



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 - April 2009

EDITORS DESK .....	2
COMMUNITY NEWS .....	3
CHOOSING A DE/WM .....	4
HARDENING SSH IN 60 SECONDS .....	6
GAMERS CORNER .....	9
TIPS & TRICKS .....	10
PIMP MY ARCH .....	11
SOFTWARE REVIEW .....	12
Q&A .....	14



# EDITORS DESK

## Welcome to the first issue of Arch User Magazine!

**H**ello, and thank you for picking up issue #1 of Arch User Magazine! While the vast majority of you probably know me (or have at least seen me around the forums), I feel that I should take a moment to introduce myself.

My name is Daniel Griffiths, and I am a 26-year-old independent contractor in Delaware, US. Throughout my life, I have wandered through various UNIX/Linux systems including (but not limited to) MINIX, RedHat, Mandrake, Slackware, Gentoo, Debian, and even two home made distributions based on Linux From Scratch. I finally found Arch in 2007 and instantly fell in love with its elegant simplicity.

Some of our more attentive readers may note that Arch already has a monthly newsletter. With the existence of the aforementioned newsletter, what is the point of adding another news medium to the mix? Fear not, newsletter readers, I have no intention of letting Arch User Magazine take the place of the newsletter. In fact, Arch User Magazine and the newsletter are intended to fill two very different needs in the Arch community. While the newsletter gives you a glimpse into the lives of the Developers and the status of the distribution itself, Arch User is designed to showcase the community itself.

As this is our first issue, I give you no assurances as to the continued release of Arch User Magazine. However, I do promise that as long as you, the community, continue to express interest, I will continue to write.

Daniel Griffiths (Ghost1227)  
Editor, Arch User Magazine  
griffithsdj@archusermagazine.org

## ARCH USER STAFF

Daniel Griffiths (Ghost1227) ..... Editor  
David Crouse (Crouse) ..... Contributor

## WHAT IS ARCH LINUX?

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.



# COMMUNITY NEWS

## Become a CLI black-belt

Whether you are a struggling newbie or a veteran CLI aficionado, most everyone would agree that knowing the command line is an important part of being a Linux user. But just how well do you really know it? Hopefully, every Arch user knows the "cd" command. Even the youngest users quickly learn that typing "cd ~" returns you to your home directory. But how many of you knew that the similar command "cd -" will return you to your previous working directory?

Most experienced users will recognize that prefixing a command with "!" will recall the last used instance of the referenced command. For example executing "!echo" will retrieve the last time "echo" was used and rerun it. This is a wonderful trick, but some users may be hesitant to use it for fear of accidentally running the wrong command and damaging their precious system. Thankfully, there is an easy way around this! Simply by affixing ":p" to the end of the command it will simply be printed to the screen rather than actually run. What a lifesaver!

Do I have your attention yet? Command-

line-fu is a community-driven site that allows users and administrators alike to share their favorite command line gems and have them ranked by the community. If you're looking to expand your knowledge of the command line, this would be an excellent place to start!

**Source:**

<http://www.commandlinefu.com>

## Ada Lovelace Day

For those of you who hadn't heard, March 24th was Ada Lovelace Day. But what is Ada Lovelace Day? It is an international day of blogging to draw attention to women excelling in technology. Typically, Linux users (as well as developers in general) are known to be predominantly male. But did you know that the first computer programmer was actually female? During the early 1840s (yes, I said the 1840s) the Lady Lovelace began writing programs for her husband's proposed invention, the Analytical Engine.

Although Charles Babbage's Analytical Engine was never actually built, it has since

been proven that Ada's program for calculating a sequence of Bernoulli numbers would have run correctly, had it been given the chance.

The contributions of women in the field of technology all too frequently remain forever unacknowledged. However, the ranks of female developers, designers, tech journalists and consultants is steadily growing. A prime example of this unfortunate oversight is the well known blog Lifehacker (<http://lifehacker.com>). Lifehacker has quickly gained recognition and has been praised by several publications including TIME, Wired and PC Magazine. But did you know that Lifehacker was developed by female programmer Gina Trapani? I didn't.

**Source:**

<http://www.findingada.com>



# CHOOSING A DE/WM

BY DANIEL J GRIFFITHS

At some point in time in the life of every Linux user, you will inevitably come across a situation in which you become disillusioned with whichever Desktop Environment or Window Manager you are using. Maybe you have been using KDE and want to slim your system down, or maybe you use XMonad and are missing the pretty 3D effects that some of the bulkier options offer. Regardless of your situation, you may find yourself asking the age-old question, 'which option is the best for me?' The recommended way to determine your best fit is to try them and decide for yourself, but there are a lot of options out there and it can be a daunting (and time consuming) task. Henceforth, I am going to try to help you narrow down your list of candidates.

The first step in deciding what best suits you is determining what you need. Are you looking for fancy graphics and compositing? A Windows-like feel? Or are you trying for something more minimalistic? If you want a little help slimming down .your options, please read on.



Based on the QT toolkit, KDE (or the K Desktop Environment) is probably the single most complete DE available for Linux today.

However, its complexity comes at a price. With a minimum install size of 210MB, KDE weighs in as the largest mainstream DE out there. Unlike Windows, that size isn't all unnecessary bloat. With its focus on ease of use and outstanding graphic design, not to mention a well supported and complete suite of prepackaged applications, KDE is an ideal system for beginners or users who prefer eye-candy to simplicity.

If you like the idea of a full DE, but not having such a large install, GNOME might be more to your liking. At roughly 180MB for a minimum install, GNOME still packs a punch but is definitely a step in the right direction for the space-conscious. Based on the GTK toolkit, GNOME shares a similar goal with KDE, allowing a fully-featured interface with lots of eye-candy yet keeping a slightly slimmer footprint and requiring less RAM to run smoothly.



A third option among the ranks of fully featured DEs is XFCE. Based

designed to be a minimalistic alternative to KDE or GNOME. Weighing in at a tiny 15MB, XFCE is designed for productivity while also giving low memory systems all the advantages of a full DE without the cost.

While it is technically a window manager rather than a desktop environment, another full featured option is Enlightenment.



Enlightenment features advanced graphics libraries (read plenty of eye-candy) and an impressive install size of only 15MB. Unfortunately, it also relies on a proprietary toolkit, requiring you to additionally install GTK or QT to use most of your favorite programs. Another downside is a very long release cycle and occasional stability issues. If you want something that doesn't require much tinkering, stay away from this one.



In the category of floating window managers, none is more popular than the \*box family. Built off of the Blackbox framework, the most popular variations today include Fluxbox and Openbox. While the default installation of Blackbox and its derivatives skips things that many users are accustomed to (such



# CHOOSING A DEWM BY DANIEL J GRIFFITHS

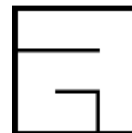
oversights are easily correctable through a plethora of available extensions, addons, and scripts. If you are looking for a minimalistic system without losing the floating window manager feel, the \*box family may well fit the bill.

DWM is probably the smallest window manager available, boasting less than 2000 lines of code. Unfortunately, it's miniscule size comes at a cost. All user configuration is done through the modification of a single header file. What does this mean? It means that the slightest change in configuration requires a recompile of the WM, making this an unlikely option for all but the most dedicated tinkerers.



Among the ranks of tiling window managers, XMonad is probably the most popular. Originally based on DWM, XMonad has far outdone its predecessor, allowing advanced configuration and dynamic extensibility through the functional programming language Haskell. While its dependancy on Haskell makes XMonad the largest of the tiling window managers, its relative stability and extensibility make it an excellent choice for someone new to tiling.

Similarly to XMonad, the Awesome window manager was originally a fork of DWM. Also like XMonad, it has far superseded its predecessor. Although its ever-changing lua-based configuration Awesome can be frustrating for new users, Awesome boasts a few firsts among tiling window managers (use of the XCB library to communicate with the X server, support for the Pango markup language and D-Bus) and a very dedicated user base.



So what is the best solution for you? You tell me! Decide what is most important to you, try out the available options, and maybe even learn a little bit along the way. Of course, if you want to be really hardcore, get rid of the graphical interface altogether and rely solely on screen and the CLI interface. What's screen, you ask? Check back with us next month for an article on the wonders of GNU screen and just what it can do!

Daniel Griffiths is Editor of Arch User Magazine, a dedicated Arch user himself, and uses the Awesome WM. For more info, head over to his website at <http://ghost1227.com>

Want an Arch USB Key?



Stop by Dusty's Schwag Shop at <http://schwag.archlinux.ca> for this and other great finds!



# HARDENING SSH IN 60 SECONDS

BY DAVID CROUSE

Before we even get started, it should be known that the number one security measure you can have in place is a good username and password.

From the OpenSSH website:

*"OpenSSH is a **FREE** version of the SSH suite of network connectivity tools that increasing numbers of people on the Internet are coming to rely on. Many users of telnet, rlogin, ftp, and other such programs might not realize that their password is transmitted across the Internet unencrypted, but it is. OpenSSH encrypts all traffic (including passwords) to effectively eliminate eavesdropping, connection hijacking, and other network-level attacks."*

Installation of OpenSSH on Arch Linux is extremely simple. As root, issue the command **pacman -S openssh**.

Open the `/etc/rc.conf` file and in the daemon section add `sshd` to start the OpenSSH server daemon at boot time. Now issue the command `/etc/rc.d/sshd start` to start the daemon immediately. You now have OpenSSH installed and running, but you should take the time to secure it as well! The Arch Linux wiki has a good page

on the installation of OpenSSH at <http://wiki.archlinux.org/index.php/SSH>. If you are totally unfamiliar with OpenSSH, read the wiki page to become more familiar with what it is and what it does. There are a couple more files to configure in order to make OpenSSH behave as expected, most notably `/etc/hosts.allow`. This configuration file has the power to limit who can and can not access your system via ssh, but for this article we will concentrate on configuring one file, `/etc/ssh/sshd_config`.

Some of us live in a command line world, some don't. As an Arch user you are probably at least somewhat familiar with the command line, as it is almost a prerequisite for this distribution. Sometimes things do get overlooked though, even by a command line junkie. Security of your OpenSSH server is one of those things that doesn't always get quite as much attention as it probably should. Let's face it, there are just too many other fun things to do with your machine. So in this article, we offer some very simple suggestions for hardening OpenSSH. It doesn't take very much effort, or time, on your part to make a huge difference in the security of your Arch Linux system. In fact, making a few simple changes can greatly improve the security of

your machine, and exponentially decrease the chances you will ever have your system compromised by crackers. How easy is that? Edit one configuration file, harden your machine to the point of making it a useless target, all in less than 60 seconds. Sound too good to be true? Ok, so maybe 60 seconds is a stretch if you have never edited this file before, but it really can take less than 60 seconds to make the changes. Now that you have your goal set for you, read on.

Here is an example that will be an eye opener for some of you.

```
less /var/log/auth.log | grep fail  
or  
lastb
```

If your machine is connected to the net and not behind a firewall, you're in for quite a surprise. I did a clean install once and let it run for five days without a firewall. Below are the results.

```
wtmp begins Thu Jul 17 21:57:08 2008  
[root@VistaCrusher1 ~]# lastb | wc -l  
25349
```

This means that there were 25,349 failed



# HARDENING SSH IN 60 SECONDS BY DAVID CROUSE

login attempts on my machine. No one ever got in, but it was very interesting to view the logs and grep the most used IP to see who was doing what.

Your best friend when locking down OpenSSH is the default `sshd_config` file. This file is normally located at `/etc/ssh/sshd_config` on Arch Linux machines. Before you do ANYTHING else, it is suggested that you backup your configuration file. This can be done quite simply by issuing the command `cp /etc/ssh/sshd_config{-bak}`. This makes quick work of starting over if something goes horribly wrong. Hey, it happens to the best of us, and it is better to be overly cautious than to be reckless and lose time and data trying to recover the file you just screwed up. So now that we have our configuration file backed up, let's start editing the `/etc/ssh/sshd_config` file!

Stopwatches please...

## **SSH TWEAK #1 - Port**

...and in my humble opinion, the very first thing you should change.

Change the port number from 22 to something else.

```
#Port 22
Port 20044
```

Every port scanner on the planet knows that port 22 is the standard ssh port. In fact, changing this one line will stop 99% of the port scans and hacking attempts through ssh. Yea, I sort of made up that number, but I know I'm close. Changing the default port to a five or six digit port can take you from 25,000+ failed login attempts in five days, to zero in over two years. This doesn't affect the usefulness of your ssh server at all, and about the only thing it might require is for you to forward a router port to that machine. In fact, I do this intentionally so that I can login to my three machines easily. I set the port numbers for ssh to three different numbers and have my router forward that port to the correct machine.

## **SSH TWEAK #2 - PermitRootLogin no**

Look for the line that says "PermitRootLogin yes". Uncomment this line and change it to no!

```
#PermitRootLogin yes
PermitRootLogin no
```

There is no sense in making it easy for

anyone. Authentication requires both a valid username and valid password. If they can log in as root, half the battle is over for them and they just have to find a correct match on a password. Make them guess both, and if they manage to get in as a user they still don't have root permissions. Don't make it easy. Remember everyone preaching about "don't run as root?" Well don't login via ssh as root either. Setting this line explicitly prevents this.

## **SSH TWEAK #3 - MaxAuthTries 3**

Look for the line that says "#MaxAuthTries 6." Uncomment it and set it to 3.

```
#MaxAuthTries 6
MaxAuthTries 3
```

After three unsuccessful login attempts... well, you get the idea. If you don't get it right in three attempts, you're out of luck.

## **SSH TWEAK #4 - MaxSessions 10**

Look for the line that says "#MaxSessions 10" and uncomment that as well.

```
#MaxSessions 10
MaxSessions 10
```

This option limits the maximum number of



# HARDENING SSH IN 60 SECONDS BY DAVID CROUSE



**2009.02 ISO  
RELEASED!**

## Features:

- **Kernel 2.6.28**
- **Full ext4 support**
- **ISOLINUX version**
- **Updated documentation**
- **Next generation installer**

*Arch installation CDs are for new installations only.  
To upgrade an existing installation, use `pacman -Syu`.*

**GET IT NOW!**



unauthenticated sessions. This will help keep someone from attempting to DDOS the server via ssh.

### **SSH TWEAK #5 - AllowUsers**

Set this. This one isn't in the `sshd_config` file by default, but it's one of the better things to look at setting explicitly. Setting it only allows listed users to login via ssh.

```
AllowUsers johnsmith
```

This will allow *ONLY* johnsmith to login via ssh. Everyone else is denied access. AllowUsers is a list of username patterns, separated by spaces. So, if you have multiple users, the config file would look something like:

```
AllowUsers johnsmith user3 admin
```

### **SSH TWEAK #6 - Protocol 2**

Find and uncomment the protocol line and set it to 2.

```
#Protocol 2,1  
Protocol 2
```

This means that only protocol 2 will be used, since protocol 1 is considered somewhat insecure.

So there are a few simple configuration changes that are guaranteed to make your box much safer. And that took what, 60 seconds to change those lines in `sshd_config`? This is by no means the extent to which you can configure your OpenSSH server. First, read *man sshd\_config*. I know, a big "duh," but I would be remiss if I didn't include that statement. There are lots of other cool things you can do in there, but I'll leave that for you to discover. Just read the man page.

Hopefully this article has provided you with some useful tips in securing your OpenSSH server. In all reality, it probably took you longer to read this than it will to edit your configuration file, so get to it!

David Crouse is the founder of the USA Linux Users Group (<http://usalug.org>) as well as setting up the `archlinux.us` webmail service for Arch users. Keep an eye on AUM for more articles by David in the series "Living at the Command Line".





# GAMERS CORNER

One of the first things that every new Linux user asks is "where are the games?" Unfortunately, the gaming industry has always been the single greatest stumbling block for the Linux community. Sure, we've always had our share of games, but with the plethora of 3D games available on Windows, somehow classics like abuse and nethack just aren't cutting it any more. And yet, there is hope! The Gamers Corner section of Arch User Magazine will be used to showcase some of the incredible games that are starting to appear on the Linux horizon.

For Issue 1, we will be reviewing the first-person shooter Smokin Guns. Based on the Quake 3 engine, Smokin Guns is a remarkable simulation of the old west. Six-shooters, long rifles, stagecoaches and bank robberies are all faithfully recreated in three dimensions. One of the most unique aspects of this game is a money system that allows for equipment purchase with reward money.

Like most games of its type, Smokin Guns features both single and multiplayer gameplay. Unfortunately, the single player mode has been severely neglected. Lacking any sort of plot or any real quests,



the single player mode is little more than a deathmatch game.

Thankfully, Smokin Guns more than redeems itself with an outstanding multi-player mode. After logging in (or starting a new game server), you will find yourself at some random location on the map. Regardless of what team you are playing on, or which map has been selected, you always start with a Remington 58 six-shooter and knife. Once you have made a

bit of a name for yourself (through gunfighting, obviously) you will be able to obtain new equipment through the in-game purchasing system. One of the first things you will probably want to pick up is a good boiler plate. While it isn't modern body armor, a boiler plate will keep a few torso shots from hitting you, and in this game that can make a huge difference as there are no health packs to be found in the Wild West.

Aside from the traditional deathmatch and team deathmatch game modes, Smokin Guns includes a bank robbery mode and a duel mode which ensure that the gameplay is plenty varied.

Smokin Guns weapons, maps, models and sounds all contribute to an incredible gaming experience. Whether you are a fan of classic westerns, or are simply looking for a new challenge, this game is definitely worth a look!



# TIPS & TRICKS

## CREATING SECURE PASSWORDS

So everyone knows that passwords like "bgAitb;Wgft!" or "emTpi4ma@a.us" are fairly safe, but who can remember them? You can! But how you ask? Your passwords are crucial to maintaining the integrity of your system and private data. Unfortunately, all too often users forget their passwords. As a result, many people use simple passwords such as a birthday, address, pet name or similar.

To anyone who might wish to gain access to your system, this is akin to leaving the key under the welcome mat. Without any real effort, a hacker can discover your basic information and try each of these potential passwords.

A second problem that affects even secure passwords is that users frequently use the same password for everything. This means that if someone does gain access to one piece of the puzzle, whether it be the actual computer, an email account, blog, or whatever, the rest of the pieces just fall into place and suddenly your life is an open book.

So how does one come up with an easy to

remember, convertible password without sacrificing security? Here is one possible option; build each password from two parts.

### **PART 1 - STATIC COMPONENT**

You can make your password more secure by mixing different types of characters. A jumble of uppercase and lowercase, numbers and special characters can make a password exceedingly difficult to figure out. One way to accomplish this while making it easy to remember is pick a sentence and use the first letter of each word to create your static component.

**"This password is for my account at  
archlinux.us"  
becomes  
"Tpi4ma@a.us"**

### **PART 2 - DYNAMIC COMPONENT**

This component can be easily defined by what the specific account is used for. In other words, for an email account, you might use something like "em" or "mail", a bank account could be "bk" and a blog could be "bl". You get the idea. Now that you have the two components of your

password, it's just a matter of putting them together to come up with a password that is both easy to remember and secure!

Finally, don't forget to regularly change the static component of your password. Using this method, there are a virtually infinite pool of possibilities from which to draw so there should never be a need to re-use a password. Just be creative!

Got a useful tip or trick you'd like to share with the community? How about an idea for an article or a particularly nifty screenshot? Drop us a line!

#### **Gamers Corner**

gamerscorner@archuser.com

#### **Pimp My Arch**

pimpmyarch@archuser.com

#### **Tips & Tricks**

tipsandtricks@archuser.com

#### **General Submissions**

submissions@archuser.com

#### **Corrections**

corrections@archuser.com

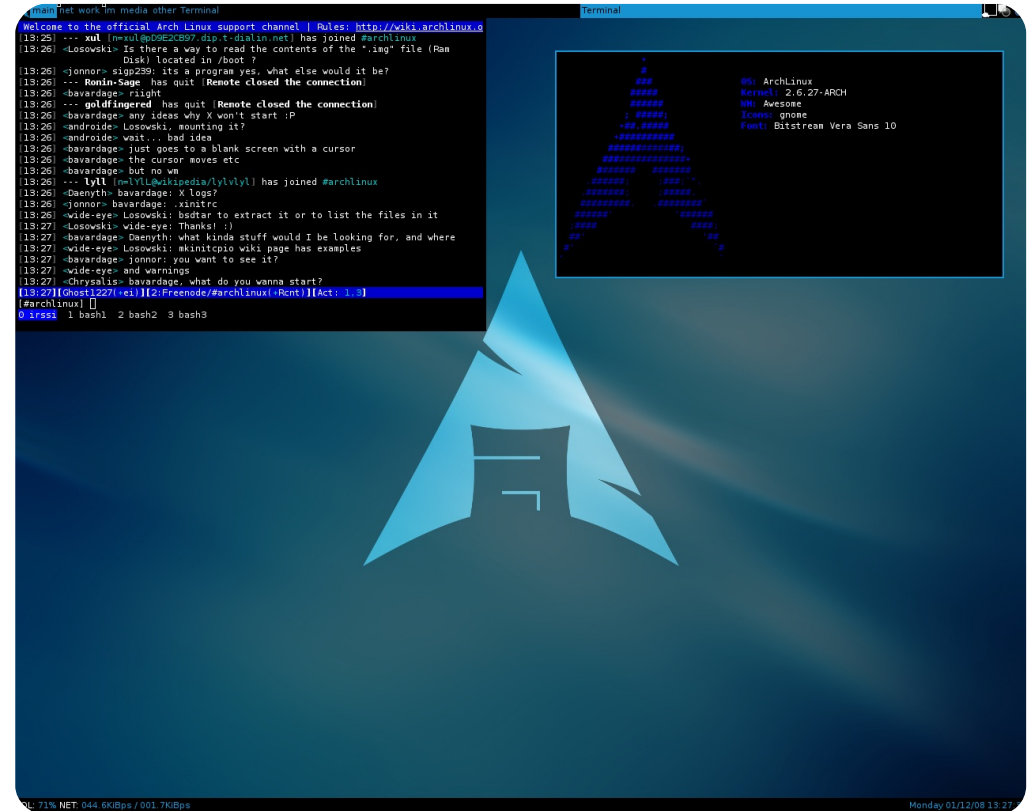


# PIMP MY ARCH



I use GNOME with the Clearlooks-Darklime GTK theme and Oxygen-Refit 2 icons. Also in use are GNOME-Do, Docky, Compiz, and Screenlets.

**Matt Runion (mrunion)**



I use Awesome 3.1 with the standard GNOME iconset (sad, I know). The wallpaper is one of mine called "Awesome Arch" and is available from my website (<http://ghost1227.com>). All of my configs are available from github (<http://github.com/ghost1227>).

**Daniel Griffiths (Ghost1227)**



# SOFTWARE REVIEW: TEXT EDITORS

## SCITE

<http://www.scintilla.org/SciTE.html>

This SCIntilla based editor was originally built solely to demonstrate SCIntilla. Since its humble beginnings it has grown to be a generally useful editor with development support for building and running programs as well as detailed syntax highlighting. Built on the GTK+ toolkit, SciTE is built to be cross-platform, functioning well on both various Linux distributions and Windows. Having virtually no dependencies and featuring an incredibly useful tabbed interface, SciTE is the perfect option if you are looking for a full-featured, yet minimalistic text editor.

To install SciTE, use the ``scite`` package in [community].

## BLUEFISH

<http://bluefish.openoffice.nl>

Bluefish is a powerful editor designed for developers and web designers. Featuring syntax highlighting for an incredible array of languages, Bluefish strives to be a lean and clean editor and boasts that it uses 55-70% less memory than comparable editors. Bluefish employs a multiple document interface, project support, tag auto-completion and built-in support for remote editing through gnome-vfs. For those of us who constantly like to tinker, Bluefish allows the user to specify custom dialogs and keybinding to speed up the development process. External compiler integration and a built-in function reference browser make this a formidable editor. If you're looking for simplicity, this definitely isn't it. However, if you need a full-featured development environment (or just want to try something different), than Bluefish just might be right for you!

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

## GEANY

<http://geany.org>

Geany is a lightweight Integrated Development Environment designed to provide a small and fast IDE with minimal dependencies. Featuring syntax highlighting, code folding, auto-closing tags, code navigation, a built-in project manager and a complete build system, Geany somehow manages to fit an impressive array of features into a relatively small package. Extendable through plugins, Geany ships with plugins that provide a classbuilder, export functionality, a sidebar filebrowser, HTML characters, and window splitting. If that isn't enough, several additional plugins are easily available through their website. Known to run under Linux, BSD, MacOS X, AIX, Solaris and Windows, Geany is said to run on every platform that supports the standard GTK libraries.

To install Geany, use the ``geany`` package in [community]. Several plugins are also available in the AUR.



# SOFTWARE REVIEW: TEXT EDITORS

## GEDIT

<http://projects.gnome.org/gedit>

As the official text editor of the GNOME desktop, most users should recognize this one. Aimed at simplicity and ease of use, gedit is a powerful general purpose text editor. Its feature set currently includes configurable syntax highlighting, remote editing, auto indentation, line numbers and font support, although it is extendable through the use of a flexible plugin system.

To install gedit, use the ``gedit`` package in [extra]. A set of plugins is also available for gedit through the official repositories. You can install them by using the ``gedit-plugins`` and ``gedit-plugins-extra`` packages in [community]. Additional plugins may be available through the AUR.

## KATE

<http://kate-editor.org/kate>

Kate is a multi-document editor based on a rewritten version of kwrite. Now an official part of KDE, Kate offers all the features of its predecessor plus a slew of new ones. Featuring window splitting and tabbing, syntax highlighting, code folding, block selection, auto indentation and completion and an integrated command line, Kate packs quite a punch. Like many other editors, Kate features a built-in plugin architecture, so users can extend the already impressive feature set to suit their individual needs. If you are using GNOME or a similar DE, or if you are looking for something minimalistic, this is definitely not the editor for you. On the other hand, if you are already a KDE user, you probably won't need to look further than this for your editing needs.

If you use KDE, Kate is installed by default. If not, you will have to install the ``kdesdk`` package in [extra]. Several plugins are available through the AUR.

## VIM

<http://www.vim.org>

Although we have focused thus far on GUI editors, this review would be incomplete without mentioning Vim. Vim (Vi IMproved) is the de-facto favorite text editor for developers and \*nix users. Although it is notorious for its steep learning curve and less-than-friendly user interface, vim maintains a radically loyal fanbase due to its efficiency, customization, and extensibility. If you are looking for features more commonly found in a word processor like font support or WYSIWYG editing, steer clear of this one. On the flip side, users might want to keep this one on hand to aid in the occasional system SNAFU. Among the ranks of console editors, vim is definitely the best.

To install vim, use the ``vim`` package in [extra]. Numerous plugins are available through [extra], [community] and the AUR.



## **WHY AREN'T THERE ANY QUESTIONS HERE? DOESN'T THAT DEFEAT THE PURPOSE OF A QUESTION AND ANSWER SECTION?**

Why yes! It does defeat the purpose of a question and answer section! However, given that this is the first issue and no questions have been asked, we have nothing to answer! If you have an Arch related question (or Arch unrelated for that matter) feel free to ask!

## **ONLY FOURTEEN PAGES? ISN'T THAT A BIT SHORT FOR A MAGAZINE?**

Once again, as this is our first issue, we don't have a lot of content. As of now, Arch User is working with a bit of a skeleton staff. If you think you'd like to contribute (as a regular contributor or just passing along an idea) please feel free to contact us!

# **THIS SPACE FOR RENT**

**No, this space isn't really for rent. However, given the lack of questions (or submissions in general for that matter... what did you expect? Issue 1 is expected to be small) I had a nice, big space to fill. I figured that at the very least, this would catch your attention and show you just how important your support is to the continuation of Arch User Magazine. Please! Contribute!**