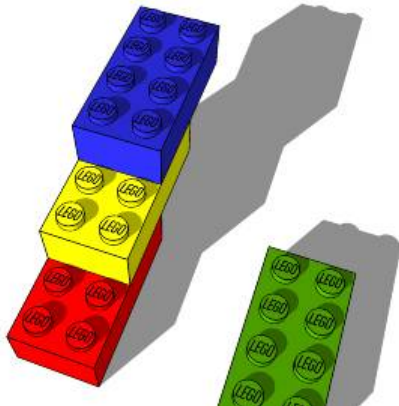
The radio industry has a thriving community of organizations using, sharing, and developing Free Software. They cluster around projects that share common goals, such as the Rivendell broadcast automation system. The #rivendell IRC channel, has participants from Eastern, Central, and Western US, the Caribbean, Spain, England, France, Germany, Hungary, Mexico, and the Philippines, to name a few.

THE SWITCH

Design

Before starting the Rivendell project, RFA had been working with a commercial vendor to replace their broadcast automation system. During the testing phase, the project came to a halt when, due to a faulty driver that would not be corrected in the foreseeable future, a cost prohibitive hardware replacement was required. RFA was in an adverse situation beyond their control. They reviewed their options, and while the use of Free Software was not as feature-rich as the commercial offerings and it required some initial effort, it positioned them on a path that guaranteed security and control over their organization. The software has proven to be robust enough for the broadcast environment, and its feature set continues to improve.

Rather than working with a vendor and adapting their work model to the the vendor's product line, RFA began a twofold study. First a review of audio related projects available upon which to build. Members of the community, like Dave Phillips and his site linux-sound.org, have filled the role of exploring, reviewing, and summarizing audio software that the community develops. With the open development process that is common of this community, one can review and track the progress on most any component of the software system, even down to the most low level driver. While it is highly technical with a very steep learning curve, it does allow for direct contact with the developers when trying to expedite the fix of a critical bug.



The second study RFA performed was a review of their current work flow. A migration plan for current and long term deployment was developed. Features were

evaluated and prioritized whether they were critical requiring completion before the initial deployment or if they could be postponed as a later improvement.

The next step was to pick out the best components and assemble them into a working system, a bit like building with LEGOs. For simple configurations the blocks to assemble are straightforward; download and install the SUSE Linux DVD iso, and then download and install the Salem Radio Labs packages for Rivendell. After a little configuration, most of it GUI based, the system pretty much takes care of itself. A more complex solution was required for the RFA deployment to work well with their environment, provide redundancy, and better support a multi-user environment.

Components that needed to be improved for the delivery plan were identified by RFA staff. Some effort was done by in house engineers; examples include creating a Pluggable Authentication Module (PAM) library to better control Rivendell authentication for multi-user environments, and improving the installation by adding wizard dialogs rather than requiring editing of a configuration file. Other efforts were outsourced to consultants, the main example is an improved integration between the Audacity multi track audio editor and the Rivendell system. When completed these improvements are contributed back to the community, improving the product for the group at large as a result of RFA scratching its itch.

To the good surprise of RFA, other items that had been planned as "later improvements" for their deployment system have been contributed to by other members of the Rivendell community. The ability to store audio in the compressed audio formats Ogg Vorbis and FLAC are examples of new features contributed by others in the community while RFA was in their testing phase.

With commercial software one has to use their products in the ways designed and planned for by them. With some thought and Free Software a better fitting end solution, one that is capable of evolving with an organizations needs, can be delivered.

Building Blocks

How does one obtain Free Software? Individual projects will often host information, source code, and documentation on Internet web pages available for download. Distributions are projects that collect many components and then prepare them by compiling and packaging them to be easily installable. RFA used both types of resources in their deployment.

The centerpiece of the RFA deployment plan is the Rivendell broadcast automation system. This is the portion that does the actual play-out to broadcast. Table 2 lists many of the pieces used in the deployment.

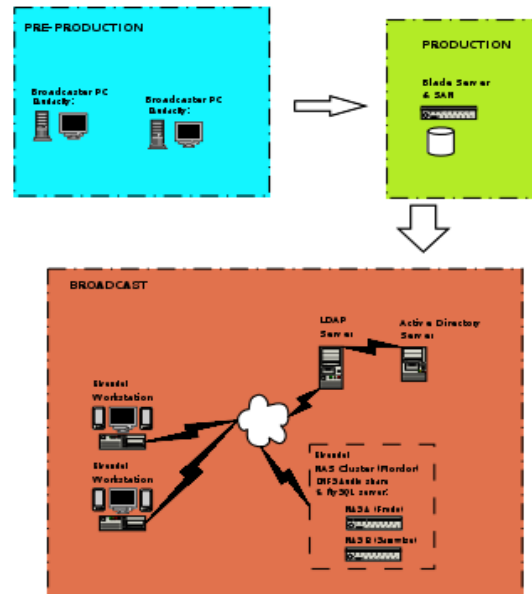
Table 2: Software Components Used

Rivendell – Broadcast automation system with a virtual library of carts, log creation, and play-out.
Audacity – Multi track audio editor, used to create and modify content.
KDE – User friendly desktop environment, with a user interface similar to the MS Windows® desktop.
Samba, NFS – Network file sharing for MS Windows® and GNU/Linux computers.
Linux High Availability project – used to provide redundancy with the back end servers.
LTSP – Linux Terminal Server project, that allows for disk-less deployment of workstations enabling easier administration in large deployments.
Linux Kernel – Core software component that talks to the hardware.
ALSA – Sound driver portion of the Linux kernel.
JACK – Low-latency audio server, routing audio amongst multiple programs and a sound card.
Portaudio - Portable audio library/API, used by Audacity.
Ardour – A full featured digital audio workstation.
OpenOffice – Free office suite.
Firefox – Web browser.
Gaim – Instant messenger chat utility.
Debian – Distribution that groups most of these packages together.

Support

A great part of the value that is paid for in a commercial product is the support. It is to receive support and continued updates that recurring yearly maintenance contracts are made.

Rivendell Workflow and LAN diagram



With Free Software there exist more avenues for support. One can purchase support contracts for Free Software products from large organizations such as SUSE, IBM, MySQL. Support can also be purchased from consultants specialized in the field on an as needed or contractual basis.

Users also have the option of helping themselves. The same tools of the Internet that allow developers to cooperate and build the software, allow users to interact and support each other in a distributed fashion. Some of those tools include email lists for discussion of problems and wikis. Wikis are excellent tools for allowing collaboration on documentation. IRC is a common group chat forum used to discuss problems in real time.

Finally, with Free Software there is always the source code available to review. While the idea may seem daunting to the uninitiated, good tools make exploring easier. Much can be learned by searching the source code for an error message a program produces, and then scanning the nearby code for comments written by the developer.

On the flip side, ways of supporting a project include micro-payments or sponsorship on web sites hosting the project. Similarly, some projects sell CDs with their software. There are also non-monetary ways of supporting projects such as offering bandwidth or server space to host a project

on the Internet. Hardware manufacturers have incentive to share sample models with established developers so they can author drivers. This results in a larger group of people buying and using the hardware device.

Returning to the idea of scratching an itch, RFA recognized a need for an improved support community around the Rivendell project. They researched what was already present and found the mailing list and community wiki. They began with participation in the email list and adding documentation to the wiki. Seeking a more live support forum, they fostered the IRC channel #rivendell on irc.freenode.net . The result has been a growth in participation on the IRC channel from 2 or 3 people one year ago to over a dozen regular users currently. With more users, the greater the speed of getting a response to a problem.

CONCLUSION

With the switch to using Free Software for their broadcast automation system, RFA has transitioned from being in an adverse situation they could not control, to now having direct influence in the development of their core infrastructure. No longer will they have to worry about product availability changing, or worse disappearing, due to market fluctuations beyond their control such as a company being sold or going out of business. Artificial constraints that limit or lock in a user to a piece of software are no longer a concern. No key dongle is required for the system to function. File formats that are used are open and well documented allowing interoperability, not restricted with expensive patents.

Benefits include a tight integration path throughout their organization. In time the broadcast system will have bindings to other systems such as the WWW Content Management System (CMS). Support now includes dedicating some staff to the project and use of consultants. Response time to problems is quickened and there is some say in the future direction of the product. Their investment in thinking has saved them money. To an organization this ensures security of ownership, control. Combining multiple Free Software components and working with the community creates a

powerful synergy that has helped deliver a working broadcast solution for RFA.

REFERENCES

Open Sources
O'reilly 1st Edition January 1999
<http://www.oreilly.com/catalog/opensources/book/stallman.html>
<http://www.oreilly.com/catalog/opensources/book/perens.html>
ISBN 1-56592-582-3, 280 pages

Special thanks to the diverse staff at RFA who have contributed to this project.