



The content of this magazine is released under the Creative Commons Attribution-Share Alike 3.0 Unported license. For more information visit <http://creativecommons.org/licenses/by-sa/3.0>

Issue #1 - August 2009

EDITORS DESK	3
DEVLAND & SCHWAG REPORT	4
FEATURED INTERVIEW	5
COMMUNITY HIGHLIGHTS	6
REVISION CONTROL SYSTEM	7
TIPS & TRICKS	9
THE FUN SECTION	10
SOFTWARE REVIEW	11

Arch Newsletter + AUM THE MISSING PIECE





NEW TO ARCH?

Arch Linux is a lightweight, flexible Linux® distribution that tries to Keep It Simple.

Our strong community is diverse and helpful, and we pride ourselves on the range of skillsets and uses for Arch that stem from it. Please check out our forums and mailing lists to get your feet wet. Also glance through our wiki if you want to learn more about Arch.

Welcome to the new Arch Linux Magazine!

This month, a new era begins for Arch Linux users. Daniel Griffiths (Ghost1227) and I have been discussing ways to further benefit our loyal readers for several weeks. We came to the conclusion that if we joined our efforts, we could create the ultimate publication, something that Arch Linux users would enjoy the most and remember for a long time. This new publication would serve the same purpose as the newsletter, keeping the users informed about the current events in the community and the development of Arch Linux, as well as ways to enjoy it even more.

Unfortunately, I couldn't do much for the magazine this month, or at least not as much as I want, as I have been very busy in real life. Thankfully, Daniel has taken over the responsibility of filling the gap left by me not being available this month. And from what I have read, he did a great job at successfully compiling the best digital magazine for a Linux operating system there is. Thanks Daniel.

For the die hard fans of digital magazines there is a PDF version, and for those of you that like to keep it simple there is the traditional HTML version, so both crowds are being provided with the same quality publication.

And now, on behalf of the Arch Linux Magazine team, I present to you the first Arch Linux Magazine. Enjoy!

Eduardo Romero (Kensai)
Editor in Chief, Arch Linux Magazine
eduardo@archlinux.org

ARCH USER STAFF

Eduardo Romero (Kensai) .. Editor in Chief
Daniel Griffiths (Ghost1227) Editor
David Crouse (Crouse) Contributor
Denis Martinez (Denis) Contributor
Ivy Foster (joyfulgirl) Contributor
Kevin Eldridge (LeoSolaris) Contributor
Loui Chang (louipc) Contributor
Sam Harada (Peasantoid) Contributor

DEVLAND & SCHWAG REPORT

Change to /dev/tty*

Those of you who haven't updated your system recently might want to take note. We recently implemented a change that switched us from the old /dev/vc* to /dev/tty*. This change involved altering the filesystem, syslog-ng, udev and initscripts packages. If you are one of the few who hasn't caught this update yet, be ready. This update will point out who doesn't pay attention to pacman's warnings!

Read more:

<http://www.archlinux.org/pipermail/arch-dev-public/2009-July/012401.html>

dbscripts update: arch=any

Thanks to Aaron, the most recent dbscripts update has potential support for the much anticipated features arch=('any') and split packages. While this does not affect AUR packagers yet, once it is implemented system-wide, packagers will finally be able to list their packages as arch=('any') instead of arch=('i686' 'x86_64').

Read more:

<http://www.archlinux.org/pipermail/arch-dev-public/2009-July/012419.html>

2009.08 RC is available!

The snapshot RC for the upcoming 2009.08 installation media was recently released for public testing. This latest revision includes more extensive use of the AIF installer, a better disk editor, support for automated installations, replacement of unionfs with aufs, and much more!

Read more:

<http://www.archlinux.org/pipermail/arch-dev-public/2009-August/012713.html>

KDE 4.3 packaged and split

The recently released KDE update features many upstream enhancements, as well as a new set of packages that allow you to only install those packages you like. The most notable of these changes are split packages and meta packages to ensure a smooth update.

Read more:

<http://www.archlinux.org/news/455/>

Arch Linux Schwag Report

Unfortunately, we were not able to get an updated schwag report for the month due to time constraints. However, this does not mean that you should stop supporting us! Visit our schwag merchants and show how much you love Arch!

Schwag @ Zazzle:

<http://www.zazzle.com/archlinux>

Dusty's Schwag Shop:

<http://schwag.archlinux.ca>

FEATURED INTERVIEW - LOUI CHANG

ALM: Hello Loui. First off, we would like to thank you for taking time out of your busy schedule to talk to us. Can you tell us a little bit about yourself and how you became a Trusted User?

Loui: Hmm that's a very open question. When it comes to Linux or computers I'm really just a hobbyist. I actually work as a machinist. I've used several different Linux distros, but when I finally came to Arch I was pleasantly surprised at how easy it seemed to be able to contribute something via the AUR. Even though I didn't use it that much, there were some nagging bugs so I decided to grab the source code and start patching. I became a Trusted User because I wanted to have a look into the way they were doing things so that maybe I could help improve those systems if they were lacking. I'm not a big package maintainer though.

ALM: You have made a bit of a name for yourself with your work on the AUR. Can you tell us what some of the notable enhancements you've made are and what big changes are still in the works?

Loui: Haha. Well, it's all been a very incremental process. I think most of the

cool stuff has been done by others. The Json interface and the initial site re-styling was implemented by Eli Janssen, for example. I guess I've just been here to help push things along and guide development. I think most of my work lies in the back-end, and refactoring code. As for the future, I'd like to attempt a complete reimplementaion with a more 'robust' client/server model. I'd like to make it a real application from the ground up rather than a web app with add-ons. I don't want to get anyone's hopes up though. It has been attempted at least twice already.

ALM: Alright. Generally speaking, is there any new technology that you are looking forward to?

Loui: I'm looking forward to trying out btrfs, getting a phone with Android, USB 3.0, and autonomous killer robots.

ALM: Nice... Loui, thank you for your time, as well as your continued work with Arch. Do you have anything else you'd like to say to our loyal readers?

Loui: Be prepared... Oh, and get hacking!

Official Arch Linux Sites

Homepage	http://www.archlinux.org
Forums	http://bbs.archlinux.org
Wiki	http://wiki.archlinux.org
Bugs	http://bugs.archlinux.org
AUR	http://aur.archlinux.org

Arch Around the World

Brasil	http://www.archlinux-br.org
Chile	http://www.archlinux.cl
Czech Republic	http://www.archlinux.cz
Denmark	http://www.archlinux.dk
France	http://archlinux.fr
Germany	http://www.archlinux.de
Hungary	http://archlinux.fsf.hu
Italy	http://www.archlinux.it
Poland	http://www.arch-linux.pl
Romania	http://archlinux.ro
Russia	http://archlinux.org.ru
Spain	http://archlinux-es.org
Sweden	http://archlinux.se
Turkey	http://www.archlinux.org.tr

COMMUNITY HIGHLIGHTS

[community] Switched to SVN

The community repository has finally been moved to SVN! After a hectic day of rebuilds, we completed the boost and libjpeg rebuilds and froze community long enough to transition to the new server and convert everything to subversion. Although this involves a bit of a learning curve for some, it should be beneficial in the long run.

Read more:

<http://www.archlinux.org/pipermail/aur-general/2009-July/005844.html>

CAW Taskbar

Arch user dcurtis has begun work on a new, updated panel written in Python. But why another panel? dcurtis found the existing options to be generally outdated and uncustomizable, so he took ideas from various sources (including the Awesome taskbar, dzen, and PyPanel) and began work on his own!

Read more:

<http://bbs.archlinux.org/viewtopic.php?id=76889>

Introducing OFileBrowser

Once again, Arch user Xyne has outdone himself. OFileBrowser is a perfect example of what happens when a good programmer gets bored, but it's still cool! Any OpenBox users looking for a pipe-menu file browser should definitely check this one out!

Read more:

<http://bbs.archlinux.org/viewtopic.php?id=77197>

Arch-Home Redux

For those of you who missed it the first time around, Pyther has given his Arch-Home script an overhaul and found it a home. Arch-Home is a simple webpy script that generates an ideal startpage for Arch Linux users. If you use Arch (and a graphic browser), you need to check this out!

Read more:

<http://bbs.archlinux.org/viewtopic.php?id=76721>

Stapler: PDFs in Python

For those of you who have worked with PDFs from the command line, PDFtk was probably an important part of your toolkit. Unfortunately, PDFtk has gone the way of the dodo. However! Archer Heller_Barde has stepped up to the plate and written Stapler, a python utility for manipulating PDF documents based on pyPDF.

Read more:

<http://bbs.archlinux.org/viewtopic.php?id=77388>

Desktop Workshop

Any OpenBox users in the house? What started as an off-hand remark by MrGreen has turned into a full-fledged obsession for Archer Ghost1227. After rising to the unintentional challenge, Ghost1227 took over the thread and started a companion website for custom OpenBox desktops. If you don't think OpenBox can look good, think again!

Read more:

<http://bbs.archlinux.org/viewtopic.php?id=74915>

REVISION CONTROL SYSTEM BY DAVID CROUSE

RCS, or the GNU Revision Control System, is a system used to keep track of files that can be edited by many people or that may need to be reverted to an earlier state at some point in time. For the purposes of this article, I will concentrate on the latter simply because I think it will actually pertain to the larger majority of our readers.

If you have multiple people working on a coding project, you probably already know what a revision control system is. However, for those of you who don't already know, read on! It's much easier than it sounds. I really love using Arch Linux as a server distro. While many think that it is not up to the task, I think they are sadly mistaken. I believe it is a great server distro that is extremely flexible and easy to maintain.

But I digress, that is not what I intended to talk to you about today. My point was that if you run a server, you make changes to configuration files quite often and having a revision control system can come in quite handy. However, regardless of your reason for using a revision control system, RCS should prove to be worth the time it will take you to do a bit of research. Its short learning curve and usefulness should be

enough to pique the interest of even the most reluctant of users. I know it's not always fun, but the old adage "Backup, backup, backup" has been ringing in my ears and RCS proves to be just the tool for many of those simple backup jobs that might otherwise not get done.

Using a revision control system does *not* have to be hard. It should be easy enough to become second nature and allow you to fluidly incorporate it into your day to day work. So, let's get started, shall we?

First and foremost, install RCS.

```
$ pacman -S rcs
```

To ensure correct usage, we should verify two things.

1. We have a log of all changes we have made to the file.
2. We can revert back to previous versions of the file with ease.

There are two primary ways to use RCS.

1. Use "co" and "ci" on files in any directory. This will cause files to be created that match <file> but have a ",v" at the end.
2. Create a subdirectory called RCS in

every directory you want to work in, and RCS will automatically store files there, and keep your work area clean.

You could get creative and create symlinks to a single master directory, but that is beyond the scope of this article. Besides, having a RCS subdirectory in your working directory lets you know quickly that you should use RCS to edit those particular files. See? It's not so hard. Moving on...

There are two commands that users will find themselves using regularly. These two commands are 'ci' and 'co' (check in and check out, respectively). Pretty darn complicated so far isn't it? Ok, let's see how we start using RCS with a new file.

Let's say that we have a file called "myshellscrip.sh" that we want revision control on. The very first thing we need to do is check the file in to RCS. This is done through the following command:

```
$ ci -u myshellscrip.sh
```

This will initialize the RCS version of the file. It will also ask you for a DESCRIPTION of the file. This is not a change log entry, it

REVISION CONTROL SYSTEM BY DAVID CROUSE

is for entering a description of the file, and can be multiline. Woot! You have now checked in your first file to RCS! Ok... so how do I check it back out and make some changes? It's just as easy! To check out myshellscrip.sh, just issue the following command:

```
$ co -l myshellscrip.sh
```

So, we now have myshellscrip.sh checked out. We can open it with our favorite editor and make any changes we want. Then we save the file. Now this is the tricky part. The part after you have saved the file, when you must check the file back into RCS. This is where people drop the ball and forget to recheck in the file. But it's so simple! We've already used the command once! Once again, just issue the following command:

```
$ ci -u myshellscrip.sh
```

This immediately prompts you for a change log entry. Again, this can span multiple lines if you need it to, just end the log entry with a period on a single line. For those of you that are like me and will most likely forget to check things back in, I wrote a quick script, amazingly enough called

RCSedit. It's very basic, but it does the job very nicely. It will use whatever editor you tell it to, and it checks out the file using your editor, then initiates the check in procedure when you have completed editing. Nice, huh?

```
#!/bin/sh
if [ ! "${EDITOR}" ]; then
    EDITOR=vi
fi
/usr/bin/co -l $1
${EDITOR} $1
/usr/bin/ci -u $1
```

So now we can initialize RCS by creating an RCS directory, we can initialize the file by running the check in command, we can checkout the file and edit it, and finally check it back in. I sort of feel like hitting my red Staples button that says "That was easy". So, what else do we need to do? Well, the following is a rundown of commands that you might find useful. I highly suggest reading the man pages for them and becoming familiar with all the options they provide. This is a very basic set of commands that will probably help you accomplish 99% of what you might want to do with RCS.

```
ci -u <filename> - Initialize a file
co -l <filename> - Check out & lock a file
ci -u <filename> - Check the file back in
rlog <filename> - Shows the long log of changes made to a file
rcdiff <filename> - Compare revisions
rcs -u <filename> - Override locked file.
```

So now we can do all sorts of stuff with RCS, but wait! How do you revert a file to a previous version? It's quite simple as well. We don't even need a new command. Let's say we have made several revisions to our myshellscrip.sh script, but we decided the last changes just sucked. We now need to go back one revision to our previous working copy. Not a problem! First, you would use the rlog command to list all of the revisions and revision numbers. Let's say the last revision was 1.2 and you want to revert to 1.1. Issue the following commands:

```
$ rcs -u myshellscrip.sh
$ co -l -r1.1 myshellscrip.sh
$ ci -u myshellscrip.sh
```

It's that simple! It may take some getting used to, but when you get the hang of it you'll wonder why you didn't start sooner!

Piping Without the Plumbing

Have you ever wished you could easily feed a shell command another command's output without dealing with plumbing? Well wish no more! It is possible, simple even, with a little bash. Anyone with a little bash experience will recognize that running ``$(someprogram)`` will return the output of the program of your choice. For example, here's a oneliner that will get rid of some pesky file suffix:

```
$ for i in *.o; do mv $i $(basename $i .o); done
```

Of course, you can still do some plumbing if you want. Say you wanted to show off one of your cool screenshots, but couldn't decide which. Let fate (or at least a pseudorandom sorting algorithm) decide.

```
$ mutt -s "Screenshot" -a "$(ls sshot/* | sort -R | head -1)" -- pimpmyarch@archuser.com
```

On Mailcap

Although a bit obscure, `$HOME/.mailcap` can save you a bit of hassle. Not every

application will source it, but mailreaders and web browsers will often take a peek and see what's there. The file itself is plain text. Comments begin with a `#`, and each uncommented line is an instruction for how to open a specific type of file such as "text/plain" or "application/pdf". By using this, you don't have to set your preferred applications in every application you use. Type them in once and let the computer do the rest of the work! The following is an example mailcap file.

```
# Sample mailcap file
# Launch images with display
# Launch videos with vlc
# etc
image/gif; display %s
image/jpeg; display %s
image/tiff; display %s
image/x-portable-bitmap; display %s
image/x-ppm; display %s
audio/*; sfplay %s
video/mpeg; vlc %s
video/*; vlc %s
application/postscript; ghostview %s
application/x-dvi; xdvi %s
application/pdf; xpdf %s
application/acrobat; xpdf %s
```

Fast Line Editing

Do you know about `readline`? `Readline` is an incredibly handy library that enables you to manipulate your command line in useful ways. And these commands work in many programs other than the shell! You can cut, paste, and move through text on the command line just as if you were using a powerful text editor.

Try opening a shell and type a few words in without hitting `[enter]`. Now hit `Ctrl-u`. The text is gone! That's called killing text. Hit `Ctrl-y`. It's back! That's yanking text.

There are many more ways to manipulate text on the command line. The following list is just a sample of what is available. See ``man readline`` for even more info!

```
Alt-b Beginning of current word
Alt-f End of current word
Ctrl-a Beginning of line
Ctrl-e End of line
Ctrl-w Delete to beginning of word
Alt-d Delete to end of word
Ctrl-u Delete to beginning of line
Ctrl-k Delete to end of line
```

THE FUN SECTION

What's your Favorite Taco?

In true Arch form, Acecero has asked for everyone's favorite taco. If you haven't told us your favorite, do it now!

Read more:

<http://bbs.archlinux.org/viewtopic.php?id=76753>

Remember the 60s?

For our younger users who don't remember the 60s, watch it and learn! For everyone else, watch it and mourn your age!

Read more:

<http://bbs.archlinux.org/viewtopic.php?id=75998>

Kiloseconds?

Have you ever wondered what time it was in kiloseconds? Well wonder no more! Archer haxit has started the Kilosecond Project (KSP) to answer just that question. See what the fuss is all about!

Read more:

<http://bbs.archlinux.org/viewtopic.php?id=74226>

```
<Animeking> can'tfigureouthowto trywebcamwih kopete
<Ghost1227> Animeking: apparantlycan't figureouthow touse the spacebareitherhuh?
```

```
<rson> bruenig: i've decided to stick with your colors
<rson> i think my eyes will get used to it
<rson> i need a better vim scheme though
<bruenig> my free colors that I have given to you out of good will
<Ghost1227> lies!
<Ghost1227> nothing in life is free!
<Ghost1227> except linux
<rson> bruenig: i'm liscensing them under a nonfree liscense and selling them to anyone who
  wants them. i am also suing you for using them without my permission.
<Ghost1227> yay!
<Ghost1227> capitalism win
<rson> i told you i'd be at the top one day
<rson> it starts now
* Ghost1227 prepares to file an antitrust lawsuit against rson(TM)
<rson> :)
<bruenig> nah
<Ghost1227> rson: do we have to pay you everytime we invoke your name?
<Ghost1227> s/rson/rson(TM)
<rson> yes. even when it's embedded in other common words that are unrelated
<rson> like person
```

```
<improvement> michael jackson died. hes dead
<jelly12gen> !next
<phrik> Another satisfied customer! NEXT!
<jelly12gen> *ouch*.
```

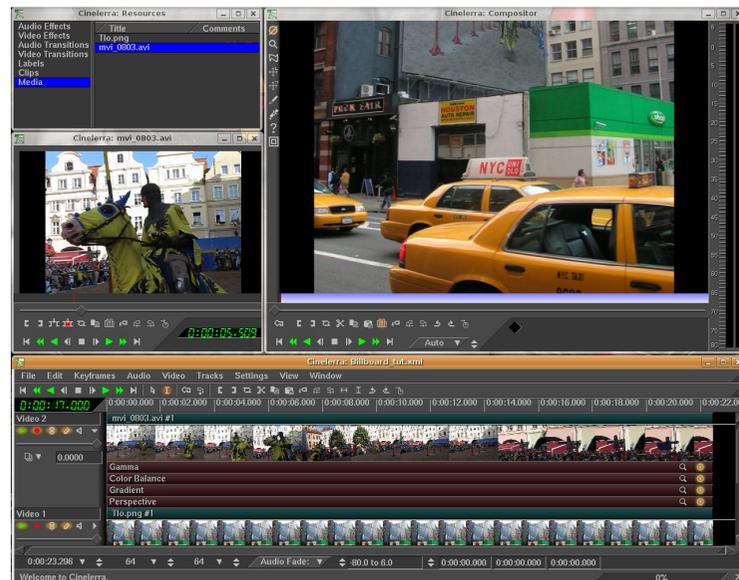
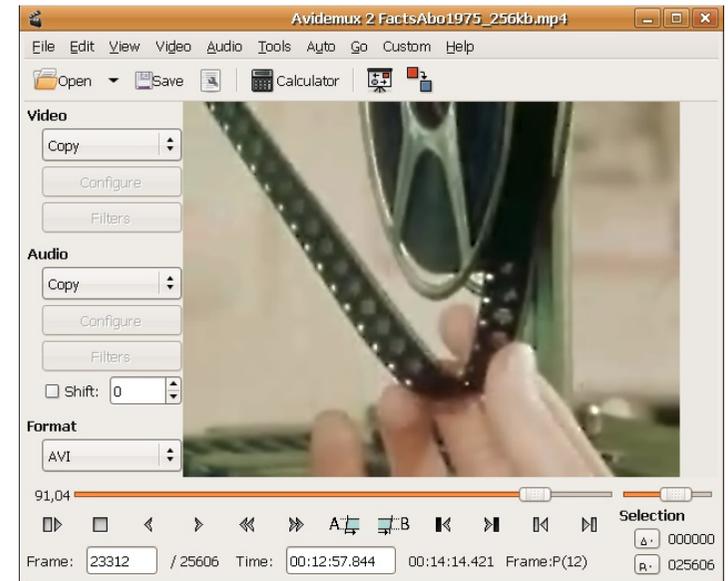
SOFTWARE REVIEW: VIDEO EDITORS

AVIDEMUX

<http://fixounet.free.fr/avidemux>

Avidemux is an open-source, cross-platform multipurpose video editing and processing program. Designed for simple cutting, filtering and encoding tasks, it supports many file types, including AVI, DVD compatible MPEG, MP4 and ASF, using a variety of codecs. Featuring task automation and powerful scripting capabilities, avidemux is one of the most popular video editors available for Linux, and with both Qt and GTK interfaces, it's easy to see why!

To install avidemux, use the ``avidemux`` package in [extra].



CINELERRA-CV

<http://cinelerra.org>

Billed as the most advanced non-linear video editor and compositor for Linux, Cinelerra-CV (or Cinelerra Community Version) is a community project to extend on the work of Heroine Virtual Ltd.'s original Cinelerra. With support for OpenGL shaders, most common codecs, realtime effects, and floating point compositing, among other things, it's easy to see why Cinelerra has become known for it's power and capabilities.

To install Cinelerra-CV, use the ``cinelerra-cv`` package in [community].

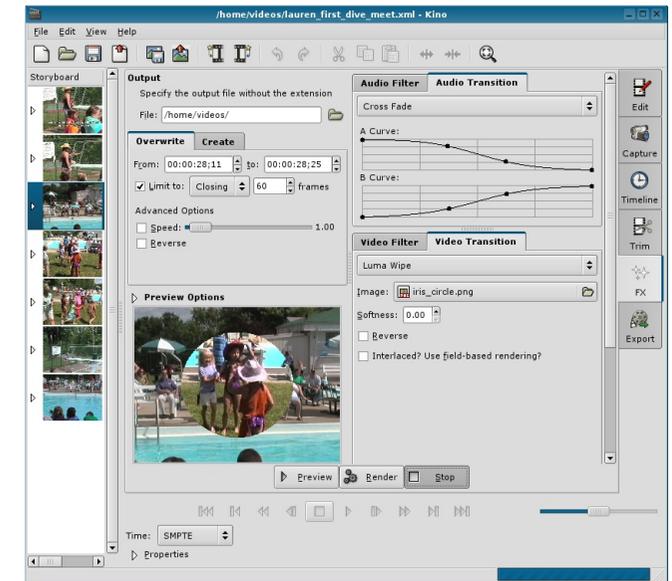
SOFTWARE REVIEW: VIDEO EDITORS

KINO

<http://www.kinodv.org>

With excellent integration with IEEE-1394 for capturing videos, VTR control and recording back to the camera, multiple video clip support, and an extensive array of supported formats including DV AVI, MP3, Ogg Vorbis, MPEG, and RAW DV, Kino is another excellent option for Linux video editing. Built for the GTK+ toolkit, Kino also features an extensible project metadata system and multilingual support. If you're looking for a lightweight, but still full-featured video editor, this might be a good place to start!

To install Kino, use the `kino` package in [extra].



LIVES

<http://lives.sourceforge.net>

LiVES (or LiVES is a Video Editing System) is an easy to use, but incredibly powerful, video editor that mixes realtime video performance and non-linear editing into one quality application. Capable of editing almost any video format, stream input and output, perform realtime blending of clips, and utilize dozens of effects, LiVES is good enough for a professional VJ, while still being simple enough for a beginner.

To install LiVES, use the `lives` package in the Arch User Repository.

