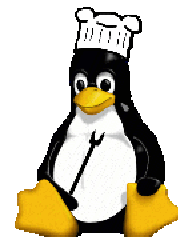


# The Open Pitt



What's cooking in Linux and Open Source in Western Pennsylvania

Issue 42

July 2010

www.wplug.org

## Book Review: *Professional WordPress* by Bobbie Lynn Eicher

Authors: David Damstra, Hal Stern,  
Brad Williams  
Publisher: Wrox Press  
ISBN13: 978-0-470-56054-9  
\$44.99, 408 pages, 2010

WordPress is one of the most popular software platforms for blogging on the market. It is developed under the GNU General Public License (GPL) and can be downloaded for free at <http://www.wordpress.org/>.

*Professional WordPress: Design and Development* is focused on explaining the workings of WordPress from the point of view of a professional programmer. This is a book for people who don't need several pages of explanation on subjects like what HTML or a blog is before moving on to more technical topics. For someone who already knows the basics, it's refreshing not to have to skim over chunks of a book that are meant for complete novices.

The authors start out with an overview of what WordPress is and how to use it. The book then moves on into explanations of the internal code, and after that features chapters on how to make additions like plugins and themes. The final section is discussion about optimization, security, and other issues related to the performance of a finished web site.

The main thing that you need to be wary of with this book is that it was published a couple of months before the 3.0 release of WordPress. As a result, it primarily covers the 2.9

release, and some of the features may not work in exactly the same way in 3.0. Generally the target audience for this book is probably savvy enough to deal with the differences, but it does add extra work to the process. It's sad that such a great book was released at a point where it became out of date so quickly.

I also would have liked to see a clearer explanation of the licensing of themes. In the explanation of the GPL at the beginning of the book, the authors do mention that the code portion of themes inherit the GPL from WordPress. However, in the chapter about themes they somewhat confuse the issue when saying that you should check the licensing on a theme before modifying it. While this is true, since some portions of a theme may not necessarily be under the GPL, the subject should have been covered more clearly. In their brief descriptions of several of the popular premium themes, it would have also been nice to see at least a brief mention of ongoing controversy about whether the Thesis theme is in violation of the GPL.

The greatest strength of this book is the attitude that it brings to development. The authors obviously share in the values of the open source community. When they're explaining why it's best to use semantic HTML, the first reason offered is that "it is the best thing for the future Web."

They also go on to discuss the more immediately pragmatic reasons, but the fact that they take the time to talk about how site design can influence the future of Internet technology made me very happy. It was also great to see a section giving advice on how to contribute to the development of WordPress itself.

On the whole, this book is written very well and a lot of thought was put into the way that it is structured. It's still a great guide to what WordPress is and how it works, so long as you're willing to work around the fact that some of its material is focused on the previous version of the software.

---

*Bobbie Lynn Eicher is a long-time member of WPLUG and holds a B.S. in Computer Science from the University of Pittsburgh.*

---

*Wrox Press provided a free copy of Professional WordPress to Bobbie so that she could write this review. There was no other compensation involved.*

### Coming Events

**Jul. 31:** Installfest. 10AM to 3PM,  
Northland Public Library

**Aug. 14:** General User Meeting.  
10:30AM to 12:30PM, Wilkins  
School Community Center

**Aug. 29:** 9th Annual Open  
Source Picnic. Snyder Park,  
Whitehall

**Sep. 18:** General User Meeting/  
Nominations. 10:30AM to  
12:30PM, Wilkins School  
Community Center

**Oct. 9:** General User Meeting/  
Election. 10:30AM to 12:30PM,  
Wilkins School Community  
Center

---

*The public is welcome at all events*

---

## July Roundup

*Jul. 10 General User Meeting:* **Toby Rule** spoke about creating extensions for OpenOffice.org. He went through the fundamentals of laying out and writing an extension in OpenOffice BASIC. He then covered some of the reference documents for the applica-

tion programming interface. This was all demonstrated by writing a sample extension and showing each step along the way. Toby finished by discussing CompPad, a Java-based extension he develops that allows equation solving and plotting.

## UNIX Curio

This series is dedicated to exploring little-known—and occasionally useful—trinkets lurking in the dusty corners of UNIX-like operating systems.

Hopefully it doesn't seem like I'm picking on *Linux Journal*, but like two months ago, this column has been inspired by an article of theirs <http://www.linuxjournal.com/content/treating-compressed-and-uncompressed-data-sources-same>. The author was demonstrating a clever *bash* script that would take a filename and send the file to standard output or, if the filename ended in *.gz*, decompress it and send the result to standard output. Slightly rearranged, he had:

```
F=`echo $1 | perl -pe
's/.gz$/''
if [[ -f $F ]] ; then
  cat $F
elif [[ -f $F.gz ]] ; then
  gunzip -c $F
fi
```

He took some heat on the web site and in letters to the magazine for cranking up a whole Perl interpreter just to chop the *.gz* off the end of a filename. Our curio for today is a standard UNIX utility made for just this purpose called *basename*. Along with its brother *dirname*, it is used to pull apart pathnames to get the part you want. What *basename* does is remove any leading path on the name given to it, and if a suffix is specified as well, removes that also. If a directory path with a trailing slash is given, it returns the last part with no slashes. Here are some examples:

```
$ basename /bin/gzip
gzip
$ basename /bin/gzip .so
gzip
$ basename
/usr/lib/libz.so .so
libz
$ basename /usr/lib/
lib
```

The counterpart, *dirname*, does essentially the opposite. It removes the last part of the pathname and returns a directory name (with no

trailing slash):

```
$ dirname /usr/lib/libz.so
/usr/lib
$ dirname /usr/lib/
/usr
$ dirname file_in_this_dir
.
```

So we can replace the first line of the script up top with `F=`dirname $1`basename $1 .gz``, get the same result, and be sure it will work on any UNIX-like system, no Perl necessary. The more observant among you may be thinking "sed could do that, too!" and you're right; `F=`echo $1 | sed "s/\.gz$//"`` also would work anywhere.

One might suspect that as a general-purpose text processor, *sed* would be slower than *basename* and *dirname*. To see how they compared, we ran each method against a randomly-generated list of 5,000 filenames. Turns out the critics were right, as Perl ran the longest at 59 seconds. Using *basename/dirname* took 44 seconds—a nice improvement, but *sed* blew past it at 34 seconds. Probably the fact that only one call to *sed* was needed versus two for *basename* and *dirname* made the difference.

Helpful suggestions in response to the article revealed a shell curio. You may have seen the brace syntax for parameters. For example, to show a filename `$F` with an "X" appended, you can't use `'echo $FX'` because that means a parameter named `FX`. Instead, you'd use `'echo ${F}X'` and the shell only interprets what's inside the braces as the parameter name.

Modifiers can also go inside the braces and one of these, `%`, is just what we need to chop off that extension. This works in *bash*, *zsh*, and any shell conforming to the current POSIX standard, but not *csh* or friends or older implementations of the Bourne shell. We can rewrite the first line of the original script as simply `F=${1%.gz}` and forgo any outside utilities. Performance? Under half a second to process those 5,000 filenames. Not bad at all.

The Open Pitt is published by the Western Pennsylvania Linux Users Group  
<<http://www.wplug.org/top>>

Editor: Vance Kochenderfer

### What is Linux?

Linux is a *kernel*, the core of a computer operating system, created by Linus Torvalds. It is typically packaged as a *distribution*, which includes the extra programs necessary to make a computer functional and useful. Since 1991, it has grown from a one-man project which ran on one computer to one with thousands of contributors running on everything from mobile phones to million-dollar supercomputers.

### What are Open Source and Free Software?

Open Source and Free Software provide you, the user, with the opportunity to see the source code of the programs you use. You are free to use it, share it with others, and even make changes to it if you wish. While the Free Software and Open Source communities differ in their philosophical approach, in practical terms they share nearly identical goals. Learn more at <http://www.opensource.org/> and <http://www.gnu.org/>.

This newsletter was produced using Open Source and Free Software.

Copyright 2010 Western Pennsylvania Linux Users Group. Any article in this newsletter may be reprinted elsewhere in any medium, provided it is not changed and attribution is given to the author and WPLUG.

## Nominations Upcoming

The Western Pennsylvania Linux Users Group will be electing a new Board of Directors in October. But before then, we need to hold nominations so we can have a set of names to put on the ballot! You can help by:

- Signing up to be a WPLUG member
- Coming to the September 18 nominations meeting
- Nominating other members who you think would make good officers
- Nominating yourself, also known as volunteering!

If you're not sure whether you are currently a member or when your membership expires, you can contact WPLUG Secretary David Ostroske at [eksortso@gmail.com](mailto:eksortso@gmail.com) to find out.

You can nominate up to five people, but feel free to think of more names—someone else may pick one of yours. Hope to see you there!