NYLXS

JOURNAL SEPTEMBER 2002

The New York GNU/Linux Scene
Office Journal
Vol 1 Number 3

Editor Micheal Richardson
http://www.nylxs.com

We are Doers

Connecting Open Office to MYSQL

By Ruben Safir

One of the most important tools in office automation is the joining of common spreadsheet and word processing with powerful database capabilities. The MS Access/MS Office integration has been responsible for billions of dollars worth of product development and value added retailing other the last few years. It's an area of revenue which has not been adequately exploited using free tools.

This article is the first of a run in the NYLXS Journal which will hopefully help you fill in the gap with Free Software between you and your clients needs in regards to office automation. We will look first at the simple connectivity between Open Office and MYSQL, and then show how we can view our databases with Office tools.

Open Office's integration with any database is not as simple as it should be. This is largely due to political reasons. Within the ranks of the Open Office Database development Team are people who are working to put this together but disagree on how to proceed. Much of the database team seems more interested in fighting about the pros and cons of Postgres versus MYSQL then integrating them, and are now mulling over the possibility of adding another database tool to the package. They are also stuck on making sure that whatever they package they provide, that it works on the Windows platform. Over all, their statergy is badly lacking common sense from the prespective of the Free Software user, and the result of this is that users of Open Office are suffering under a number of bad programming and design problems.

Another impedance to Database Connectivity is that Open Office inherits much of Sun's Star Office bloodline. Star Office was developed with a 'Java First' design. The Open Office developers have tried very hard to remove as much of the Java Dependency from Open Office as possible. But the first choice for the system's database connectivity is still JDBC. In fact, the Open Office Team has even considered using a Java based database for it's packaged database design. Need I say more about this problem.

On the positive side of the ledger, nearly every free software distribution comes with a rigorous database server built right into the distribution. Either MYSQL and Postgres or both are usually installed with the core of the distribution. MYSQL is a GPL'ed program which is a rapidly maturing as a product. And is light years faster and better than MS Access, Foxpro or any of the other commonly used desktop database solutions for office automation. MYSQL is designed with true client, server capability. It has a simple learning curve, huge support, multi-threaded capability, and has a complete ANSI 92 SQL set. MYSQL is a very good choice for your offices database needs. It's very scalable, so as your office product grows in usage and deployment, a system administrator can add features and optimizations MYSQL to keep it humming with minimal trouble. It can be used to the largest scales. It now even comes with Commits and Roll Backs, one of it's largest feature holes. And depending on how you set it up, it will create roll back logs, and perform replication.

In addition to MYSQL, Postgres is also distributed very widely on with the major distributions. Postgres has many of the advanced features which one normally would expect to see with Oracle or Sybase. For those who need row level locking and the such, Postgres is a fine option. Both databases give a level of multi-user data protection unseen on the MS Windows platform.

This paper will focus on the MYSQL implementation. Much of it can also apply to Postgres.

The steps to giving MYSQL connectivity to your Open Office environment are:

1. Install Open Office
2. Install MYSQL
3. Install IODBC
4. Install MYODBC
5. Write a /etc/odbc.ini file
6. Test

Because of the limitations of hooks for the Open Office Suite, ODBC middleware needs to be used. ODBC has two parts. The server level, and the ODBC client. For this reason we need the generic ODBC libraries, iODBC, and the database specific database client, MyODBC.

PREPERATION

In the fashion of Star Wars, there is also a prequel to these steps. One should know their distribution or working environment. While installing Open Office and MYSQL by hand by tarballs is possible, one would be better off not installing Open Office from source because of it's complex dependencies. It has completely functional RPM's available. It's a complex application.

In addition, it's a good idea to know where things are stored on your system, especially system libraries and such. Here are a few things you might want to look at before preparing to make this work. Mind you that if you have a relatively new distribution, you probably already have all your ducks in order. But take a look anyway.

First, look to see if you have ODBC libraries already installed on your box. Do a '**locate**' command at the prompt. IODBC is probably installed on your system, and I will not be discussing it's installation.

You'll should see libodc or libiodbc already on your system under the /usr/lib directory. (see Illustration 1). If not, you need to download an ODBC package like iodbc from http://www.iodbc.org/software.htm. Or better, install it from your distribution disk or from your distributions download site.

In addition, now might be the time to look at your mysql libraries and configurations (illustration 2).

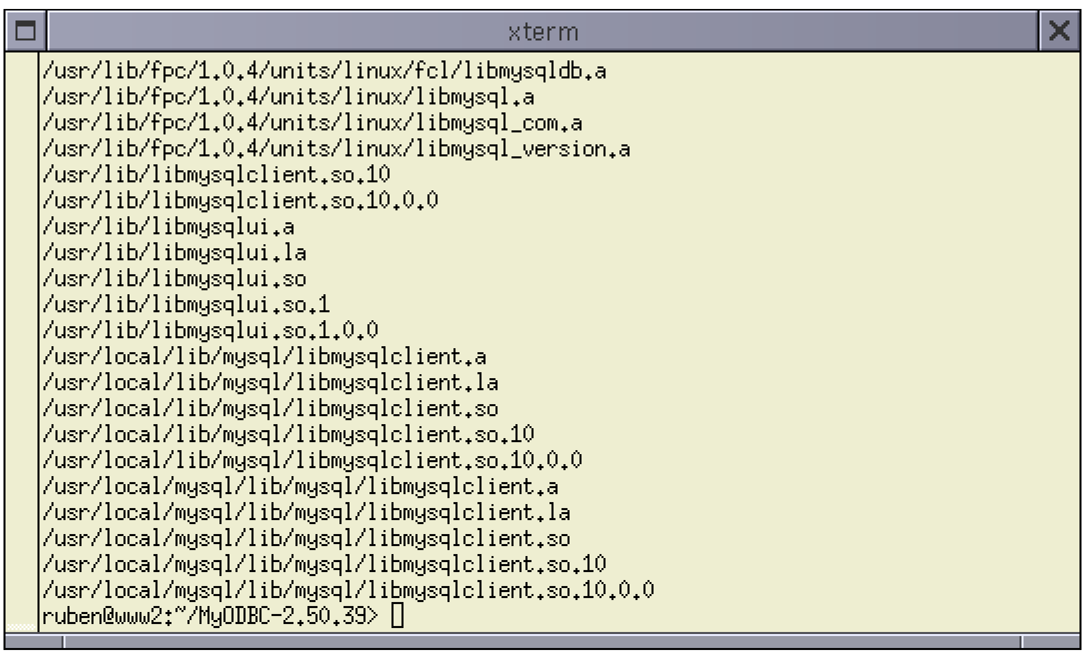


Illustration 1locate lib|grep mysql

Finally, we can download the MYSQL ODBC client from http://www.mysql.com. The client is called MyODBC-2.50.39.tar.gz and as of this writing is available from

http://www.mysql.com/downloads/download.php?file=Downloads/MyODBC/MyODBC-2.50.39.tar.gz

Drop the file MyODBC-2.50.39.tar.gz into your home directory and untar it as a regular user (not root).

ruben$: tar -xzvf MyODBC-2.50.39.tar.gz

This should create a local sub-directory ./MyODBC-2.50.39. Move into the sub-directory

cd ../MyODBC-2.50.39.tar.gz

Now look at what you got:

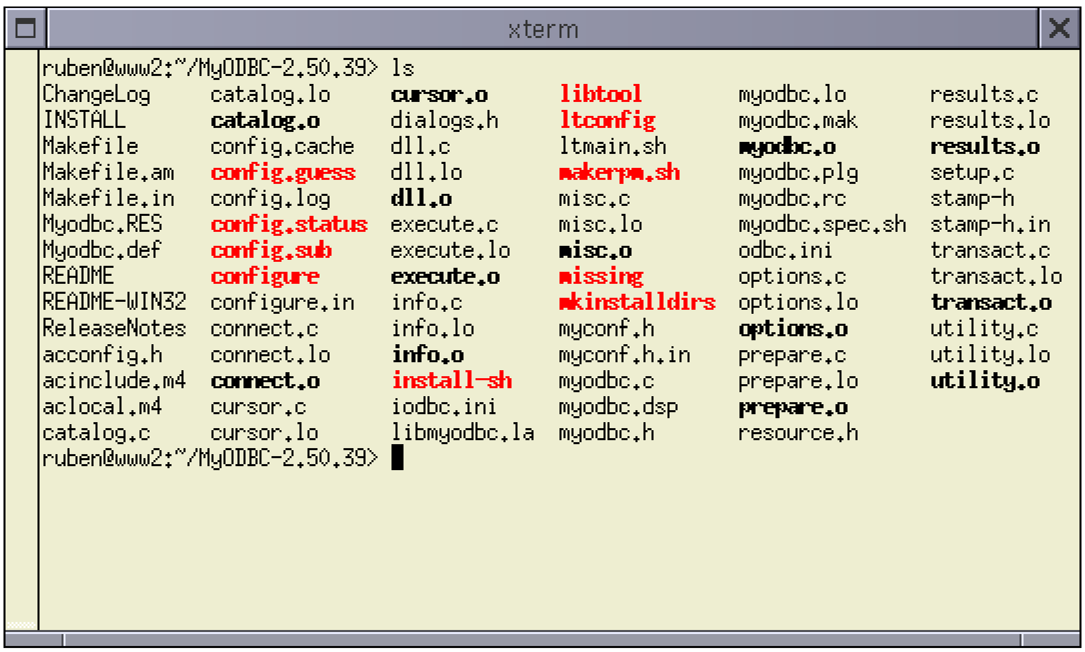


Illustration 2The untarred MyODBC directory

Read the INSTALL file, where the instructions are. Installation from source is fairly straight forward. You'll need to create an ODBC.INI file. Create a system wide one under /etc. We will be creating it later after installing MyOBDC.

Change directory to the newly created sub-directly created by your tarballs.

cd ./MyODBC-2.50.39

./configure --prefix=/usr with-iodbc=/usr

make

make install

If any of the needed libraries are not found by default, then you can specify their location on your system with the proper configure options. This is why we noted the location of the mysql libraries and the ODBC libraries before starting our compile. See the INSTALL file for more specific information.

Also try:

./configure --help

to display your options. This will output a huge number of options, but you should only need to configure those options where the default autoconf can't find the libraries on your system.

--with-iodbc=DIR

--with-iodbc-includes=INCDIR

--with-iodbc-libs=LIBDIR

--with-mysql-dirs=DIR

--with-mysql-includes=DIR

are the only ones which you normally need to be concerned with.

After we do a make, we now need to do a make install as the root user.

-ruben$: su

-ruben$: passwd:<enter your passwd>

-ruben$: make install

Finally we need to create an ODBC.ini file under the /etc. Open up the mysql client and log into your mysql database. Run 'show databases' on the client's command line. The NYLXS server has the following databases:

```
+---------------+
| Database      |
+---------------+
| addmysite     |
| apache        |
| calender      |
```

```
| cart          |
| ezpages       |
| fairuse       |
| jobs          |
| lccb          |
| lost+found    |
| mesages       |
| mike_cart     |
| mysql         |
| newschool     |
| nylsx         |
| studentgrades |
| students      |
| test          |
| test_www      |
| wireless      |
+---------------+
19 rows in set (0.00 sec)
```

As root, use vi to create your /etc/odbc.ini file. It should look something like this:

```
[NYLXS]
DSN                = NYLXS
Driver             = /usr/local/lib/libmyodbc-2.50.39.so
Database           = nylsx
Server             = www2.mrbrklyn.com
Port               = 3306

[EZPAGES]
DSN                = EZPAGES
Driver             = /usr/local/lib/libmyodbc-2.50.39.so
Database           = ezpages
Server             = www2.mrbrklyn.com
Port               = 3306
```

The strings in the square brackets can be anything as long as they are unique, The DSN can also be anything unique.

If everything is well, then we are now ready to register the MYSQL ODBC diver with Open Office.

Click on your Open Office icon on your desktop. And for the fun of it, create a new spreadsheet by clicking on File->New->Spreadsheet.

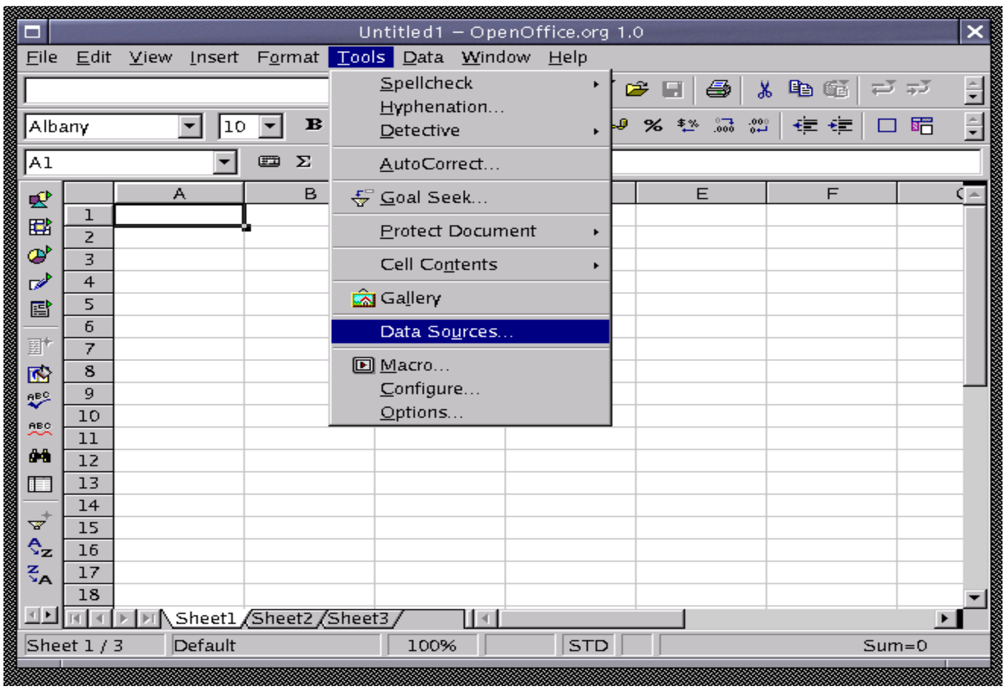
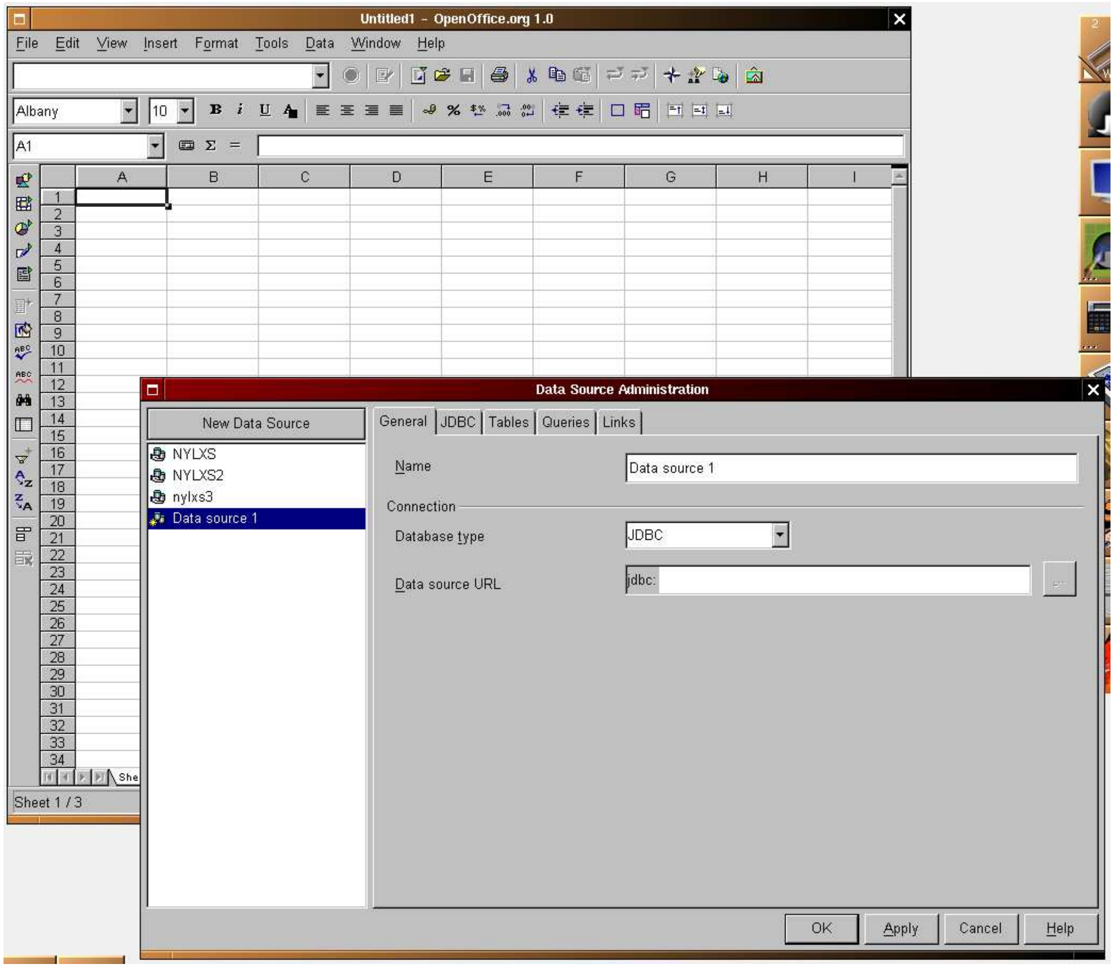


Illustration 3Adding the ODBC Data Connection

Now we can add the ODBC Driver connection to Open Office.



Select tools->datasource (see illustration 4) and open the data source GUI (see illustration 5).

Click the 'New Data Source' box on the upper right to get the data source registration GUI. Give your data source a name. This can be any string. Select the data type as ODBC. Click on the three dots and choose an ODBC data source from your ODBC.ini file. Now Click on the 'ODBC' tab on the top. Enter your user name and check the 'password required' check box. Now choose to click on either 'Tables', 'Queries' or 'Links'. An alert will pop-up asking if you want to save the settings. Say yes and continue. To keep this simple, we will use a table (and clicked on the 'Table' tab).

We are now prompted for our password. Enter your mysql password and check the box asking to cache the password until the end of the session. You should now see all your available tables for this database.

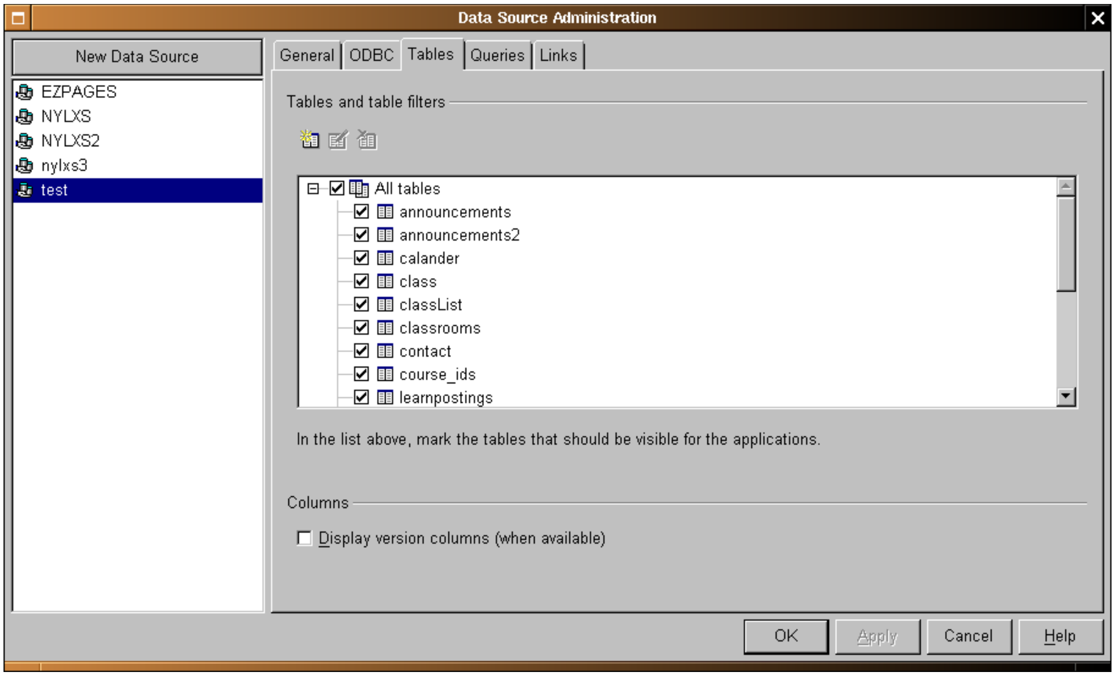


Illustration 4 Available Tables

These tables are the ones you can make available to Open Office Applications. Now click OK and you are set to put live data from your database into your Open Office Applications. One word of warning: ODBC is fat and slow. You might want to lower the priority of Open Office when using it with large databases.

Adding Database information automatically into a Document

Let's now see how we can put data into an application. Open a spreadsheet, or use the one we have opened. Click on Data->Datapilot:

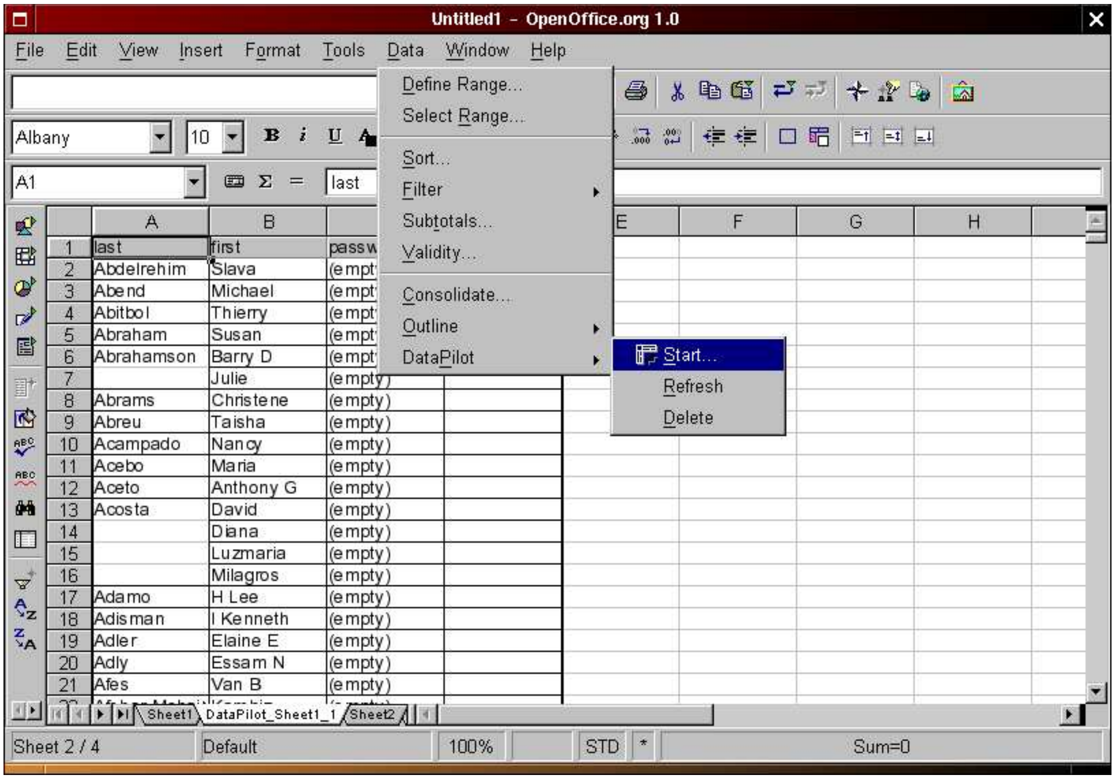
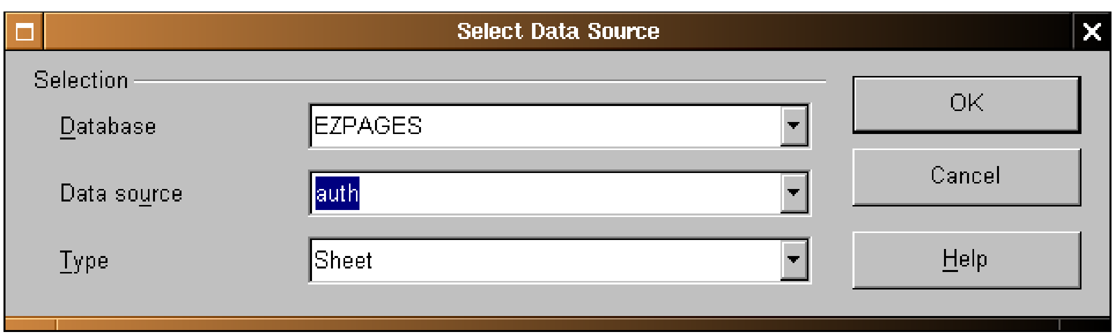
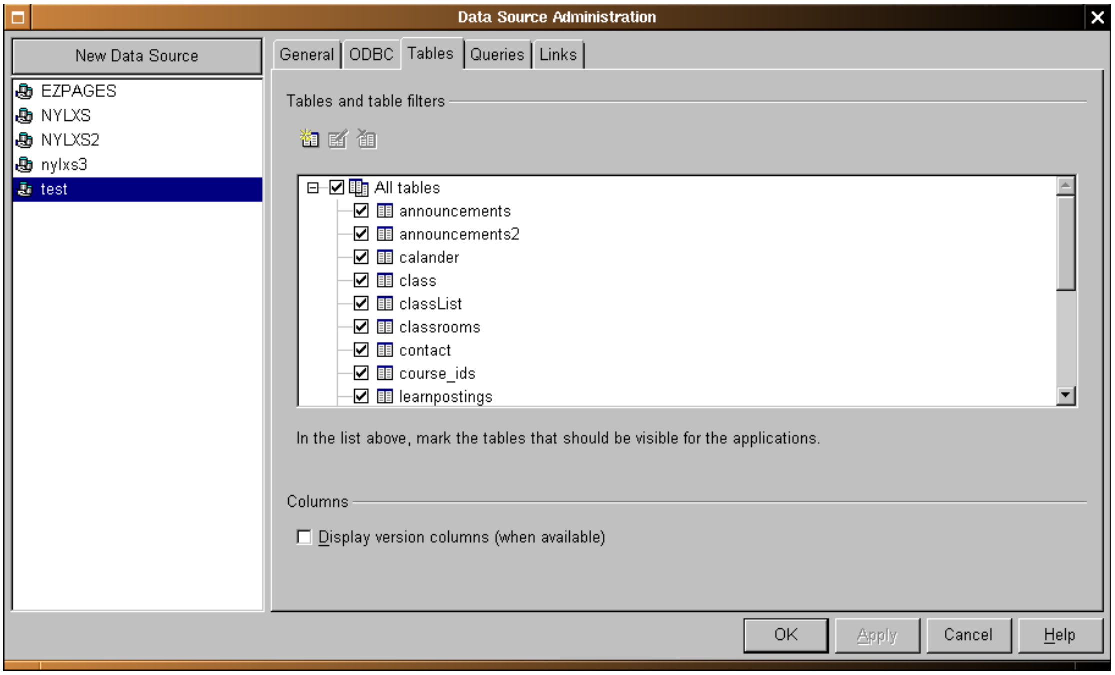


Illustration 5 DataPilot

Click on Start. Select 'Data Source registered in Open Office', which should be the one we just created. Click OK and choose the data source:



Click 'OK' again after choosing a table listed under 'Data source'. Select 'sheet' to enter our data into a new sheet within our spreadsheet. If this is a large table, this might take a few seconds. Drag and Drop your database columns. And Click OK. Now your Spreadsheet is talking to the database. Wait a moment while the datapilot does it's work.



Voilia – You have your database connected to your spreadsheet.!

New Yorkers for Fair use/NYLXS Washington Report

US Department of Commerce DRM Workshop Political Action

July 17th, 2002

On July 17th, 2002, New Yorkers for Fair Use and NYLXS, with help from the Free Software Foundation, held a political action at the Department of Commerce where they were holding their second Digital Rights Management (DRM) Workshop. New Yorkers for Fair Use spearheaded the efforts to bring public opinion to the meeting in a reasonable and effective manner. New Yorkers for Fair Use is happy to announce that all of our goals were met in this action, and even exceeded, without disrupting the normal activities of the meeting. It was our goal to have a voice and participate in the Democratic process, and not to disrupt entirely the process.

Prior to the meeting in Washington DC, the President of NYLXS and co-founder of NY Fair Use, Ruben Safir, was attending at the request of the French Free Software movement, the Libre Software Meeting in Bordeuax France. Upon arriving to France, Ruben checked his email and received from the NY Fair Use Secretary, Seth Johnson, a message outlining the Commerce Departments announcement with a list of participants, and asking if NY Fair Use wanted to attend the meeting, scheduled to happen the day after Ruben's arrival back in New York the next week. Ruben was reluctant to schedule a trip to Washington DC so soon after arriving home, but the meeting appeared to be too good of a target for the NY Fair use agenda to pass up. It would give us an opportunity to speak face to face with most of our chief opponents including the likeable and articulate Jack Valenti of the MPAA, the RIAA, and several industrial leaders including Microsoft, and Intel. So Ruben sent a reply back to Seth to prepare for the trip and assigned him the duty of preparing everyone for the trip.

Once the decision was made within the NY Fair Use management to go ahead with the plan, several goals were agreed upon by the management including Ruben, Seth, Co-founder of NY Fair Use Brett Wynkoop, and Jay Sulzberger our general public relations manager. NYLXS also geared into action providing network services and funding for the trip. NYLXS members, as usually, sprang into action in support of the NY Fair Use activity. In particular, Micheal Richardson, the NYLXS Membership Chairman and Journal Editor designed buttons and helped drive everyone down to Washington, Kevin Mark contributed to the PR material, Vinnie alerted as much of the press as possible and Joe Maffia offered technical support. Even with large parts of NYLXS on vacation, the organization worked admirably according to it's charter, to support and educate the public about issues which affect Free Software in business, education, and in the home.

The first agenda item by NY Fair Use was to get us representation on the panel, specifically by including Ruben Safir as a panel representative. Although Ruben was in France and cut off from many of the Washington connections which have been nurtured over the last few years, NY Fair Use was petitioning for a place on the panel from the start. We had Sarah Brown of the EFF in Washington who put Seth Johnson in contact with Chris Israel inside the Commerce Department. Seth Johnson also tried to contact Congressman Weiners Office in an effort to bring some Congressional pressure on the matter. But with the limited time frame and our chief advocate in Europe, we were not able to get representation on the board. Ultimately, NY Fairuse had to fall back to our secondary plan, to participate as members of the audience.

Limited to audience participation, we now opened the discussion on a new mailing list created by Seth for the purposes of organizing this action in Washington. The first order of business was to produce proper slogans for our goals. One goal of NYLXS is to change the whole lexicon of the DRM and Copyright discussion. We felt that certain messages needed to be brought to the public and the press to assure our future success, not just in this battle over DRM, but also in a wide variety of copyright and fair use issues as they have trickled into the public eye over the last few years. We decided, after much discussion and after considering many opinions on the mailing lists, to attempt to drive into the public lexicon the phases, “DRM is Theft” and “We are the Stakeholders”. We chose these expressions very carefully to counter the rhetoric coming out of the copyright monopoly content industry, especially the claims by Senator Hollings that he had assembled all the "Stakeholders" to write his CBDTPA bill, and Jack Valenti's rhetoric that the simple act of listening to a DVD on a GNU/Linux operating system is stealing property from the motion picture industry.

The next stage of planning fell into the lap of Seth Johnson, who mobilized NYLXS and NY Fair Use for the practical matters of making the trip. Fortunately, NYLXS is today a well oiled machine. We are very proud of the volunteer spirit of the organization and it's ability to deliver when called upon. Joe Maffia researched our previous work in find a minibus for the trip down. Seth Johnson offered use of his credit cards to get the truck. Michael Richardson volunteered to drive everyone to DC. Brett Wynkoop oversaw all the system administrations and co-ordination duties needed to keep everyone in touch. Jay Sulzberger wrote up a terrific position paper for the trip and a media guid. Vinnie contacted nearly every press contact he could find with information about the trip. Members with almost every political background came from every part of the city to volunteer to make the trip. NYLXS has become very effective at organizing events. They are motivated, proactive, and competitant.

Meanwhile in France, Ruben Safir and Richard Stallman, over much wine, reviewed the political strategy for the action. The principals agreed to a joint effort of the FSF and NY, and even planned on backup contingencies including how to hold a protest outside of the building if we were eventually not allowed into the conference room. Some of the ideas we knocked around was showing up with steaks, as in meat, to drive home the message that the public is the stakeholders on copyright monopoly matters, or dessing in costumes. However time would not allow us to co-ordinate these efforts. They are still on the table for future political action.

On Richard and Ruben's arrival home to the States, all the pieces were in place for a successful run at changing the direction of the DRM discussion.

The afternoon Ruben came home to New York, Seth and Ruben touched base and made certain that the arrangements for the van was in place, and that everyone was clear about the time and place we were to meet. We agreed to meet early in the afternoon the day Ruben arrived home, and the day before going to Washington, to pick up the truck from another neighborhood in Brooklyn across town. Seth, Micheal and Ruben went out to rent the van, and drove it home. We decided to meet at Ruben's house the night before for a final planning session Since we were leaving at 4AM in the morning to get to DC on time for a meeting with the media at 10AM, everyone assembled very early in Flatbush. Brett left with his own care from Park Slope in Brooklyn and we stayed in touch on the road. In the minibus we had Michael Richardson, Jay Sulzberger, Seth Johnson, Vinnie, Forrest Mars, Murray, Kevin Marks, and Ruben. Nearly everyone was dressed in a suit and tie.

The drive down to Washington went smoothly and we met with media guru Eric Hensal at the Corner Bakery in the Press Club building a few blocks from the Commerce Department Building. Eric gave NY Fair Use great media tips on how to get seen by the press, in addition to a packet with vital press contacts both in the building and across town. Forrest Mars ran our Media Alert through the building, while Richard Stallman met with us at the bakery with copies of the flyers. Richard and Ruben had spent most of the night on the phone and through mail dotting the i's and crossing the t's on the press release. Vinnie ran off more copies. Brett and Kevin ran down to the Commerce Department Building to scout out the conditions. As it turned out, this was critical because the Department of Commerce was convinced that they could close the meeting to the public. Brett's charm and force of rhetoric convinced the panel that they couldn't close the workshop, and by the time that we were ready to actually go to the meeting, the Panel had rolled out the red carpet for NY Fair Use, giving us an escort to upstairs to the 4th Floor.

Finally, at about 12:30, Ruben, Richard, Seth and Kevin marched to the Commerce Department building, with the rest following later. Ruben handed out our flyers, and met with several friends in the press and with other organizations who are aligned with us. We met with Declan McCullagh from

CNN Net, and Robin Gross of the EFF. We also met with American Library Association Representative, Miriam Nisbet and told them of our efforts to protect Libraries last year with our save the Libraries Campaign. People lined up to meet Richard Stallman before the meeting, and eventually the rest of our group arrived at the meeting, filling up the room.

We also had the pleasure of seeing many old friends from the Press including Seattle reporter Sarah Strickland and Bloomberg reporter Katherine Reynolds Lewis. As the meeting started, everyone in our group was wearing the “DRM is Theft” Buttons, and the “We are the Stakeholders” Buttons. In fact, we almost ran out of buttons.

It was not the plan of NY Fair Use in any way to disrupt the Workshop Panel. Neither was it the goal of the organization to sit mute in the audience and not be felt. Our goal was to clearly participate as audience members of the panel. In this matter, and within the bounds of normal political discourse as is the rule with such contentious issues. For example, as the meeting started, the Chair began by saying that he'd like to announce that nobody on the panel is a villain. This brought audible laughter from the audience. We were then quiet for the next hour. At one point, Brett Wynkoop, who couldn't find a chair and finally sat near the panel's table, raised his hand when the Chair asked if anyone else had a comment. The Chair recognized Brett, who then proceeded to announce his name and position as co-founder of NY Fair Use. The Chair tried to un-recognize him, but Brett just plowed forward, within the bounds of Rogers Rules of Order, by asking the panel how they could consider regulations and laws which would turn every teenager in America with a Magic Marker or Wite-Out, into a felon. Brett was refering to the recent flap over the attempt by the music industry to produce DRM for audio CD's which was quickly defeated with a single line made by a marker. Several minutes of interesting debate followed which finally climaxed when one of the lawyers on the panel gave the legal opinion that the courts would never convict anyone for using a Magic Marker as a circumvention device because the courts had not considered that a sufficient circumvention method under the DMCA. This caused Ruben to ask if we can get that in writing for the next DMCA trial dealing with digital music. Everyone was jovial at this point, laughing and smiling and, enjoying the open participatory democratic process unfolding before their eyes. Only the most pretentious of individuals in the crowd refused to smile and participate.

Earlier, a schematic diagram was given of the current state of DRM development. This alphabet soup of circles and letters were discoursed upon for several minutes; the thing looked looked similar to a Network Map of the internet. Nobody could understand a word of what was being presented, but everyone sat quietly and listened anyway. At the end of this part of the representation, the image on the projector was flipped, to connect the dots in complete chaos. This fitting image of the state of current DRM schemes would come back to haunt the panel as the presentation went on.

NY Fair Use mostly sat quietly through the meeting, occasionally whispering to each other, until MPAA spokesman Jack Valenti took the floor. At this point, Jack elegantly said in a beautiful self-effacing fashion, that he hopes that everyone on the panel would give him the courtesy to be heard and then he would yield the floor to others. At this moment, Vinnie stood up and said, “How can you expect everyone to present their response to your comments when you've left off the panel the most important stakeholders, the public.”. Jack then graciously said that if we let him speak without interruption (not that we seriously interrupted anyone), that he would be glad to listen to Vinnie's reply. Jay Sulzberger stood up and said we all agree, and the Chair was feeling a little uncomfortable at this point, feeling he was losing control. But everyone sat silently and heard Jack speak. He put on a classic Jack Valenti performance, and said that it was his position that Government intervention in this matter wasn't a bad thing. And that his experience in the Johnson Administration passing the 1965 Civil Rights Act showed him how important and good proper Government intervention can be. He then continued by saying in his charming Texas drawl, that it was his hope that the leaders in the IT industry and the computer field would come to a consensus in the next month on a DRM standard which would protect the property of the Movie Industry from theft. This caused Ruben to whisper to Vinnie that when the floor returned to him, he should defer comment to Richard Stallman, who was the most qualified and highest ranking IT professional in the building.

When Jack finished, the Chair reluctantly deferred to Jack's wishes and gave the floor to Vinnie. Vinnie then identified himself as a member of NY Fair Use and asked to defer the answer the Richard. Richard stood up, but the Chair didn't let him talk, in violation of the rules of order, and instead said, We've already accidentely added one of your members to the panel, and he saw no reason why we should get another member to give their opinion. This caused the audience to get visibly upset. In response, NYLXS President Ruben Safir intervened and asked the Chair, “Pardon me, Mr Chair, however, Mr. Valenti nicely pleaded that the leaders of the IT industry to come to a consensus on DRM. However, one of the greatest figures in the Computing Field was standing right here to the left, as part of NY Fair Use. Richard Stallman has just been given an award in France by the United Nations for his contributions to World Heritage with the invention and development of the GNU/Linux system and it's variants. He's the most qualified person in the room to make a public comment in response to Mr Valenti's request.” Ruben's intervention at this point quieted the crowd as nobody wanted things to break down into a raucous confrontation. Though Richard was still denied a chance to speak, our discipline assured that the meeting could continue without derailing the workshop.

However, much of the conversation from that point forward was affected by the events. The Digital Recording Rights Coalition presented more forcefully their position that DRM eroded Fair Use. Jack, in order to convey just how serious the MPAA is about getting DRM enacted quickly, said that while the MPAA responded to a letter from Microsoft about progress toward DRM in 24 hours, that when the MPAA sent such a letter to Microsoft, Microsoft took a long time to return a response. Microsoft at this point all but threatened to buy all the Movie Producers if they continued to be such a pain in the neck. Although this was not their exact words, their threat was neither veiled or lost on Mr Valenti.

Meanwhile, Jack tried to persuade the panel that the Movie Industry had never really been against the VCR. This caused some agitation of panelists, and the crowd just laughed. The panel pointed out that despite the Movie Industry's professed love for the VCR, they brought an injunction against panel members whose companies made VCR, which injunction was eventually defeated in the supreme court.

But Jack was not the only Panel member capable of bald faced lies. EMI tried to convince everyone that artists really don't hate their record labels, but only say so in public because it's good PR. And later they claimed that the record labels weren't responsible for preventing music from being available, but that it was the artist's fault. AOL Time Warner delightfully wants to close the analogue hole. And a bunch of other positions were espoused, many of which have been covered by Slashdot Ad Nauseum and need not be repeated here.

At the end of the session, everyone had a chance on the panel to express themselves exactly as they wished to. And when it finished, Jay announced that NY Fair Use was to have a press conference in front of the Commerce Department Building at 4:30. Ruben gave his press conference with the help of Richard and Jay. We talked extensively with Bloomberg and other reporters. We announced our position and we announced our sponsorship of the NY Fair Use “Fair Use Bill”.

NY Fair Use attained almost all of our goals for this action. The phrases, “We are the stakeholders” and “DRM is theft” have begun to make their way into the press. We are being invited to a new panel on DRM for consumers and we are debating the merits of this now. We are leery of the formation of another panel and question whether is is a stalling tactic. And we reiterated our position that NY Fair Use wants a seat with the original panel. Capital Hill was abuzz the next day with our activities. And Congressman Weiner's Office has promised to led it's efforts to bring us into the panel in the future.

NY for Fair Use has kicked down the door that everyone else will now run through,

You are welcome.....

Ruben Safir
Co-founder of NY Fair Use.
President NYLXS.

US CONSTITUTION:

Amendment IV

The right of the people to be secure in their persons, houses, papers, and effects, against unreasonable searches and seizures, shall not be violated, and no warrants shall issue, but upon probable cause, supported by oath or affirmation, and particularly describing the place to be searched, and the persons or things to be seized.

Amendment V

No person shall be held to answer for a capital, or otherwise infamous crime, unless on a presentment or indictment of a grand jury, except in cases arising in the land or naval forces, or in the militia, when in actual service in time of war or public danger; nor shall any person be subject for the same offense to be twice put in jeopardy of life or limb; nor shall be compelled in any criminal case to be a witness against himself, nor be deprived of life, liberty, or property, without due process of law**; nor shall private property be taken for public use, without just compensation.**

Article 1:

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;

Who Owns That Downloaded File Anyway?

We are the Stakeholders

It's more than a slogan. It's a fact, and one which the public demands Congress to recognize. There it is. The rights of every individual citizen, clearly stated in the US Constitution in English as plain to understand today as it was when it was written over 200 years ago. Individuals are guaranteed to right to be secure in their persons, houses, papers and effects. This is absolute and the foundation of Freedom. We can not have a free people if we are subject to inspection, wiretapping, bullying and preemptive control of our digital communication. We can not tolerate this any more than we can tolerate wiretapping of our phone lines, spying on our private letters, or burning of our books. It doesn't matter if the public chooses to share music files or movies on a massive scale, individual to individual. All that matters is that we are protected from corporations and the government from doing so.

This summer the Music, Software and Movie Industry has pressed both the executive branch and the legislative branches of our government to allow spying, preemptive filtering, and commandeering of personal property in order to prevent the common distribution of information which is protected by copyright. Anyone who reads the plain meaning of the constitution can clearly see that these copyrighted articles are owned by the public. Congress is only authorized to give limited rights to creators of works ***in order to promote the publication of more on them***. And then, only if they are useful. We fully demand protection from those who scam the public, and steal from us our cultural heritage.

Who owns copyrighted works? It is the public. Tomorrow Congress can end all copyright protections, and then there is none. But Congress can never pass a law which ends our ownership of our papers, books and property. We demand the same protection to share information on our hard drives, on CD's, Vinyl, Tape and paper that is afforded used books.

On September 26th, 2002, in the House Sub-Committee on Intellectual Property, while reviewing HR 5211, Congressman Berman publicly announced that nobody is talking about preventing the sharing of a few files between individuals.

Evidently, Congressman Howard Berman seems to be misinformed. The Recording Industry Association of America clearly says that they disagree. Their statements on http://www.soundbyting.com/html/copyright_101/music_copyright.html clearly states:

> "So the only LEGAL way to reproduce a piece of recorded music -- uploading, downloading, copying from a CD, whatever - is to get permission from the owners of these different copyrights. It's called, obtaining a license"

This is just wrong. Our homes are not businesses open to in discriminant industrial policies. Our homes are private. We do not license our music and our computers. We purchases them for cash. And we will use them as we choose to, and we will share information with others without prior constraint, and of our own free will. We are not pirates. We are voters, and **WE ARE THE STAKEHOLDERS.**

In our private lives, we will share what we choose to, we will educate as we see fit, and we will build libraries for public access and to help educate and provide for our children and the poor. And we will not have the public treated as robbers and criminals.

If the RIAA sees a legitimate copyright violation, we are protected from their gross exaggeration of their legal rights by the fourth and fifth amendments in the Bill of Rights. If the legalities of prosecuting individuals under the law is not convenient for the RIAA, we do not authorize a digital police state instead. We will not walk down the path of George Orwell's 1984, where computers are used to control the the thoughts and activities of the population. We will not allow our property to be continued to be described as the property of a Copyright Holders.

Finally, the public is tired of the foot dragging by the Movie and Recording industry in publishing materials electronically. We demand Congress force the Copyright Holders to provide their end of their bargain in Copyright. Congress should move immediately to force the media companies to publish music on-line. Use forced licensing if you must, but get our cultural heritage on the internet. The Music industry, especially, has impeded innovation and job creation because of it's refusal to publish. Nobody has the 'right' not to publish, once they have been given a copyright and made works available. The NY Times reporter, Amy Harmon, wrote on September 23rd:
(http://www.nytimes.com/2002/09/23/technology/23MUSI.html?pagewanted=print&position=top)

> Concerns over piracy, money or unrelated contract disputes have prompted artists like Madonna and Radiohead to insist that their music not be distributed digitally. And even if the artist and the label are on board, the publisher who represents the writer of a song may not be. Sometimes it takes months to figure out who the publisher is, since there are more than 30,000 of them in the United States and their names are often not included on the CD.
>
> "It's as if Franz Kafka designed this system and employed Rube Goldberg as his architect," said Rob Glaser, the chairman of MusicNet, which is part-owned by his company, RealNetworks, along with AOL Time Warner, Bertelsmann and the EMI Group. "It's full of tripwires."
>
> The most glaring omissions from the services are the entire catalogs of major labels that have so far declined to license to the services backed by their offline competitors. The Warner Music Group, the music division of AOL, and the BMG unit

> of Bertelsmann have yet to license their music to Pressplay. Likewise, Pressplay's owners, the Universal Music Group of Vivendi Universal and Sony Music Entertainment, have not licensed their recordings to MusicNet. The EMI Group, the smallest of the five major labels, has licensed to both MusicNet and Pressplay. And Listen.com, which has no record labels among its corporate parents, has licenses from all five major labels.

It is time for Congress to step in and make the copyright holders publish on the internet. We don't care is Madonna makes money or not. We care that works are made available to the public. We give publishers narrow, but well defined exclusive rights to profit from by distributing copies of copyrighted work. We give these rights in exchange for making them public. Their 'rights' are not the inalienable rights, such as those defined in the Bill of Rights, but subject to the will of the People. The People demand that they publish, or otherwise strip them of their copyrights all together.

The Use of CVS on Free Operating Systems

as presented by Ari Jort

NYLXS In Service Presentation:

September 2002

Why CVS; Why is it important to discuss CVS Usage?

CVS is the most commonly used Revision Control Software used with Free Software today. With an eye towards collaboration, CVS allows multiples of developers to work on and maintain software and documents. CVS allows developers world wide to work on the same project and is key to leveraging the internet as a tool for remote development.

CVS is most useful for Free Software Projects which have programmers on several remote locations. Under this condition, an easy way to merge multiple changes and patches for whole development trees becomes paramount for developing a usable program and code base. In addition, CVS becomes a primary distribution mechanism for end users to get updates and new fixes for their programs and code.

CVS is underutilized for a number of reasons. CVS takes a small learning curve to use correctly, and often developers don't take the time to learn it's proper use. In addition, many projects are started without the intent for collaboration, only later needing to play catch-up. More to the point, developers don't always make for good project managers, and often overlook the importance of version control.

What are the basic building blocks to CVS?:

Repository and Working Environments:

CVS has a client server model. Although CVS was not originally developed with a client server model, it has grown to be dependent on this ability. Unfortunately, the model is not implemented well, as can be by the source code and the interface. Never the less, despite its shortcomings, CVS has proven to be useful and vital. Much of CVS is inherited from RCS. While we will not discuss RCS, we might note that RCS allows for file versioning, logging and patching of files.

A CVS repository is a location on a computer where all versions of all the files to a project exist, including the logs of all the changes to the files.

A CVS working environment is a snap shot of a CVS repository version, with possible alterations, on either a remote workstation or in another location on the computer which holds the repository.

When work is finished on the Working environment, it is sent back to the repository. Working environments and Repositories can interact in a number of ways. An 'update' for example, is a call to the repository for just patches. Branching is creating a space where work is done from the main line of development. Mergers join together new work to a branch or the main development line.

Every Branch has a version number. By default a version starts with 1.1 on check in. This versioning system gives us a major and minor number. For example the version number 1.12 would be a branch. If we create a new branch, we would get version 1.12.1.1. The 1.12.1 consists of the new branch and 1.12 is the trunk. The last .1 is the increment.

CVS also has the very useful facility called tags. Tags allow you to create a selectable label for sets of files and versions. By convention, tags use all capital letters for their easy identification. The default tag is HEAD.

Creating a CVS Environment:

There is 3 steps to creating a CVS environment:

A - Create an initial CVS Root Path:

Create a directory for your CVS repository. Traditionally this is called CVS root.

su
mkdir /usr/local/cvsroot
chmod 777 /usr/local/cvsroot

It is important for this path to be defined in the environmental variables of your shell. You can do this in ~/.bashrc or ./.profile with the bash shell.

Add the following line in your startup script:

export CVSROOT=/usr/local/cvsroot

change to the root user and create your CVSROOT directory:

Not run the cvs init command:

cvs init

or

cvs init -d /path/to/cvsroot

if you wish to not use the CVSROOT environmental variable.

B- Populate the repository

Before Populating your repository, make sure your shell environment is correct:

CVSROOT should be set to the proper path. If remote access is desired, make sure CVSROOT_RSH is set to ssh for secure communications:

CVSROOT_RSH=ssh

Server roots can be remote:

CVSROOT = www.nylxs.com:/usr/local/cvsroot
l environment is
correct:

CVSROOT should be set to the proper path. If remote access is desired, make sure CVSROOT_RSH is set to ssh for secure communications:

CVSROOT_RSH=ssh

Server roots can be remote:

CVSROOT = user@www.nylxs.com:/usr/local/cvsroot

Now you can populate your repository by using the cvs import command.

for example:
In a source code tree of /home/user/backup_prog/ we run the command:

cvs -t import BACKUP enterprise 'R&M' 'GTK_BACKUP'

This imports the BACKUP repository with the 'R&M' vendor tag and
'GTK_BACKUP'
release tag. All the files in /home/user/back_prog/ are imported into the repository directory.

Once your environment is set, you need to go to the directory where the files you want moved into the repository are located.

Now we are ready to create a working copy of the CVS repository. This working copy is the backbone of your working environment:

```
$: mkdir ~/src/project
$: cd ~/src/project
$: cvs -t co BACKUP
```

Now check out your new source code tree.

```
$: cd ./BACKUP/
$: ls
CVS      backprog.c  errors  rmbackup    tarlib.c
Makefile  backprog.h  main    rmbackup.c  tarlib.h
```

Once you have your source code tree, it might be necessary from time to time to update the source code from the repository. Remember that several people can be working on the code simultaneously. bug fixes and security updates are common. If you use 'update', you only get patches.

```
$cvs -t update
enterprise:ruben:backprog2: $ cvs -t update
 -> main loop with CVSROOT=/usr/local/cvsroot
cvs update: Updating BACKUP
 -> rename(backprog.c,CVS/,,backprog.c)
 -> checkout (/usr/local/cvsroot/BACKUP/backprog.c,v, 1.2, , backprog.c)
 -> chmod(backprog.c,100664)
 -> Register(backprog.c, 1.2, Wed Oct  9 20:11:31 2002, ,  )
 -> fopen(/usr/local/cvsroot/CVSROOT/history,a)
U BACKUP/backprog.c
-> unlink_file_dir(CVS/,,backprog.c)
 -> rename(CVS/Entries.Backup,CVS/Entries)
 -> unlink_file(CVS/Entries.Log)
```

This added a patch to backprog.c from the repository.

While a checkout brings files from the repository to your work area, the commit command places your changes into the repository. You can commit a single file, or a whole environment. Let's first change a file in our source tree.

```
$: vim ./backprog.c

#include <stdio.h>
#include <gtk/gtk.h>
#include "backprog.h"
#include "tarlib.h"

/* This makes Toggles */
/* This is our change to CVS - maketoggles make toggle buttons for our */
/ * application */

struct s1 maketoggles(){
        static struct s1 handle;
        GtkWidget *button;
        GtkWidget *vbox;
        GtkWidget *label;

        vbox = gtk_vbox_new(TRUE, 0);

        label = gtk_label_new("Choose an Option");

        gtk_widget_show(label);


...

:wq
```

```
$: cvs -t commit
Added some useful documentation

CVS: ----------------------------------------------------------------------
CVS: Enter Log.  Lines beginning with `CVS:' are removed automatically
CVS:
CVS: Committing in .
CVS:
CVS: Modified Files:
CVS:    backprog.c
CVS: ----------------------------------------------------------------------
~
~

$: cvs -t commit
 -> main loop with CVSROOT=/usr/local/cvsroot
cvs commit: Examining .
 -> checkout (/usr/local/cvsroot/BACKUP/backprog.c,v, 1.2, , (function))
 -> ParseInfo(/usr/local/cvsroot/CVSROOT/commitinfo, BACKUP, ALL)
 -> ParseInfo(/usr/local/cvsroot/CVSROOT/rcsinfo, BACKUP, ALL)
 -> ParseInfo(/usr/local/cvsroot/CVSROOT/editinfo, BACKUP, not ALL)
 -> system('vi' '/tmp/cvszhcteG')
 -> unlink_file(/tmp/cvszhcteG)
 -> ParseInfo(/usr/local/cvsroot/CVSROOT/verifymsg, BACKUP, not ALL)
 -> unlink_file(/tmp/cvsqFJWwf)
Checking in backprog.c;
/usr/local/cvsroot/BACKUP/backprog.c,v  <--  backprog.c
 -> checkout (/usr/local/cvsroot/BACKUP/backprog.c,v, 1.2, -ko, /tmp/cvsvG5tAQ)
new revision: 1.3; previous revision: 1.2
 ->
rename(/usr/local/cvsroot/BACKUP/,backprog.c,,/usr/local/cvsroot/BACKUP/back
prog.c,v)
 -> unlink_file(/tmp/cvsvG5tAQ)
 -> unlink_file(/tmp/cvsOwAEzr)
done
```

```
-> checkout (/usr/local/cvsroot/BACKUP/backprog.c,v, , , (function))
-> Register(backprog.c, 1.3, Wed Oct  9 20:24:56 2002, ,  )
-> fopen(/usr/local/cvsroot/CVSROOT/history,a)
-> unlink_file(CVS/Base/backprog.c)
-> ParseInfo(/usr/local/cvsroot/CVSROOT/loginfo, BACKUP, ALL)
-> rename(CVS/Entries.Backup,CVS/Entries)
-> unlink_file(CVS/Entries.Log)
```

Now we can check the status of our updated repository:

```
$: cvs status
....

===================================================================
File: backprog.c        Status: Up-to-date

   Working revision:    1.3     Wed Oct  9 20:26:29 2002
   Repository revision: 1.3     /usr/local/cvsroot/BACKUP/backprog.c,v
   Sticky Tag:          (none)
   Sticky Date:         (none)
   Sticky Options:      (none)

===================================================================
File: backprog.h        Status: Up-to-date

   Working revision:    1.1.1.1 Wed Oct  9 15:23:24 2002
   Repository revision: 1.1.1.1 /usr/local/cvsroot/BACKUP/backprog.h,v
   Sticky Tag:          (none)
   Sticky Date:         (none)
   Sticky Options:      (none)

===================================================================
```

```
======
....
```

Or we can review the log of changes to the repository.

```
RCS file: /usr/local/cvsroot/BACKUP/Makefile,v
Working file: Makefile
head: 1.1
branch: 1.1.1
locks: strict
access list:
symbolic names:
        GTK_BACKUP: 1.1.1.1
        R&M: 1.1.1.1
        enterprise: 1.1.1
keyword substitution: kv
total revisions: 2;     selected revisions: 2
description:
----------------------------
revision 1.1
date: 2002/10/09 15:23:23;  author: ruben;  state: Exp;
branches:  1.1.1;
Initial revision
----------------------------
revision 1.1.1.1
date: 2002/10/09 15:23:23;  author: ruben;  state: Exp;  lines: +0 -0
This is the Rosenzweig and Maffia Backup Program

=============================================================
================

RCS file: /usr/local/cvsroot/BACKUP/backprog.c,v
Working file: backprog.c
head: 1.3
```

```
branch:
locks: strict
access list:
symbolic names:
        GTK_BACKUP: 1.1.1.1
        R&M: 1.1.1.1
        enterprise: 1.1.1
keyword substitution: kv
total revisions: 4;     selected revisions: 4
description:
----------------------------
revision 1.3
date: 2002/10/09 20:26:29;  author: ruben;  state: Exp;  lines: +3 -1
Added some useful documentation
----------------------------
revision 1.2
date: 2002/10/09 20:09:59;  author: ruben;  state: Exp;  lines: +1 -1
Add some comments to backprog.c
----------------------------
revision 1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;
branches:  1.1.1;
Initial revision
----------------------------
revision 1.1.1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;  lines: +0 -0
This is the Rosenzweig and Maffia Backup Program

=============================================================
================

RCS file: /usr/local/cvsroot/BACKUP/backprog.h,v
Working file: backprog.h
head: 1.1
branch: 1.1.1
```

```
locks: strict
access list:
symbolic names:
        GTK_BACKUP: 1.1.1.1
        R&M: 1.1.1.1
        enterprise: 1.1.1
keyword substitution: kv
total revisions: 2;     selected revisions: 2
description:
----------------------------
revision 1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;
branches:  1.1.1;
Initial revision
----------------------------
revision 1.1.1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;  lines: +0 -0
This is the Rosenzweig and Maffia Backup Program

=============================================================
================

RCS file: /usr/local/cvsroot/BACKUP/errors,v
Working file: errors
head: 1.1
branch: 1.1.1
locks: strict
access list:
symbolic names:
        GTK_BACKUP: 1.1.1.1
        R&M: 1.1.1.1
        enterprise: 1.1.1
keyword substitution: kv
total revisions: 2;     selected revisions: 2
description:
```

```
----------------------------
revision 1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;
branches:  1.1.1;
Initial revision
----------------------------
revision 1.1.1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;  lines: +0 -0
This is the Rosenzweig and Maffia Backup Program

=============================================================================

RCS file: /usr/local/cvsroot/BACKUP/main,v
Working file: main
head: 1.1
branch: 1.1.1
locks: strict
access list:
symbolic names:
        GTK_BACKUP: 1.1.1.1
        R&M: 1.1.1.1
        enterprise: 1.1.1
keyword substitution: kv
total revisions: 2;     selected revisions: 2
description:
----------------------------
revision 1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;
branches:  1.1.1;
Initial revision
----------------------------
revision 1.1.1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;  lines: +0 -0
This is the Rosenzweig and Maffia Backup Program
```

```
=============================================================================
RCS file: /usr/local/cvsroot/BACKUP/rmbackup,v
Working file: rmbackup
head: 1.1
branch: 1.1.1
locks: strict
access list:
symbolic names:
	GTK_BACKUP: 1.1.1.1
	R&M: 1.1.1.1
	enterprise: 1.1.1
keyword substitution: kv
total revisions: 2;	selected revisions: 2
description:
----------------------------
revision 1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;
branches:  1.1.1;
Initial revision
----------------------------
revision 1.1.1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;  lines: +0 -0
This is the Rosenzweig and Maffia Backup Program

=============================================================================
RCS file: /usr/local/cvsroot/BACKUP/rmbackup.c,v
Working file: rmbackup.c
head: 1.1
branch: 1.1.1
locks: strict
```

```
access list:
symbolic names:
        GTK_BACKUP: 1.1.1.1
        R&M: 1.1.1.1
        enterprise: 1.1.1
keyword substitution: kv
total revisions: 2;     selected revisions: 2
description:
----------------------------
revision 1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;
branches:  1.1.1;
Initial revision
----------------------------
revision 1.1.1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;  lines: +0 -0
This is the Rosenzweig and Maffia Backup Program

=============================================================================

RCS file: /usr/local/cvsroot/BACKUP/tarlib.c,v
Working file: tarlib.c
head: 1.1
branch: 1.1.1
locks: strict
access list:
symbolic names:
        GTK_BACKUP: 1.1.1.1
        R&M: 1.1.1.1
        enterprise: 1.1.1
keyword substitution: kv
total revisions: 2;     selected revisions: 2
description:
----------------------------
```

```
revision 1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;
branches:  1.1.1;
Initial revision
----------------------------
revision 1.1.1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;  lines: +0 -0
This is the Rosenzweig and Maffia Backup Program

=============================================================================

RCS file: /usr/local/cvsroot/BACKUP/tarlib.h,v
Working file: tarlib.h
head: 1.1
branch: 1.1.1
locks: strict
access list:
symbolic names:
        GTK_BACKUP: 1.1.1.1
        R&M: 1.1.1.1
        enterprise: 1.1.1
keyword substitution: kv
total revisions: 2;     selected revisions: 2
description:
----------------------------
revision 1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;
branches:  1.1.1;
Initial revision
----------------------------
revision 1.1.1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;  lines: +0 -0
This is the Rosenzweig and Maffia Backup Program
```

```
===================================================================
=================

RCS file: /usr/local/cvsroot/BACKUP/Makefile,v
Working file: BACKUP/Makefile
head: 1.1
branch: 1.1.1
locks: strict
access list:
symbolic names:
        GTK_BACKUP: 1.1.1.1
        R&M: 1.1.1.1
        enterprise: 1.1.1
keyword substitution: kv
total revisions: 2;     selected revisions: 2
description:
----------------------------
revision 1.1
date: 2002/10/09 15:23:23;  author: ruben;  state: Exp;
branches:  1.1.1;
Initial revision
----------------------------
revision 1.1.1.1
date: 2002/10/09 15:23:23;  author: ruben;  state: Exp;  lines: +0 -0
This is the Rosenzweig and Maffia Backup Program

===================================================================
=================

RCS file: /usr/local/cvsroot/BACKUP/backprog.c,v
Working file: BACKUP/backprog.c
head: 1.3
branch:
locks: strict
access list:
```

```
symbolic names:
        GTK_BACKUP: 1.1.1.1
        R&M: 1.1.1.1
        enterprise: 1.1.1
keyword substitution: kv
total revisions: 4;     selected revisions: 4
description:
----------------------------
revision 1.3
date: 2002/10/09 20:26:29;  author: ruben;  state: Exp;  lines: +3 -1
Added some useful documentation
----------------------------
revision 1.2
date: 2002/10/09 20:09:59;  author: ruben;  state: Exp;  lines: +1 -1
Add some comments to backprog.c
----------------------------
revision 1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;
branches:  1.1.1;
Initial revision
----------------------------
revision 1.1.1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;  lines: +0 -0
This is the Rosenzweig and Maffia Backup Program

=============================================================================

RCS file: /usr/local/cvsroot/BACKUP/backprog.h,v
Working file: BACKUP/backprog.h
head: 1.1
branch: 1.1.1
locks: strict
access list:
symbolic names:
```

```
                GTK_BACKUP: 1.1.1.1
                R&M: 1.1.1.1
                enterprise: 1.1.1
keyword substitution: kv
total revisions: 2;        selected revisions: 2
description:
----------------------------
revision 1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;
branches:  1.1.1;
Initial revision
----------------------------
revision 1.1.1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;  lines: +0 -0
This is the Rosenzweig and Maffia Backup Program

=============================================================
================

RCS file: /usr/local/cvsroot/BACKUP/errors,v
Working file: BACKUP/errors
head: 1.1
branch: 1.1.1
locks: strict
access list:
symbolic names:
                GTK_BACKUP: 1.1.1.1
                R&M: 1.1.1.1
                enterprise: 1.1.1
keyword substitution: kv
total revisions: 2;        selected revisions: 2
description:
----------------------------
revision 1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;
```

```
branches:  1.1.1;
Initial revision
----------------------------
revision 1.1.1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;  lines: +0 -0
This is the Rosenzweig and Maffia Backup Program

=============================================================
================

RCS file: /usr/local/cvsroot/BACKUP/main,v
Working file: BACKUP/main
head: 1.1
branch: 1.1.1
locks: strict
access list:
symbolic names:
        GTK_BACKUP: 1.1.1.1
        R&M: 1.1.1.1
        enterprise: 1.1.1
keyword substitution: kv
total revisions: 2;     selected revisions: 2
description:
----------------------------
revision 1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;
branches:  1.1.1;
Initial revision
----------------------------
revision 1.1.1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;  lines: +0 -0
This is the Rosenzweig and Maffia Backup Program

=============================================================
================
```

```
RCS file: /usr/local/cvsroot/BACKUP/rmbackup,v
Working file: BACKUP/rmbackup
head: 1.1
branch: 1.1.1
locks: strict
access list:
symbolic names:
        GTK_BACKUP: 1.1.1.1
        R&M: 1.1.1.1
        enterprise: 1.1.1
keyword substitution: kv
total revisions: 2;     selected revisions: 2
description:
----------------------------
revision 1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;
branches:  1.1.1;
Initial revision
----------------------------
revision 1.1.1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;  lines: +0 -0
This is the Rosenzweig and Maffia Backup Program

=============================================================================

RCS file: /usr/local/cvsroot/BACKUP/rmbackup.c,v
Working file: BACKUP/rmbackup.c
head: 1.1
branch: 1.1.1
locks: strict
access list:
symbolic names:
        GTK_BACKUP: 1.1.1.1
```

```
        R&M: 1.1.1.1
        enterprise: 1.1.1
keyword substitution: kv
total revisions: 2;     selected revisions: 2
description:
----------------------------
revision 1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;
branches:  1.1.1;
Initial revision
----------------------------
revision 1.1.1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;  lines: +0 -0
This is the Rosenzweig and Maffia Backup Program

=============================================================
================

RCS file: /usr/local/cvsroot/BACKUP/tarlib.c,v
Working file: BACKUP/tarlib.c
head: 1.1
branch: 1.1.1
locks: strict
access list:
symbolic names:
        GTK_BACKUP: 1.1.1.1
        R&M: 1.1.1.1
        enterprise: 1.1.1
keyword substitution: kv
total revisions: 2;     selected revisions: 2
description:
----------------------------
revision 1.1
date: 2002/10/09 15:23:24;  author: ruben;  state: Exp;
branches:  1.1.1;
```

Initial revision

----

revision 1.1.1.1

date: 2002/10/09 15:23:24; author: ruben; state: Exp; lines: +0 -0

This is the Rosenzweig and Maffia Backup Program

===

RCS file: /usr/local/cvsroot/BACKUP/tarlib.h,v

Working file: BACKUP/tarlib.h

head: 1.1

branch: 1.1.1

locks: strict

access list:

symbolic names:

    GTK_BACKUP: 1.1.1.1

    R&M: 1.1.1.1

    enterprise: 1.1.1

keyword substitution: kv

total revisions: 2;   selected revisions: 2

description:

----

revision 1.1

date: 2002/10/09 15:23:24; author: ruben; state: Exp;

branches: 1.1.1;

Initial revision

----

revision 1.1.1.1

date: 2002/10/09 15:23:24; author: ruben; state: Exp; lines: +0 -0

This is the Rosenzweig and Maffia Backup Program

.....

Branches and Mergers:

A Branch is a space where work is done off the main line of deveolopement

A Merge is done to combine the changes of a branch to the main trunck or another branch of development.

Each repository asset has a revision number. By default, the main trunk is called 1.1

Each entry into a repository is assigned a version number. The default number is 1.1 As new revisions to a file is added, the first number remains the same and the second number is increased. The first number its the trunk, the second number (or minor number) is the revision of the trunk. As one branched, more digits are added to the revision number, depending upon where the branch was split off the main development path.

Keep in mind that each file has a revision number assigned to it by CVS. If you add a new file, it starts with the highest revision number which the branch your adding to has. So if one file in your project has a revision number 1.17, then, then a new file would start fresh with the revision number 1.17. A '-r' option to the commit command updates the revision number to the expressed revision number given as an argument on the command line. For example:

```
$: cvs commit -r 2.0

Fresh Revision of the Backup Program into the repository.


CVS: ----------------------------------------------------------------------
CVS: Enter Log.  Lines beginning with `CVS:' are removed automatically
CVS:
```

```
CVS: Committing in BACKUP
CVS:
CVS: Modified Files:
CVS:  Tag: 2.0
CVS:    Makefile backprog.c backprog.h errors main rmbackup rmbackup.c
CVS:    tarlib.c tarlib.h
CVS: ----------------------------------------------------------------------
~
~

$: cvs status
cvs status: Examining BACKUP
===================================================================
File: Makefile          Status: Up-to-date

   Working revision:    2.0     Wed Oct  9 15:23:23 2002
   Repository revision: 2.0     /usr/local/cvsroot/BACKUP/Makefile,v
   Sticky Tag:          2.0
   Sticky Date:         (none)
   Sticky Options:      (none)

===================================================================
File: backprog.c        Status: Up-to-date

   Working revision:    2.0     Wed Oct  9 20:24:56 2002
   Repository revision: 2.0     /usr/local/cvsroot/BACKUP/backprog.c,v
   Sticky Tag:          2.0
   Sticky Date:         (none)
   Sticky Options:      (none)

===================================================================
File: backprog.h        Status: Up-to-date
```

```
  Working revision:    2.0     Wed Oct  9 15:23:24 2002
  Repository revision: 2.0     /usr/local/cvsroot/BACKUP/backprog.h,v
  Sticky Tag:          2.0
  Sticky Date:         (none)
  Sticky Options:      (none)

===================================================================
File: errors           Status: Up-to-date

  Working revision:    2.0     Wed Oct  9 15:23:24 2002
  Repository revision: 2.0     /usr/local/cvsroot/BACKUP/errors,v
  Sticky Tag:          2.0
  Sticky Date:         (none)
  Sticky Options:      (none)

...
```

While revisions are important, they may not adequately describe the release of a project well, or make an adequate description of the entirety of a project. It is therefor preferable to work with tags when describing a whole project.

When working with different branches, it comes to a point where one needs to merge one branch into a main tree. Under this condition you can merge the branches with the 'update -j' command to CVS. This effects your working copy of the repository. We will look at an example of this after looking at TAGS.

TAGS Tags are a selectable label for a set of files and versions. Tags are generally all CAPITAL letters out of convention.

```
$: cvs tag BACKPROG-1_0 ./BACKUP/
cvs tag: Tagging ./BACKUP/
T ./BACKUP//Makefile
T ./BACKUP//backprog.c
T ./BACKUP//backprog.h
T ./BACKUP//errors
T ./BACKUP//main
T ./BACKUP//rmbackup
T ./BACKUP//rmbackup.c
T ./BACKUP//tarlib.c
T ./BACKUP//tarlib.h
```

Now you can checkup this revision by it's tag, including all it's files:

```
$: cvs checkout -r BACKPROG-1_0 BACKUP
cvs checkout: Updating BACKUP
U BACKUP/Makefile
U BACKUP/backprog.c
U BACKUP/backprog.h
U BACKUP/errors
U BACKUP/main
U BACKUP/rmbackup
U BACKUP/rmbackup.c
U BACKUP/tarlib.c
U BACKUP/tarlib.h
```

Now let's create a new branch which adds a search to our program.
Create a few source code file for the program in your working directory.
And then create a new branch

```
$: vi seach.c
```

Creat out Branch

```
$: cvs rtag -b -r BACKPROG-1_0  BACKPROG-1_0-branch BACKUP
```

Add our new file to the working version of the branch

:$ cvs add serach.c

Commit the Branch to the repository:
cvs -t commit -r BACKPROG-1_0-branch

:$ cvs status serach.c
```
===================================================================
File: serach.c          Status: Up-to-date

   Working revision:    1.1.2.2 Sat Oct 12 21:30:13 2002
   Repository revision: 1.1.2.2 /usr/local/cvsroot/BACKUP/Attic/serach.c,v
   Sticky Tag:          BACKPROG-1_0-branch (branch: 1.1.2)
   Sticky Date:         (none)
   Sticky Options:      (none)
```

Now that our changes are committed to the repoitories branch. We can merge the branch with update and -j like this:

:$ cvs update -j BACKPROG-1_0-branch
cvs update: Updating BACKUP

:$ cvs commit -m 'MERGED BRANCH'
-m adds the log entry.

We now have merged the branch to the trunk. Run status to see.
End --

Peer to Peer File Sharing: Is it Theft?

Ruben Safir

The Recording industry of America, and the Motion Picture Industry is flooding Congress with proposals to combat peer to peer trading of music and copyrighted material. They want to end what they call 'casual theft' and to educate you about that is legal and what is right. The proposals they put together will affect your future for a long time and affect your life in a very personal way.

We agree with industry that theft is wrong, and that young people need to learn the difference between theft, copyright violation, fair use, the DMCA, force contracts, intellectual property, DRM and copyright abuse.

What is intellectual property?

Intellectual property is bad term which has made it's way into our legal system. It is designed to describe creative works and inventions of the mind. In the US, all intellectual property, that is music, recordings, paintings, books, and writings, are owned by the public. The people, according to the Constitution, are the sole owners of all intellectual properties. In truth, this is a great insight by our founding fathers. It's obvious that as a species, we naturally all have a stake in our shared culture. In fact, culture and intellectual property is largely interchangeable words. Unless an idea is shared, it can not become part of our culture. Culture is the fabric of our existence. And our rights to share equally in our culture is guaranteed to us through several Constitutional provisions, including Freedom of the Press, Free Speech, The Right to be Secure in our Property, the Freedom of Religion, and so on. Each of these guarantees assures us Freedom, the Freedom to participate as equals in our larger society.

Anything that obstructs your Freedom to share in cultural expression is a hindrance to your equality among men (and women). All intellectual property is owned by the people.

What is copyright?

Copyright is a limited restriction on the peoples freedom. It grants limited control of newly created cultural artifacts to the individual creator, in order to promote the publication of more works. Copyright is an 'exclussive' right of the creater to a cultural artifact. It allows the owner of a Copyright the right to prevent the distribution of replicures of a cultural work.

Copyright is limited by two principles of American Law. First, Copyright is restricted by time. Copyright can only be for a limited time. Congress can not create a law of indefinite Copyright under the Constitution. Naturally enough, if works remain indefinitely under 'exclussive' control , then an aggregation of cutural works are then controlled increasingly by a few major businesses. This creates an erosion of Freedom, and eventually enslaves society as a whole, making us a sharecropper of our own cultural heritage.

Secondly, copyright is limited by those rights reserved to the people under the Bill of Rights and other Constitutional Amendments. Copyright, for example, can not be used to invade private property. You can not use Copyright to discriminate against people of color by restricting distribution to only white people. You can not use copyright to prohibit freedom of speech, especially political speech. All these things limit the 'exclusivity' of copyright.

What is Fair Use?

Fair Use is a legal doctrine grounded in common law, and later outlined in section 102 of of the copyright act. It's a legal defense against accusations of copyright violations. While many think of Fair Use as our protection from Copyright abuses, most legal scholars consider it only as a legal defense. Copyright holders are not required, for example, to prove in court that a use under consideration by the court is not a fair use. It's the defense which needs to prove that a use is fair use, and exempt from Copyright restriction.

What is the DMCA?

The DMCA stands for the Digital Millenium Copyright Act of 1998. This law was designed to give holders of Copyright franchises greater control over the distribution of copyrighted cultural artifacts. It allows the copyright holder to limit access to these artifacts in society, even when copies of those artifacts are purchased by individuals and is their personal property. This law is universally considered to be written poorly, and it is contriversal because it, in theory and in practice, it encroaches on protections we have under the Bill of Rights. It has led to the arrest of jailing of innocent people trying to make proper use of their property. It also prevents ownership of copies of cultural artifacts in digital forms, making everyone a share cropper to their own cultural heritage.

What is a Sharecropper?

A sharecropper is a person, normally a recently freed slave or a serf, who owns no property, but rents land, tools, and housing in exchange for a portion of the produce farmed on the landlords land. They have no rights to the land, or the tools they use or their homes.

What is DRM?

DRM stands for Digital Rights Management. DRM is theft. It takes control of your property, often a computer, but also a DVD player, a MP3 players, a VCR, or a tape recorder, any digital device used to play media that you own. It seizes control of your property by determining when, where and how use can use it.

The purpose of DRM is to make you a sharecropper to your own media and property. It prevents you from watching a DVD, accept to watch it in a pre-approve place and time. It can prevent you from viewing your media more than a specific number of times. It can prevent you from fast forwarding through unwanted material, to prevent you from printing parts of it, or copying for archival reasons, or for note taking, and to prevents you from sharing the information which you own. In addition, DRM can spy on you, and report information about you to others, prevent you from using the devices or software of your choosing to view the material, and prevent you from archiving or using the material for creative processes.

For example, DRM currently prevents you from watching a DVD on anything other than an approve device or operating system. You can not watch a DVD without a DMCA approved DRM controlled device. You can not watch a DVD on the GNU/Linux operating system, or other new operating systems on the market. Some books purchased at universities must be brought on DVD. At NYU Dental School, these books turn themselves off at the end of the year if you do not make subscription payments to the publisher. It is not legal to sell the books on the DVD second hand to another student. It prevents ownership of purchased computers and media by forcing you into a contract after a cash purchase. Contracts normally can not be enforced if they are not fairly negotiated contracts. This is called forced licensing. Forced licensing enslaves the purchaser by forcing them into an economic agreement which they have no control of or say in.

Without 'special' exemptions in the law, DRM prevents the normal exchange of ideas and information. For example, lending libraries become illegal, playing music for a friend on an non-assigned player is illegal, sharing a research paper with a college is legal, mixing a collection of music for a party is illegal, sharing a song with a friend is illegal. With DRM, full control of where, when, and how you use any book, music or video is the sole discretion of the Copyright Franchise holder.

What is peer to peer trading?

Peer to peer trading is sharing a file with information between two private individuals. This is the most fundamental human behavior which defines us as a species. Our ability to share, communicate and co-operate is how we develop our culture, and how we define our existence. The sharing of privately owned and publicly owned copies of information is the lynchpin of democratic discourse.

The peer to peer trading of files for non-commercial use is legal in every sense of the word. Your protected by the Bill of Rights in this activity. Your guaranteed security in your property, freedom of speech, freedom of the press, and against arbitrary search and seizure without a warrant. These protections guaranty you the freedom to share music with other individuals for non-commercial purposes.

For example, recently the House of Representatives took up a bill in the Intellectual Property sub-committee which proposed that the recording industry should be allowed to break into your computer and prevent you from sharing files with others. During the testimony, Congressman Howard Berman, the sponsorer of the Bill, proclaimed that, "Nobody is suggesting that we end the sharing of files between individuals. That would violate 'Fair Use'".

However, the recommendations currently being passed by Congress and proposed by the Cultural Distribution Companies, such as the record companies, and movie companies, have already done just this.

The recording industry is spending millions of dollars a year trying to convince you that it is illegal and immoral to share music and other information. You should know they are wrong. The act of receiving a file with music, text or video from someone and listening to is is not only legal, but it is one of the most cherished rights of mankind. Our ability to share information is the foundation of our society and the basis of the civilization which you are a participant in.

What is Water?

Water is a natural resource available freely to everyone in a civilized society, and in fact it's delivery and purity is subsidized by the government.

What is bottled water?

Water which is put in a bottle, marketed with advertising, and sold under a trade mark.

What is Culture?

Culture is like Water.

What is a Copyrighted Work?

A copyrighted work is like bottled water. In the US, copyright law says every new creation is automatically bottled water.

Why is this important to me?

Congress today is trying to pass laws which will end private ownership of all computers in order to support copyright control. The recording industry and movie industry resents you as a free person who uses the culture around you in a natural way. There is serious efforts underfoot to give the publishing industry preemptive rights in the use of any media you buy, may it be a book, a video, or music. They want to be the great gatekeeper of American Culture, dishing it out in small pieces to those who they think fit.

Just in the last few months Congress has accused an entire generation of being file sharing people thieves. The are trying to impose smart chips which spy on you into the hardware of your digital music equiptment and computers. They have proposed laws to allow the record industry to hack into your computer and invade your privacy if they think your sharing files of copyrighted works,they want to give the recording industry the right to do this without needing to get a warrant from a court. They are doing everything they can to prevent you from ripping music, or copying video. They are trying to turn the interactive internet into a broadcast stream. They have made it legal for students who buy required books for school, to have those books turned off at the end of the semester. They want to end your ability to participate in society through digital means. They have made this a crime. They have arrested people for making it possible for owners to read their books, and for people to watch their movies in their home.

The publishing industry is trying very hard to make you feel you live in freedom, while simuteanously ending your freedom through digital control. What is Digital Rights Management? It is the means to digitally control your behavior in cases where they would never get you to agree to such terms in a contract.

Everything created in the last 100 years is under copyright control. This means every book report, every letter, and ballet and ever photo in the 20th and now the 21st century is under copyright today. By preemptively restricting access to your property through copyright, every normal activity is restricted by a third party.

It's been said that there is only "Freedom of the Press" for those who own a press. Today, we all own a press, and it must remain a Free Press, individually owned and under control of the user.

Congressman Coble of North Caroliner, chairman of the IP sub-committee in the house, stood up at the end of the hearings on the Berman Bill on pontificated, "Can some here tell me why all this file sharing is not theft. This is simple larceny." Congressman Coble is worng. Some arguement can be made that massive file sharing damages the value of copyrighted works, and therefore, might be considered Copyright Infringement. The record companies are afraid, however, to tell it's public the truth, that in their opinion, you do not own any records, or recordings. And that you can not use them freely and naturally. While they tell Congress that an entire generation of American's are thieves and pirated, they refuse to prosecute a single one of them. Because then everyone would be exposed to the copyright abuses that the law today has created. It's your future they have shackled, and you can do something about it now.

What can I do about this, nobody listens to me?

or

What can we do about it, why fight it, everyone in Congress is on the take.

While fighting for your rights in the digital age is difficult, it is not impossible. New Yorkers for Fair Use, http://fairuse.nylxs.com, has fought this battle for over a year, and we have had success. We have frustrated efforts in the Department of Commerce to impose DRM by administrative executive action. We have succeeded in fighting against laws which would allow the recording industry to hack your computer without a warrant. We have raised on Capital Hill the slogan, 'DRM is THEFT, We are the Stakeholders', but we have not done enough. With your help, we can turn the tide on this battle.

We can have the DMCA repealed,. We can put the burden of proof for infringement on the Copyright Holder. We can make laws which ground the fair use doctrin properly into the 4th ammendment which guarantees individuals protection in their property, and we can make sure that nobody is forced into a license for music and information which they actually own.

Join New Yorkers for Fair Use and register to vote. Tell Congressman Weiner that we reserve for ourselves the right to trade files, and to share information, and without the oversight of Time-Warner, Disney and Fox. Tell Congressman Weiner that you will vote on this issue, and they he must protect our rights. If file sharing is theft, which it isn't, then so be it. But do not take control of our computers and our media. They are ours.

The Free Software Chamber Of Commerce

Opening New Fronts in the Licensing War

By Adam Kosmin:

Adam Kosmin is a 29 year old, blond haired young man, who looks wiser than is age, with a steel intensity to his glaze and a soft spoken voice. Adam is one of the most recent victims of the Microsoft forced licensing con game, one which the US Government has tried for years to shutdown with no results. But Adam doesn't take abuse easily, and he is taking all the proper steps to finally drive a stake into the OEM-Microsoft collusion which gives Microsoft defacto taxing rights on every laptop computer sold in America.

For those not in the loop, the issue is not new. Every computer which is sold with a Microsoft Operating system tries to enforce rules on the end users for which they can agree, or get a refund. That click through license that you see when you buy a new computer, gives you the right to refuse, and to get a refund on the operating system. It's been known for many years that nobody ever gets a refund. Furthermore, it is known that OEM contracts force computer manufacturers to pay a fee for every computer shipped, weather they have an OS installed or not. And it's well known that the courts have ordered Microsoft to take back it's software and to give a refund to those who don't want to agree to the license which Microsoft tries to force on the public. But try to get your refund, and somehow all this is meaningless to the manufacturers and Microsoft.

Adam is a top rated network administrator for FCGMS, the IT management company which keeps things going at Columbia Presbyterian Hospital in upper Manhattan. Presbyterian has a mixed computing environment and Adam has been keeping their systems working for the last few years. Adam's favorite opening system, both at work and at home is GNU/Linux. An avid Free Software user for 2 years, Adam's computing environment is choice is Debian Linux, allthough he that a family of computers which use a variety of free software, accept for one windows computer he keeps, just in case.

In April of 2002, a few months ago, Adam started shopping for a new laptop. He did his research throughly, comparing price and features until he came to settle on a Toshiba Protige 2000. While he looked at Dells, Sony's and other brands, he found that the 2.5 pounds of weight, very low failure rate and superior customer service locked his decision for this machine. The only trouble he had was that the computer came preinstalled with a Windows OS. in fact, all the computers he looked at came exclusively pre-installed with a Microsoft Operating system. In total he paid about $2300 and had the system shipped to work.

At the time he made the purchase, he asked the retailer to ship the system without the OS. The Toshiba on-line retailer refused to ship the system without an operating system. Adam inquired with several levels of tech support and sales support prior to the purchase. Up and down the entire supply chain, the authorized Toshiba dealers refused to sell the system without the OS. He documented every aspect of the transaction. And when the system arrived at the office, he immediately pulled out a Red Hat GNU/Linux boot disk and proceeded to do a network install on the laptop. Adam never saw any EULA, as Windows was new booted for use. He then proceeded to try to get a refund for the Microsoft OS, as it is stated in the packaging of the system.

Adam did his homework when he made the decision to by a laptop. After it became appence that he would not be able to purchase the system without Windows preinstalled, he looked at what would need to be done to get a refund. He looked on the Website called the 'Windows Refund Center, and was careful about how he proceeded to use the system. He tired to get a refund through but Toshiba International and Toshiba on-line. He finally got a hold of John Snithson. Smithson refused the request for a refund claiming the following major reasons.

First, Smithson claimed that the OS was like any component of the computer rather than a separate product. They compared it to the hard drive. Just as they didn't give you a choice of hard drive, they don't give you a choice of OS. Secondly, they claimed that Toshiba gives you a separate license in the bag. Ripping open the bag forces Adam into a contract. Thirdly, they said he didn't like it, to take them to court.

Adam objected because the plain meaning of the license permitted a refund. And nobody seemed to be interested in honoring even these terms of even the forced license. When Tosiba said they would see Adam in court, Adam's response was to say that they have to give a refund and he read them the license. He documented everything that happened, however he never mentioned that Microsoft was under a court injunction to give refunds to purchasers who refuse the license. After documenting everything, he sent his write up to the Linux Journal where it was published. He corresponded with the FSF about the matter, and OSI. He will be contacting the State Attorney General this week, and he has filed a complaint with the FTC via a web form on the internet.

But Adam has done more than complain. He's taking a very aggressive stand on

this issue. He's started a new website called windowsrefund.net and he is collaborating with others on this drive to protect consumers choice. When asked why it is so important to get a refund, Adam responds with clarity, "We need choice in order to protect Free Markets that Microsoft
was under a court injunction to give refunds to purchasers who refuse the license. After documenting everything, he sent his write up to the Linux Journal where it was published. He corresponded with the FSF about the matter, and OSI. He will be contacting the State Attorney General this week, and he has filed a complaint with the FTC via a web form on the internet.

But Adam has done more than complain. He's taking a very aggressive stand on this issue. He's started a new website called windowsrefund.net.

Adam says, "Free Markets require freedom of choice. I had an option to buy the laptop pre-installed with GNU/Linux, from a vendor like EmperorLinux, but he would still be required to buy the Microsoft License which had no value to him. Emperor is
an after market product. All the OEM's, especially those that carried the product he specifically wanted, only shipped the Laptop with the OS, whether you want it or not. And then they claim the form a single product. And then they make you sign a specific and separate license for the software. If it has a separate license, it must be a seperate product. Since it's true that all the major vendors refuse to sell you a laptop without windows, and because all of them are selling the same software product, then there is no lateral competition either. The is just no way to buy a laptop from any of the major vendors without paying for Windows, and I don't want windows. This is a complete monopoly. It's like Microsoft has a tax on every laptop, whether I want to pay for their products or not. When your forced into a contract like that, it's a form of slavery. It's just not right"