MAGAZINE BSD

FOR NOVICE AND ADVANCED USERS

Adobe® PDF Magazine Version

FREEBSD: PORTABILITY WITH VMWARE

INSIDE

PACKAGE MANAGEMENT FOR THE UPCOMING PC-BSD 9

MUTT ON OS X PART II

INTERVIEW WITH DRU LAVIGNE

CONTENT MANAGEMENT MADE EASY THE OPEN SOURCE WAY!

OPENBSD IMPROVES UPON /ETC/RC.D/

BUILD APPLIANCES WITH QEMU AND OPENBSD

DRUPAL ON FREEBSD PART 5

BENCHMARKING DIFFERENT KIND OF STORAGE

WHY YOU USE FREEBSD JUST MAY START WITH A 'Z'

VOL.4 NO.4
ISSUE 04/2011(21)
1898-9144

10 Gig On Board

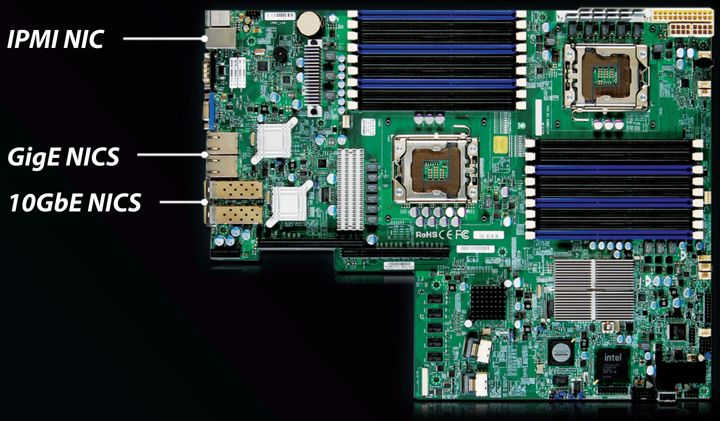
On-Board 10 Gigabit Ethernet Adapters leave your existing PCI-E slots available for other expansion devices.

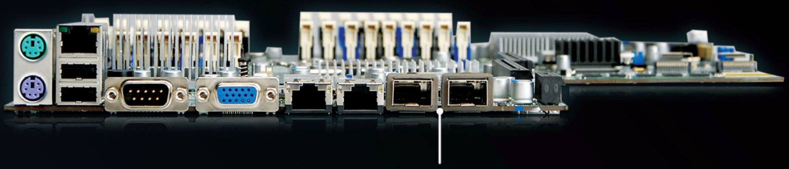
$ 30% cost savings/port over equivalent Dual-Port 10 GB PCI Express add-on card solution

Blazing Fast, Embedded 10Gb Ethernet

10G Rackmount Servers in the iX-Neutron server line feature the Intel® Xeon® Processor 5600/5500 Series, and come with 10GbE networking integrated onto the motherboard. This eliminates the need to purchase an additional expansion card, and leaves the existing PCI-E slots available for other expansion devices, such as RAID controllers, video cards, and SAS controllers.

For more information on the iX-1204-10G, or to request a quote, visit: **http://www.iXsystems.com/neutron**





KEY FEATURES:

- Supports Dual 64-Bit Six-Core, Quad-Core or Dual-Core, Intel® Xeon® Processor 5600/5500 Series
- 1U Form Factor with 4 Hot-Swap SAS/SATA 3.5" Drive Bays
- Intel® 5520 chipset with QuickPath Interconnect (QPI)
- Up to 192GB DDR3 1333/1066/800 SDRAM ECC Registered Memory (18 DIMM Slots)
- 2 (x8) PCI-E 2.0 slots + 1 (x4) PCI-E 2.0 (in x8 slot -Low-Profile - 5.5" depth)
- Dual Port Intel® 82599EB 10 Gigabit SFP+ - Dual Port Intel® 82576 Gigabit Ethernet Controller
- Matrox G200eW Graphics
- Remote Management - IPMI 2.0 + IP-KVM with Dedicated LAN
- Slim DVD
- 700W/750W Redundant AC-DC 93%+ High-Efficiency Power Supply

Call iXsystems toll free or visit our website today!

1-855-GREP-4-IX | www.iXsystems.com

Powerful.
Intelligent.

Dear Readers!

Make yourself comfortable and get ready for the April issue of BSD Magazine!

This month we start with an interview with our BSD Guru and probably the most famous person in the community: Dru Lavigne who answers the questions from Anton Borisov.
After that you will find another batch of articles and news written by BSD developers in a section under a new name „Developers Corner" – which is going to be a regular section of the magazine now.

As always you will find plenty of interesting and educational articles in „How To's" and „Let's Talk" sections: Another part of Drupal series as well as second article about Mutt, and Juraj Sipos will tell us how to transfer a physical partition with FreeBSD to a VMware image.

Enjoy your reading and as always – contact us with your feedback :).

Thank you!

Zbigniew Puchciński
Editor in Chief
zbigniew.puchcinski@software.com.pl

MAGAZINE BSD

Editor in Chief:
Zbigniew Puchciński
zbigniew.puchcinski@software.com.pl

Contributing:
Kris Moore, Ian Darwin, Josh Paetzel, Anton Borisov, Girish Venkatachalam, Bill Harris, Rob Somerville, Juraj Sipos, Michael Hernandez, Sufyan bin Uzayr, Svetoslav Chukov

Proofreaders:
Corby Agid, Melanie Vonfange

Special Thanks:
Denise Ebery, Matt Olander

Art Director:
Ireneusz Pogroszewski

DTP:
Ireneusz Pogroszewski

Senior Consultant/Publisher:
Paweł Marciniak *pawel@software.com.pl*

CEO:
Ewa Dudzic
ewa.dudzic@software.com.pl

Production Director:
Andrzej Kuca
andrzej.kuca@software.com.pl

Executive Ad Consultant:
Karolina Lesińska
karolina.lesinska@bsdmag.org

Advertising Sales:
Zbigniew Puchciński
zbigniew.puchcinski@software.com.pl

Publisher :
Software Press Sp. z o.o. SK
ul. Bokserska 1, 02-682 Warszawa
Poland
worldwide publishing
tel: 1 917 338 36 31
www.bsdmag.org

Software Press Sp z o.o. SK is looking for partners from all over the world. If you are interested in cooperation with us, please contact us via e-mail: *editors@bsdmag.org*

The editors use automatic DTP system AUPUS

Mathematical formulas created by Design Science MathType™.

Contents

Interview

Developers Corner

How To's

Let's Talk

Interview with
Dru Lavigne

Dru Lavigne is a network and systems administrator, IT instructor, author and international speaker. She has over a decade of experience administering and teaching Netware, Microsoft, Cisco, Checkpoint, SCO, Solaris, Linux and BSD systems. She is author of BSD Hacks, The Best of FreeBSD Basics, and The Definitive Guide to PC-BSD.

As Director of Community Development for the PC-BSD Project, she leads the documentation team, assists new users, helps to find and fix bugs, and reaches out to the community to discover their needs. She is the former Managing Editor of the Open Source Business Resource, a free monthly publication covering open source and the commercialization of open source assets. She is founder and current Chair of the BSD Certification Group Inc., a non-profit organization with a mission to create the standard for certifying BSD system administrators, and serves on the Board of the FreeBSD Foundation.

Dru, your name is known for almost everyone in FreeBSD community. Could you please tell about your very first acquaintance with FreeBSD?

I went back to school in the mid-90s to get a diploma in system administration and network engineering. A few months before graduation, I noticed in my job search that the interesting jobs wanted *Unix experience*. Since I had neither the Unix experience nor the funds to pay to learn about Unix, I did an Internet search for *free Unix*. The very first hit was for freebsd.org and I dived right in, trying to figure out how to install FreeBSD, how mailing lists worked, etc. I was hooked almost immediately as it offered so many interesting things to learn and figure out. Since then I've branched out to the other BSDs as well, but FreeBSD was my first acquaintance.

There exist many BSD-like, or if we take a broader look, UNIX-like systems – ranging from GNU/Linux to OpenSolaris. However, FreeBSD still has so many fans and keeps a strong community. Why it is so in your opinion?

Several reasons. Every operating system struggles with the balancing act of maintaining stability/compatibility and introducing new features/innovations. FreeBSD maintains a pretty good balance between the two: it is rare for a new version to break something that previously worked and new features tend to be well tested before they are released. Tending towards stability can mean that new drivers and features aren't immediately available in a released version (though they are always available for bleeding edge users and testers to try out); on the flip side, users aren't left wondering *what did they change on me now or what will be broken in this version?*. FreeBSD also provides some features that are unique or not widely implemented on non-BSD systems; examples include operating system-level virtualization through jails [1], continued ZFS support [2], security event auditing [3], Linux binary compatibility [4], high availability storage [5], and the GEOM modular disk transformation framework [6]. FreeBSD tends to be well documented. For example, most man pages contain usage examples (once you get used to these you really miss them when a man page does not); another example is that driver man pages (e.g. for Ethernet and wireless devices) contain the hardware models of supported devices. One can usually find what they are looking for in the man pages that come with the operating system; for users who wish to widen their knowledge or get a better understanding of the larger picture, the FreeBSD Handbook (*http://www.freebsd.org/handbook*) covers pretty well anything you'd ever want to do on a FreeBSD system. On the development side, the FreeBSD Developers Handbook [7] and Porters Handbook [8] are very comprehensive and detail the FreeBSD development process. There are also many opportunities to be mentored by more senior developers and to earn the right to commit code to the FreeBSD code base. Finally, some users are attracted to the BSD license and the BSD definition of *free*.

Every complex software product needs to have a strong funding behind the scenes. If we take Linux ecosystem, as an example, then such companies like Linux Foundation, RedHat, IBM, Google, and many others contribute resources to the project. What organization is responsible for the same tasks for FreeBSD?

That would be the FreeBSD Foundation, a registered US-based, non-profit organization dedicated to supporting and building the FreeBSD Project and community worldwide. The Foundation actively solicits donations from individuals and corporations and uses those funds to sponsor BSD conferences and to pay developers to work on projects that benefit the FreeBSD code base and FreeBSD users. Examples of previously or currently funded projects include support for flattened device tree [9], userland Dtrace support [10], native DAHDI driver [11], jail-based virtualization [12], 5 new TCP congestion control algorithms [13], improvements to BSNMP [14], and the implementation of GEM, KMS, and DRI support for Intel drivers [15].

I guess nobody can be surprised nowadays with labels Embedded Linux, Powered by: Linux, and Linux powered – these devices are network routers, digital modems, printers, as well as usual servers. Where could we find devices with Powered by FreeBSD stickers?

The Hardware Vendor List (*http://www.freebsd.org/commercial/hardware.html*) contains the names of many vendors who provide hardware or solutions for FreeBSD. Other notable vendors include Juniper Networks (Junos is based on FreeBSD), Check Point (IPSO is based on FreeBSD), Apple (MAC OSX includes a FreeBSD userland), Borderware appliances, Coyote Point load balancers, Dell iSCSI SAN arrays, IronPort (AsyncOS is based on FreeBSD), Isilon (now part of EMC), nCircle IP360, NetApp filers, SGI disk arrays, Sophos email appliance, and VXworks. Additionally, many hosting providers use FreeBSD and are consistently in the top 10 on the Netcraft Most Reliable Hosting Sites survey [16]. Looks like in February, 2011 the top 3 most reliable sites all run FreeBSD: Datapipe, Swishmail, and NYI.

In order to efficiently promote FreeBSD it's highly advisable to have certification centers, courses and so forth. Whether such organizations and events exist and where a person could obtain information about it in his/her region?

The BSD certification Group (BSDCG, *http://www.bsdcertification.org*), a registered non-profit founded in 2005, develops and maintains certification exams for system administrators of BSD systems. The exams themselves are psychometricly valid, meaning that they meet rigorous assessment standards. Yet, the BSDCG is community-based, meaning that they depend on the community to help organize exam events, assist in the creation of study materials, and to donate time and funds to keep the certification program going. Taking the exam (even if you don't need it for employment purposes) and helping to spread the word assist the certification program in creating value both within and outside the FreeBSD community. You can learn more about the certification program at its website and can register for upcoming events at *https://register.bsdcertification.org//register/events*. If you've ever wondered what goes into making a certification exam, check out the videos *BSD Certification Group: A Case Study in Open Source Certification* [17] and *BSD Certification Update: The Lab Exam* (*http://blip.tv/file/4844392*).

It is believed that FreeBSD is suitable only as server side operating system. Is this a myth?

FreeBSD makes an excellent server platform and it also makes an excellent desktop system. However, the default installation assumes that you wish to manually configure your system for its intended purpose. Desktop users who don't have the time or inclination to manually configure a desktop should check out PC-BSD (*http://www.pcbsd.org*). PC-BSD is a pre-configured FreeBSD desktop; this means that sound, networking, X, a desktop manager, flash, etc. are already setup and working for you immediately after installation. The current version of PC-BSD comes pre-configured with KDE and Fluxbox. Version 9 (due sometime in summer 2011) will let you select which desktop(s) to install both during and after the initial installation. PC-BSD also includes some features useful to desktop users that don't come with FreeBSD. You can learn more about PC-BSD in the PC-BSD Handbook [18] and keep up-to-date with its progress at the PC-BSD blog (*http://blog.pcbsd.org*).

If a company, or an enterprise decides to migrate existing local infrastructure onto FreeBSD, what would you advise to do first, and which common mistakes should be expected?

Any migration is going to require some research upfront, a transition plan, and a plan for ongoing support. If the company decides to do all of this in-house, they'll need a team who is familiar with FreeBSD and with its documentation resources, mailing lists, IRC channels, etc. Having someone on that team with a *commit bit* or who knows a FreeBSD committer can be beneficial if the migration requires any code customization. If the company is looking to hire consultants, they can find a list sorted by geographic area at *http://www.freebsd.org/commercial/consult_bycat.html*. Commercial support packages are also available, for example from *http://www.iXsystems.com*.

What technological achievements within FreeBSD do you consider to be outstanding during the last 5 years? And what new functionality do you expect to appear in the near 2 years?

Features in the past 5 years and not mentioned previously in this interview include:

- Superpages support
- SCTP (FreeBSD was the IETF reference implementation)
- Dtrace support (both kernel and userland)

Some things to watch for in the future:

- Capsicum (*http://www.cl.cam.ac.uk/research/security/capsicum/*)
- CLANG/LLVM (*http://wiki.freebsd.org/BuildingFreeBSDWithClang*) replacing GCC as the base compiler
- USB 3.0 support via the XHCI driver

Ivan Voras blogs regularly about FreeBSD's upcoming features (*http://ivoras.sharanet.org/freebsd/freebsd9.html*).

Sometimes developers are stumbled, either to choose GPL/LGPL, or stick with BSD license. Where's the optimal choice?

Licenses are a tool and ideally one is able to select the best tool for the job. When selecting an open source license, a general rule of thumb is as follows:

- choose a license without a strong copyleft clause (e.g. BSD, MIT, Apache) if you are seeking wide adoption of new functionality. For example, protocols and reference implementations are often licensed this way to encourage vendors to create products that implement the software. The more products that become available, the faster the software becomes used by a wide range of users.

Resources

[1] *http://en.wikipedia.org/wiki/FreeBSD_jail*
[2] *http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/filesystems-zfs.html*
[3] *http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/audit.html*
[4] *http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/linuxemu.html*
[5] *http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/disks-hast.html*
[6] *http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/geom.html*
[7] *http://www.freebsd.org/doc/en/books/developers-handbook/*
[8] *http://www.freebsd.org/doc/en/books/porters-handbook/*
[9] *http://wiki.freebsd.org/FlattenedDeviceTree*
[10] *http://wiki.freebsd.org/DTrace/userland*
[11] *http://freebsdfoundation.blogspot.com/2010/10/update-on-dahdi-project.html*
[12] *http://freebsdfoundation.blogspot.com/2010/08/update-on-freebsd-jail-based.html*
[13] *http://freebsdfoundation.blogspot.com/2011/03/summary-of-five-new-tcp-congestion.html*
[14] *http://freebsdfoundation.blogspot.com/2010/06/bsnmp-improvements-project.html*
[15] *http://freebsdfoundation.blogspot.com/2011/02/freebsd-foundation-announces-new.html*
[16] *http://news.netcraft.com/archives/2011/03/01/most-reliable-hosting-company-sites-in-february-2011.html*
[17] *http://www.archive.org/details/BsdCertificationGroupACaseStudyInOpenSourceCertification*
[18] *http://wiki.pcbsd.org/index.php/PC-BSD_Users_Handbook*
[19] *http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/eresources.html#ERESOURCES-MAIL*
[20] *http://www.freebsd.org/doc/en_US.ISO8859-1/books/faq/support.html#IRC*

- choose a license with a strong copyleft clause (e.g. GPL) if you are looking to corner a market. This strategy is most successful if you already have a strong brand and other non-open source products/solutions available to generate revenue. If the open version of the software becomes successful, it can kill off the competition (after all, it is hard to compete with *free*).

Individual developers often don't fall into either scenario, meaning that their license choice is driven by the community they most often associate with or by the license that most closely matches their own feelings on the definition of *freedom*.

How to become involved into FreeBSD world?

If you want to become involved as a developer, join the mailing list(s) [19] and IRC channel(s) [20] for the area(s) that interest you. That way you get an idea of who is involved with what, who may be helpful to you, and get an idea of what is being worked on and what is needed. You'll also want to become familiar with using the GNATS database and how to submit a PR (*http://www.freebsd.org/support/bugreports.html*).

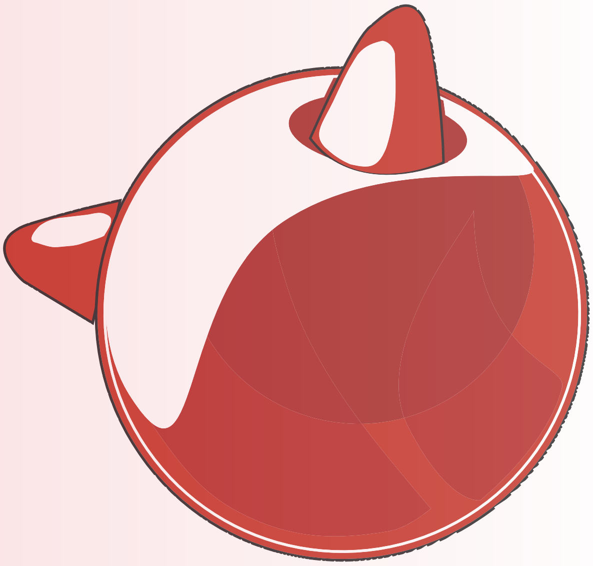
If you are interested in contributing documentation, details about the Documentation Project and how to join are at *http://www.freebsd.org/docproj/*. Information on translation teams are at *http://www.freebsd.org/docproj/translations.html*.

Meeting regularly with other FreeBSD users can be very helpful. See if there is a user group in your area (*http://www.freebsd.org/usergroups.html*). If not, find a local Linux or Unix user group as they often include BSD users.

If you blog about your FreeBSD experience, create FreeBSD related videos or podcasts, have written how-tos, or are organizing a FreeBSD booth at an upcoming conference. let me know and I'll make sure it gets announced at *@bsdevents*. You can reach me at *dru@freebsd.org*.

ANTON BORISOV

Anton is a tech geek, and likes different technological ideas, concepts and devices especially with labels „Powered by: „. But what attracts him most are people, who create these „Power" devices and software. That's why he tries to bring the vision of developers, designers and architects to a broader community.



Why You Use FreeBSD Just May Start With A 'Z'

You may have been using FreeBSD for a long time. You may have just started using it. Regardless of how long you've been using it, whether it's been fifteen years or fifteen days, you have needs, and FreeBSD fulfills some or all of them.

FreeBSD is good at a few things. Networking, firewalling, stability, security just to name a few. The BSD license can be attractive to people who are interested in that sort of thing. The development model is attractive to people who are interested in that sort of thing.

One of the areas that FreeBSD hasn't traditionally been a leader in is storage and filesystems. Sure UFS and Softupdates is a great piece of technology, and countless FreeBSD boxes have booted off it for years. You may be familiar with the guilt of having that 40TB NFS share off your NetApp though. There's a lot of things a big storage box can do that a unix server can't, and some of those things justify having storage devices around, but in some cases you don't need (or even want) all of the features of a NetApp or other storage device, but you'd be able to use a few of those features if you had them.

It's been a few years in coming, but FreeBSD has a solution for you in the form of ZFS, and thanks to a lot of early adopters, people willing to fight through the pain, and countless hours put in by developers, you can enjoy the benefits of ZFS on FreeBSD.

ZFS brings a lot of things to the table, and the benefits that you may care about are a subset of what is available. I'll briefly mention a few of the major highlights.

No need for fsck

As storage volumes get bigger and bigger the fsck runs used by UFS2 have gotten longer and longer. An accidental power cycle of that box with a 6TB array can lead to unacceptable downtime with UFS2. A ZFS filesystem is available immediately after reboot.

Snapshots are fast and cheap

If you haven't fallen in love with snapshots it might be because you're used to them being slow and painful on UFS. Once you get used to being able to use them you'll find they are incredibly useful. Whether it's rolling back a patchset you didn't care for in `/usr/src`, deciding you really don't like today's ports tree and want your previous one back, or recovering that file you deleted in your homedir a few hours ago, the usecases for cheap snapshots and easy rollback are nearly endless. The storage industry considers this a must have feature, and for good reason.

Thin provisioning

Once you get past all the marketing hype, buzzwords, and lingo you'll find that ZFS is capable of thin provisioning using it's ability to set quotas on datasets. I won't attempt to do this feature justice, google can tell you all about the wonderful assets of being able to thin provision, but in a shared resource environment being able to over-commit resources is a great way to maximize usage of those resources.

Data integrity

RAID controllers have used checksums on blocks to detect bit-rot for years. Now you can have a filesystem that has the same functionality. ZFS checksums it's data and can provide device redundancy in pools, or keep multiple copies of your data.... or both. If you have a configuration where ZFS can't repair your data, it can at least notify you that there are problems, as opposed to the silent corruption you can face with other filesystems.

Integrated volume and filesystem management

ZFS does filesystems. ZFS does RAID. One set of tools, one paradigm. No need to learn how to manage geom_mirror and UFS2. In ZFS with the combined approach to volume management and filesystems everything is under one roof so to speak.

Remote replication

ZFS send and receive make it easy to replicate filesystems between systems. By sending only the diffs since the last sync the amount of data on the wire is reduced. While ZFS version 28 isn't in a production release of FreeBSD yet, it's making it's way there, and that brings deduplication to the table, for an even better remote replication experience. The storage industry is making a big deal about deduplication. If you have an onsite and offsite backup server you'll think it's a big deal too.

Capacity expansion

ZFS expands to fit the size of the underlying device, whether you are using a RAID controller that can do volume expansion, adding to an existing pool, or upping the size of the virtual disks in your VMWare instance, ZFS simply expands to fit the underlying storage device. No need to take filesystems offline and mess with partitioning, labelling, or growfs.

Installing FreeBSD to a system where ZFS will be used for the OS isn't as straightforward as installing to a UFS system. There are a few guides available online for CLI types. I use and recommend the guide at *http://wiki.freebsd.org/RootOnZFS/GPTZFSBoot*.

The PC-BSD installer will install a vanilla FreeBSD system to ZFS as well. Hopefully soon the FreeBSD installer will be able to handle ZFS as well. That being said, there are a lot of situations where a system booting off UFS makes sense, and the large storage is on ZFS, in which case installer issues are a moot point.

JOSH PAETZEL

A 37 year old advocate, user and developer of BSD UNIX based systems. he resides in Minneapolis, Minnesota, USA where he hacks on FreeBSD and PC-BSD, both as a volunteer and as part of his full time work as the Director of IT at iXsystems.

OpenBSD improves upon /etc/rc.d/

The OpenBSD developers did not adopt a change like this until they were sure they had a mechanism that was both simple to implement and simple to use.

What you will learn…

- Basic concepts of OpenBSD's rc.d scripts mechanism
- How to start, stop and control daemons that use it
- How to create your own rc.d scripts if you need to

What you should know…

- How to install packages using pkg_add
- How to edit configuration files like rc.conf.local

The init process is the beginning of all userland activity on UNIX and most *nix-like systems. In historical *V7* Bell Labs Research Unix, init ran a single shell script file called `/etc/rc` for starting *standard* daemons. Berkeley added another called `/etc/rc.local` for *locally added* daemons, to simplify maintenance of many machines on a local area network as well as to avoid the chance of the administrator accidentally messing up the script and thus failing to start the important daemons. This mechanism worked well during the epoch of mentored sysadmins, but started to break down as Unix spread beyond the walls of academia/techiedom; people found it hard and/or tedious to edit this file to add or remove local daemons.

Remember that in those days most software was distributed in source code form, to be saved from your newsreader, compiled, and installed. The rise of *packaged* software – both commercial and open source, like Solaris pkg* and later Linux RPMs – further jeopardized the *one file fits all* represented by `/etc/rc.local`. Thus System V, the commercially-developed version of UNIX from AT&T *Unix System Labs*, came up with a scheme of directories to contain locally added startup scripts. `/etc/rc.d/` was born.

In the early days of `rc.d`, there was a single directory, and files would be installed with names like S67lpr, meaning it was to *start* [S] the lpr services, and it would run in alphabetical order against all other S* scripts (scripts with K for Kill were for stopping daemons).

The developer of the package was thus in charge of picking a number that wasn't already used, and a name that didn't conflict. Since ISV developers made their own packages, if two of them happened upon exactly the same name and number, installation would end badly. This rarely happened, but it was a possibility. With System V *run levels*, this scheme became even more complex...

This practice, with variations, was adopted by Linux and picked up but somewhat simplified by FreeBSD.

Linux has the full-blown System V version, with *run levels*, files in `/etc/init.d`, symlinked with S- and K- filenames into `/etc/rc0.d`, `rc1.d`, `rc2.d`, and so on (see the full set of files from Ubuntu 10 here: *http://wwww.darwinsys.com/openbsd/ubuntu10.txt*) FreeBSD 8, for example, has about 150 files in its `/etc/rc.d` for starting the standard daemons, and a program (originally from NetBSD) for sorting them into the correct order based on text tags in each file. This seems complex, but if it works for them, fine. OpenBSD currently starts its standard daemons from a single script, since their order very rarely changes.

But OpenBSD did not go there. OpenBSD up until 4.8 still used the traditional single `/etc/rc.local` script alone.

In typical style, the OpenBSD developers did not adopt a change like this until they were sure they had a mechanism that was both simple to implement, and simple for *normal users* to use. In mid- to late 2010, Robert Nagy (*robert@*) and Antoine Jacoutot (*ajacoutot@*), with help from Ingo Schwarze (*schwarze@*) hammered out the first version of rc.d for OpenBSD, which continues to evolve.

Antoine adds that when he *proposed the idea of using an* `rc_scripts` *variable in* `rc.conf.local(5)` *to control which and* *in what order daemons are started, I originally took the idea from Arch Linux rc system. They use a DAEMONS field in their* `rc.conf` *which our* `rc_scripts` *variable is modeled from.*

OpenBSD's `/etc/rc.d` script mechanism looks similar to that on other systems, but simpler, and with some crucial differences.

A key difference is that creation of a script in `/etc/rc.d` does not automatically start anything. There can be many scripts, but only the ones listed in the `rc_scripts` variable in `/etc/rc.conf.local` will actually get started. While it might seem distressing to those favoring the all-automated approach, it is the right choice for OpenBSD, for several reasons.

First, we work on the principle of least surprise. Installing a package should not automatically make it start when I later reboot – I might just be looking at it, or modifying it, or building it for use on another, remote system.

Another is that our packaging mechanism works a bit differently, again because it is designed to benefit end users. When you do *make build* in a ports directory, any dependant ports will be installed (and built if necessary). Again for least surprise, building one port should not make another start running when you reboot! The extreme case of this is the ports builders – people and their machines who build every port, to make packages available for *the rest of us* to install. Imagine the thrashing if you rebooted such a machine and found 768 new servers starting up!

However, while starting new servers is not automatic, it is simple. For any rc.d scripts that you want to start at reboot, you need only add them to the `rc_scripts` variable in `/etc/rc.conf.local`. On one of my systems, for example, I have

```
rc_scripts="dbus_daemon postgresql freshclam clamd cupsd aox mrtg"
```

(aox is *ArchiveOpteryx*, a cleverly named and designed mail archiver/backend server). On another machine I use

```
rc_scripts="freshclam clamd postgresql"
rc_scripts="$rc_scripts mailman milter_regex spamassassin
                    smtp_vilter mrtg nagios"
```

This not only lists exactly the servers I want to run, but allows me to specify the order among them. People running web sites are presumed to be smart enough to start the database before running web applications that may use them (where this is not the case, a little trial and error may be needed). The current `/etc/rc.local` runs all the scripts named in `rc_scripts`, before running any other shell commands the user may have added. But wait, there's more! If I just this moment added package FOO and don't want to reboot to start it, I only need to type

```
/etc/rc.d/FOO start
```

No worrying about which of several directories to look in, or whether there's an S42 at the front of its name... Similarly, the expected results obtain from

```
/etc/rc.d/FOO reload
/etc/rc.d/FOO restart
/etc/rc.d/FOO check
/etc/rc.d/FOO stop
```

The difference between reload and restart is that reload checks to make sure the daemon is running, and signals it (usually by *kill -HUP*, though the port maintainer can override how this works) – in other words it reloads the daemon – whereas restart does a *stop* followed by a *start*.

The check action is silent but sets the exit status, success (non-zero) meaning that the daemon is running.

The last one – *stop* is also invoked automatically on a system reboot or shutdown, from `/etc/rc.shutdown`, which reverses the order of scripts named in `rc_scripts` before running them all with the *stop* argument, so they shut down in the reverse order to that in which they were started. Again, this happens automatically, and does the right thing.

Ingo adds that *in contrast to some other systems, it is an important design goal to limit the number of actions, so you need not remember the differences between try-restart and force-reload and whatnot, and porters need not implement all that code, and sysops need not read it when looking for configuration bugs. For example, a few days ago, when a mail server running Debian stopped working after an exim update, I looked at the* `init.d` *script script that comes with exim on the Debian system. It was 280 lines long, full of variables read from various places all over the system, and contained complicated code to automatically generate files in* `/etc/exim` *from various template files...*

I decided that I would need at least an hour to understand how to fix that, if not more, so I just deleted `/etc/init.d/exim4` *and rewrote it from scratch. A quarter of an hour later, it was 30 lines long and just worked. With Robert's rc.d framework on OpenBSD, it would be more like five than ten lines and obvious at a glance, so I would even save the 15 minutes to get it working.*

As he says, the scripts themselves are short and easy to write. Here is a hypothetical but otherwise typical-sized rc.d script for the hypothetical FOO daemon, as it might be installed:

```
daemon="/usr/local/sbin/FOOd"
. /etc/rc.d/rc.subr
rc_cmd $1
```

Here is a more complicated one for a daemon that changes its ps name, as seen by the ports maintainer:

```
daemon=”${TRUEPREFIX}/bin/FOOd”
. /etc/rc.d/rc.subr
pexp=”${TRUEPREFIX}/bin/FOO-binary”
rc_reload=NO
rc_start() {
   (sleep 5; ${rcexec} „${daemon} ${daemon_flags}” >/dev/null)&
}
rc_cmd $1
```

Note that TRUEPREFIX will be substituted automatically when the script is installed. As you can see, the typical script only needs to define the *daemon* string (used to start the daemon), source `/etc/rc.d/rc.subr` (which contains all the *complexity* but is still very readable at just over 100 lines), optionally define a *ps expression* or *pexp* to use in pkill, and optionally write one of several overriding functions. The pexp is only needed if the running program (as seen in ps) is other than

```
${daemon} ${daemon_flags}
```

There are several possible `rc _ *` shell functions. In this example we need for some reason to sleep a bit before starting the server. If not for this, the entire `rc_start` function would be omitted. The last line, `rc_cmd` $1, is the *dispatcher* line, and must always be present exactly as shown. Up until now, this mechanism has been used only for ports. As of this writing (March, 2011) about 90 of the most-used daemon-running ports have been converted to provide an rc startup script instead of requesting users to modify `/etc/rc.local`.

Needless to say the old way still works for hand-made servers and for packages that haven’t yet been converted. If you do any manual editing of rc.local and friends, just be careful not to disrupt the part that loops over the `rc_ scripts` variable. If you are a ports maintainer, there’s enough information here to get you started, but you will want to read the man page `rc.subr(8)`. If your ports tree is up-to-date, there is even a template file in `/usr/ports/infrastructure/templates/rc.template`; just copy and hack. All you have to do is create your rc.d script in `pkg/FOO.rc`, and add an *@rcscript* entry for it in PLIST (manually at present; *make update-plist* should soon automate this).

The packaging system will then install it automatically under the name FOO in `/etc/rc.d`. Note that unlike on some other OSes, the ports mechanism will detect conflicts, if two ports both do try to install a script with the same name.

This mechanism is not used for daemons that are in the base OS. Whether it will someday is unclear. The `/etc/rc` (and `rc.conf.local` variables setting) mechanism is already in place and working well to ensure that all the ones a given administrator wants to run do so, and in the correct order.

References

man rc.subr(8) describes the internals of the rc scripts; man rc.d(8) describes the usage of the rc.d mechanism, and man rc.conf(8) describes the general rc.conf and rc.conf.local mechanism. This work was primarily done by OpenBSD committers Robert Nagy and Antoine Jacoutot. Antoine and Ingo offered helpful comments on this article. Ian’s list of *http://www.darwinsys.com/openbsd/ubuntu10.txt* Ubuntu 10 rc.d files and *http://www.darwinsys.com/openbsd/freebsd8.txt* FreeBSD 8 rc.d files

So here is the simplified usage rule for the new mechanism. When you install a package (or compile and install it yourself from ports using *make install*) that has a daemon to be started upon reboot, then if the port has an `rc.d` script it will tell you the name of the script when you install the port. Here’s an example: The following new rcscripts were installed: `/etc/rc.d/tomcat` See `rc.d(8)` for details.

If you miss that, you can just do an `ls /etc/rc.d` to find the exact name of the script; Then you can either do

```
/etc/rc.d/FOO start
```

(where FOO is the name of the script) to start it manually now (unless you want to reboot), or, to always start it on reboot, add the name (e.g., FOO) to the list of scripts in the `rc _ scripts` variable in `/etc/rc.conf.local`.

That’s it – you’re done. This mechanism is not universal, does not solve every imaginable problem, but is widely used in OpenBSD 4.9 – especially -current – and does solve most of the common problems faced both by ports maintainers (who needed a standard way of informing people how to cause the daemons to be started) and package users (who needed a simple way of causing daemons to start). We think we have achieved the right balance of simplicity and functionality. So if you have an older OpenBSD, try upgrading it (Buy a 4.9 CD from the OpenBSD web store, or download -current, in either case following the upgrade instructions in the FAQ), and rip out most or all of those complex startup commands from `/etc/rc.local` and replace them with one or two lines of `rc_script` definitions.

I think you’ll like it!

IAN DARWIN

Ian Darwin is an OpenBSD committer who lives in the country well north of Toronto, Canada. He runs *NIX on just about all his computers; he once said that his only Windows looked out over the hillsides where he lives.

DragonFly News

There's been some dramatic changes for DragonFly in the past month; all positive but having significant effects.

GCC upgrade

On March 4th, Sascha Wildner changed the default system compiler to GCC 4.4. DragonFly is the first BSD to move to this release, as far as I know. The previous system compileri, GCC 4.1, is still in the system and available. To avoid building it during a buildworld, use *make NO_GCC41=1 buildworld* and *make NO_GCC41=1 installworld*.

So far, the actual fallout has been minimal. The kernel and world build normally. The place for problems, if there is to be problems, is in pkgsrc. DragonFly uses pkgsrc for handling third-party software, and pkgsrc has not been used with GCC 4.4, historically. It's dealt with other compilers on the various platforms pkgsrc runs on, so this is not out of the ordinary.

I can report that in my bulk builds of pkgsrc using DragonFly and GCC 4.4, I see very little problems. most of the software in pkgsrc also runs on Linux, so moving to GCC 4.4 may be *fixed* by most, if not all of the packages becoming compatible upstream, with the original developers.

The work is continuing; binutils 2.21 was imported but not yet part of the build as of March 24th.

Summer of Code

Along with FreeBSD and NetBSD, DragonFly is participating again in the Google Summer of Code event. The BSDs have been very successful at putting together Summer of Code projects in the past few years, and this year should continue the trend.

DragonFly has 8 mentors lined up for this year's season, so there could be a large number of projects this year. Discussion of potential work happens on the *kernel@dragonflybsd.org* mailing list, so check there for more details. Depending on when you are reading this issue, the student application period will either be running or completed.

Details on project ideas for this year are on the DragonFly website, along with pages from previous years. Check *http://www.google-melange.com/* for details on the 174 other organizations involved in this year's work.

Interrupt Code

Longtime DragonFly developer Sepherosa Ziehau delivered a complex and useful project for DragonFly this month: modern interrupt routing and APIC support. The interrupt routing uses the very new 20110211 version of ACPI.

What does this mess of acronyms mean? Some modern machines would not run DragonFly, or would only work with a uniprocessor kernel. This update means that more machines will boot DragonFly, and/or run with a SMP kernel. This update is a huge task, and very helpful for machine support.

Miscellaneous

There's been some interesting benchmark data produced recently, with a recent run of the fefe.de scalability tests and also some Ruby microbenchmarks. DragonFly appears to do well on those, though the usual caveats about benchmarks apply. Visit the DragonFly Digest for more information and links.

Euraeka.com is a new news search site running on DragonFly. A new DragonFly BSD group on identi.ca has been created.

Upcoming work

What about LLVM/Clang as a compiler?, you may ask. (or even *What about pcc*?) Those have been used on DragonFly, and clang has been available via pkgsrc for some time. DragonFly has historically always had two compilers in the base system – one by default, and one either as a fallback version, if it's older, or as an experiment, when it's a newer version. It has always been versions of GCC, but that is more by circumstance than design.

If this is your particular bug to scratch, try installing clang from pkgsrc and compiling. Building world and perhaps even the kernel is possible using clang at this point. The project could use a comparison in file size and speed when compiling the same code with different compilers.

Samuel Greear has been working on code for Samba to translate Hammer snapshots into Windows-style shadow copies, so that a Windows system mounting a Samba share on a DragonFly system can access those snapshots *natively*. Some example code has been produced; this may turn into a Summer of Code project if someone is willing.

Convention season is starting, with AsiaBSDCon proceeding despite the recent earthquake, and BSDCan coming soon. There's no planned DragonFly presence at the shows, but there may be developers showing up.

JUSTIN C. SHERRILL

Package Management for the upcoming PC-BSD 9

Among the various improvements planned for PC-BSD 9.0, among the largest of these is the refreshed PBI package management format.

The updated PBI format offers many new features such as digital signing, binary diff patching, and repository management, while preserving the original goals of format, by allowing packages to be installed and run in a self-contained manner. Lets first take a look at how the PBI format differs from traditional package management, and then explore the changes in the upcoming 9.0 release.

The biggest difference between applications packaged in the PBI format vs traditional FreeBSD packages or RPMs is that each archive contains a complete set of libraries and dependent data required for for the target software to function.

This means in effect, that a PBI is self-contained, and doesn't require changing system packages to load the appropriate dependencies. A PBI file can be installed or removed at will, without effecting other applications or running the risk of causing breakage elsewhere in the process. With the re-implementation of the format for the upcoming PC_BSD 9, this core concept has remained and been expanded upon.

One of the first major changes for the next PBI format has been the addition of library and file sharing between PBIs. In the previous implementation of the PBI format it was common that identical files existed between various applications. These duplicates, while necessary to provide self-contained functionality, still wasted both disk and runtime memory space. In the new format this waste of space has been greatly reduced through the usage of what has been dubbed the *hash-dir*.

In this directory libraries and common files are able to be shared between various PBIs through a system of hard-links. When an identical file is found in a PBI, the original will be removed and a hard-link created to the copy already in the *hash-dir*. After a PBI has been removed, any unneeded files left in the *hash-dir* are cleaned up by the *pbid* daemon, which monitors and maintains the integrity of the shared files.

Another new feature is the ability for administrators and PBI builders to create and manage their own *repositories* of PBIs. This repository system provides a number of tools for PBI distribution, release management and more. For the end user tools are now provided which allow the browsing of PBIs in a repository, enabling auto-updates and mirror configuration.

Tied into the new repository system is a new feature to digitally *sign* PBI files, and verify their contents. When a distributor creates a new repository, it includes an openssl public key file which is installed on the end users system. When the distributor runs a PBI build process, the resulting PBI file includes several signatures for the content archive and installation and removal scripts. During the PBI installation process, these signatures are checked to confirm that the archive has not been tampered with during transit. This key file is also used to associate a particular PBI with a parent repository for upgrade purposes, since it is possible that multiple repositories will have the same applications.

Since the very nature of self-contained packages tends to produce larger installation files, one thing which needed improvement was the updating process. In the previous implementation updating an installed PBI would require the re-downloading of the entire installation archive.

For larger applications this could be a rather time-consuming process, especially for users running over low bandwidth connections. In the typical application version update, only a small percentage of files had actually been changed, and a majority of these may simply be building time-stamps.

In order to solve this problem the new PBI specification now supports updating via binary diff patches. Distributors running PBI builds can now enable options to generate small PBP (Push Button Patch) files, which are often a fraction of the size; in some cases less than 5% of the original PBI archive.

When the end user begins the update of a PBI file from a repository, it automatically checks for for the presence of a PBP file, and attempts to use it, only falling back to the original archive should the process fail.

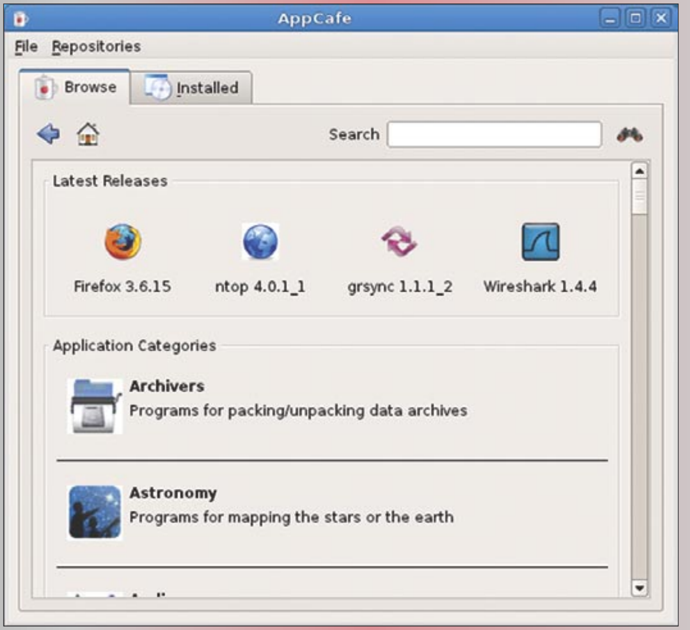


Figure 1. *Sneak peak at the upcoming front-end to the PBI 9.0 Format – The AppCafe*

A more recent addition to the new PBI format is the ability for applications to be installed by user (non-root) accounts. Since each application is entirely self-contained, and doesn't require changing around other installed packages, it became very easy to implement functionality for user-installation. By default the PBI format allows users apart of the systems *operator* group to perform installations and upgrades. This allows enhanced security in office or home situations, where users can now add/remove software on their desktop without being able to use the *root* account.

All these new features in the PBI format for PC-BSD 9.0 have already made it far superior to the existing legacy format. For traditional FreeBSD users though, perhaps the most important new feature is the implementation. In the previous PBI incarnation it was developed in QT/KDE C++, which made running on native FreeBSD cumbersome, especially on a system which had no need for X11 installed. The new format is implemented 100% in shell, and is able to run on traditional FreeBSD entirely from the command-line. The various functionality is broken up into approximately 15+ command-line utilities with man-pages for each, which makes native FreeBSD usage a natural fit.

We've just taken a brief look at this reimplemented PBI format, and some of the new features it offers. This format will ship as the default for PC-BSD 9.0 and beyond and is currently available for beta testing in our PC-BSD 9-CURRENT snapshots. Once out of beta, it will be available to install on traditional FreeBSD via the ports system.

For further information:

- PBI Manager Wiki: *http://wiki.pcbsd.org/index.php/PBI_Manager*
- PC-BSD 9-CURRENT Snapshots: *ftp://ftp.pcbsd.org/pub/snapshots/*
- PBI Developer Discussion: *http://lists.pcbsd.org/mailman/listinfo/pbi-dev*

KRIS MOORE

Kris Moore is the founder and lead developer of PC-BSD. He lives with his wife and four children in East Tennessee (USA), and enjoys building custom PC's and gaming in his (limited) spare time. kris@pcbsd.org

Converting a Physical Partition

with FreeBSD to a vmware Image

Portability is something people increasingly value, because it has a number of advantages – you can, for example, carry your desktop (or server) anywhere with you and thus also all your very important personal data that you have created over some time, or perhaps over many years.

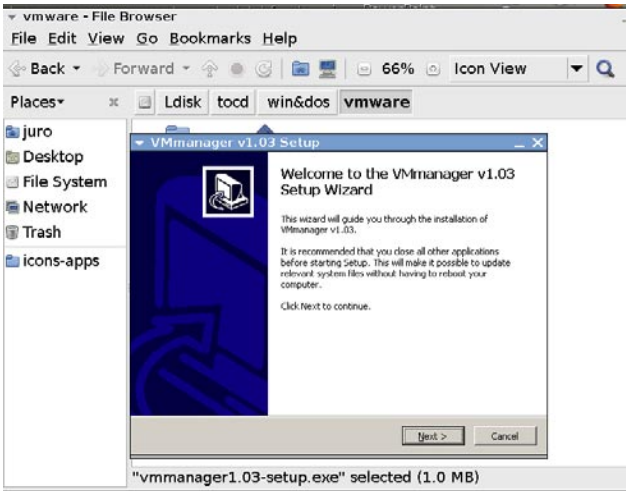
What you will learn…

- How to transfer a physical partition with FreeBSD to a VMware image
- How to use the image across a number of platforms

What you should know…

- Basic knowledge on virtualization

There are many virtualization solutions today, but I chose a VMware solution for purposes of this article (conversion of a physical partition), as VMware is a pioneer in virtualization industry, company' products are very fast, extremely portable, and many of them are free. VMware images are playable in VMware Player, which is also free for platforms that support it. This means that you can carry VMware Player and your images anywhere on a USB disk and play them without restrictions in Linux and Windows.



Welcome to the VMmanager v1.03 Setup Wizard

This wizard will guide you through the installation of VMmanager v1.03.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

"vmmanager1.03-setup.exe" selected (1.0 MB)

Figure 1. *VMmanager should run under Wine in Unix (you see installation in GNOME)*

Putting a VMware image back to a physical disk is quite simple, too, but I will mostly focus on how to make a portable FreeBSD (applicable also to OpenBSD) virtual computer. I decided for this solution because my motherboard recently died. I realized that with a portable virtual computer on a portable USB disk I can enjoy real freedom.

Although I have a notebook, my Internet browser in it is without e-mail accounts and without all my bookmarks, music and books. When my motherboard died I understood that I must have a couple of virtual images, but not just the ordinary ones like I can download from the Internet, but the ones with all the software I love and with all my personal customizations – something an image of a virtual PC downloaded from the Internet can never substitute.

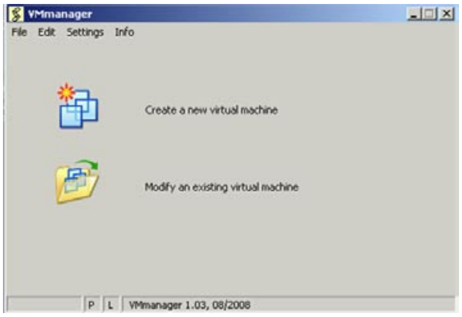

VMmanager
File Edit Settings Info
Create a new virtual machine
Modify an existing virtual machine
P
L
VMmanager 1.03, 08/2008

Figure 2. *With VMmanager you can create a new VMware image, or edit the existing configuration of virtual disks*

What you must do to convert a physical disk to a VMware image?

I will carefully describe all the steps I did, but as we are moving into the waters of portability, I must, too, leave the BSD ocean and move into Linux or Windows waters.

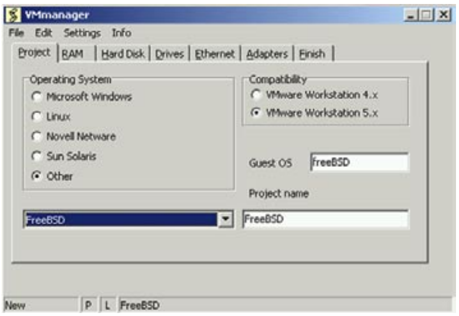
You must prepare an empty and configured (with CD-ROM devices, disks, networking, etc.) VMware image (the configuration is actually held in an accompanying VMX text file). To do this,

- either use proprietary VMware software; there are several packages, VMware Server (*https://www.vmware.com/products/server/faqs.html*) is free (Linux and Windows);
- or download one of many VMware images (*http://www.vmware.com/appliances/directory/cat/508* free premade appliances), delete its content and use it; however, one of the drawbacks of this solution is that the premade images for VMware Player do not always come in the sizes you may like.
- You may use some free utilities like qemu-img, which is included in the Qemu package and which creates empty disk images for use not only with QEMU, but also with VMware products. Vmmanager (*http://www.softpedia.com/get/System/OS-Enhancements/VMmanager.shtml*) is another free tool that does the same job but in Windows (see Figure 1-3).

Before you make the backup, empty spaces on your physical partition must be zeroed, otherwise the gzipped partition will be too big. FreeBSD has internal partitions, so run the following command in all of them:

```
dd if=/dev/zero of=/usr/000file bs=200M
```

Note please that I use the term *partition* here as it is used generally for Windows or Linux. However, in the FreeBSD terminology I should use *slice* (*s* in device such as ad0s2). In FreeBSD, partitions (internal partitions that fdisk of Windows or Linux does not see) have letters assigned to them (*ad0s2a*, *ad0s2d*, etc. – that is, here we deal with slice 2 and internal partitions *a* and *d*).


VMmanager
File Edit Settings Info
Project | RAM | Hard Disk | Drives | Ethernet | Adapters | Finish

Operating System
○ Microsoft Windows
○ Linux
○ Novell Netware
○ Sun Solaris
● Other

Compatibility
○ VMware Workstation 4.x
● VMware Workstation 5.x

Guest OS: FreeBSD

Project name: FreeBSD

FreeBSD

New | P | L | FreeBSD

Figure 3. *Your VMware images (files with VMDK extension) are accompanied with a VMX file, which contains all the configuration (RAM, devices, etc.) and which also describes the virtual disk*

After a very big file is created in `/usr` with the above command, delete it and run the command again, but point the execution of zeroing empty spaces into `/var` (`of = /var/000file`), `/tmp`, etc., because FreeBSD (during its installation process) always assigns (and mounts) its own internal partitions to these directories (`/ = ad0s1a`, `/var = ad0s1e`, `/tmp = ad0s1d`, etc.). To avoid possible boot problems, make sure that you copy (backup) the physical partition (if unsure, look into `/etc/fstab`) from `/dev/ad0s2` to `/dev/ad0s2` (not to `/dev/ad0s1` in your virtual image). If the FreeBSD partition on your physical hard drive is `/dev/ad0s2`, make two primary partitions in your virtual image and restore the backup to `/dev/ad0s2` (not to `/dev/ad0s1`). However, FreeBSD boots more intelligently than Linux. In case you have a different disk specified in `/etc/fstab` in your physical hard drive, the system (shortly after it boots) will ask you to supply the correct boot device. I inform about this because I took care of this issue and I advise you to do the same.

Now you must backup your FreeBSD partition. You may use any image creation program (like Norton Ghost, Acronis, etc.) that makes an image of a physical disk, or just use the following Unix command:

```
dd if=/dev/ad0s2 bs=1M | gzip -2 > /mnt/diskimg.gz
```

(and replace `/dev/ad0s2` with your own device).

The VMware image must be partitioned and you must use the primary partition for FreeBSD (not a logical one). The size of it cannot be smaller than the size of the uncompressed image (the copy of your physical disk) now residing in the gzipped file (or in the Norton Ghost's GHO image, etc.). To partition your virtual disk, you can also use the standard FreeBSD `sysinstall` tool (available on a bootable installation FreeBSD CD/DVD), or utilities such as gparted (*http://gparted.sourceforge.net/*).

In my case I used an old DOS version of Partition Magic (I created three unformatted primary [DOS] partitions in my virtual disk) – I made two small primary partitions and the one required for FreeBSD (`/dev/ad0s3`).

You must create a bootable ISO image with the compressed (gzipped, etc.) partition in it; after you boot your virtual computer with it, restore the copied physical partition to your newly created VMware image. The way I did this was that I prepared a bootable ISO of the Norton Ghost image (compressed physical partition) with a bootable DOS floppy image in it (run the command like: `mkisofs -b DOSfloppy.img -o /mnt/DOSandIMAGE.iso`. With the dot at the end of the command mkisofs will put the directory where you presently work into the ISO file, so keep also your partition in the GHO or gzipped format in this directory). If you want to do it the same way as I did it, visit *http://www.bootdisk.com* and download some DOS/Win98 floppy images.

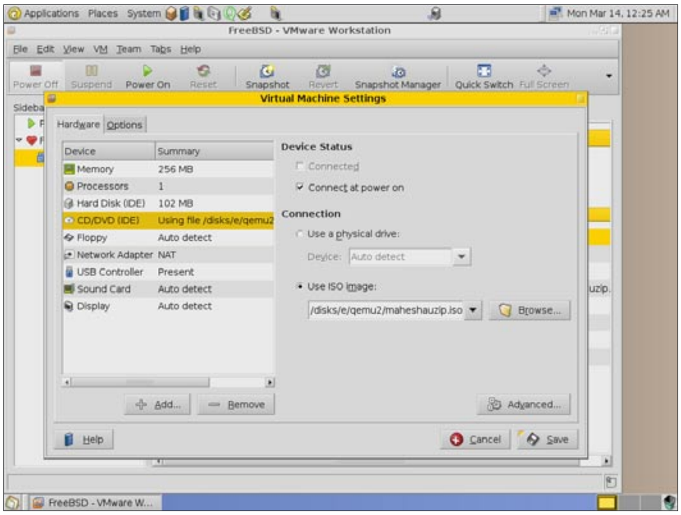
The dd command to restore the image can be used, too, but you need some Unix bootable floppy or CD/ISO. If you feel a little bit uncertain, use the NetBSD tool g4u (*http://www.feyrer.de/g4u/*), which is Harddisk Image Cloning for PC's and which works the same way as Norton Ghost.

You may also work with two ISO images; however, one of them must be bootable (ISO image of g4u, for example; the other ISO image containing your copied physical partition does not need to be bootable in case you use two CD-ROM's – that is, two ISO files; see Figure 4).

After you create the partition(s) with fdisk (or gparted, Partition Magic, sysinstall, etc.) in your VMware image and with the size corresponding with the size of your physical FreeBSD disk, reboot the virtual computer. With a DOS or Unix bootable floppy or bootable CD (ISO image) you may now restore the backup. To do this with dd in Unix, run the command:

```
gzip -dc /path/to/image.gz | dd of=/dev/ad0s1
```

In case you decide to go on with the Acronis (*http://www.acronis.com/main.html*) products, you can easily create a bootable CD applicable for image creation and restoration. But in this case the easiest way is to put the Acronis image file (the copy of your FreeBSD physical disk) into another ISO image (with `mkisofs`). With utilities such as g4g or Live CD's such as my MaheshaBSD project (*http://www.freebsd.nfo.sk/maheshaeng.htm*) and with a second computer around it is painless to restore the copy of your physical disk via FTP in Unix. MaheshaBSD has a working VSFTPD server in it, so just boot two computers with this CD and run the dd or the dump command to restore the image. It can even reside on a NTFS partition, as MaheshaBSD has no problem to mount NTFS volumes. To learn how to use networking in your virtual computer, see Some Questions And Answers below.



Device	Summary
Memory	256 MB
Processors	1
Hard Disk (IDE)	102 MB
CD/DVD (IDE)	Using file /disks/e/qemu2
Floppy	Auto detect
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Display	Auto detect

Device Status

☐ Connected

☑ Connect at power on

Connection

○ Use a physical drive:

Device: Auto detect

● Use ISO image:

/disks/e/qemu2/maheshauzip.iso

Figure 4. *You can add another CD-ROM in your VMware Server's virtual machine settings and use two CD-ROMS (two ISO files without need to burn them onto a CD/DVD)*

The last and very important thing is to make sure that the virtual partition that holds your FreeBSD in it is set active. You may also install FreeDOS, or some very small Unix systems (or even full installation of OpenBSD) onto the first two small partitions (if you have them) and use GAG boot manager (*http://gag.sourceforge.net/*) to switch between them (multiple boot). This does not mean that small partitions are really needed, but they may appear very handy for testing purposes.

Some Questions and Answers

How to play your VMware images in FreeBSD?

VirtualBox

In VirtualBox, just select *File>Virtual Media Manager*, click on Add in Virtual Media Manager and select your newly created VMware VMDK file. VirtualBox runs under FreeBSD very well and it is free (see Figure 5).

Qemu

Alternatively, you can use some conversion utilities like qemu-img and convert the VMware virtual disk for use with Qemu. Unfortunately, Qemu is slow, but there are a lot of sites dedicated to VMware-to-Qemu conversion. You need to install Qemu and use its qemu-img utility:

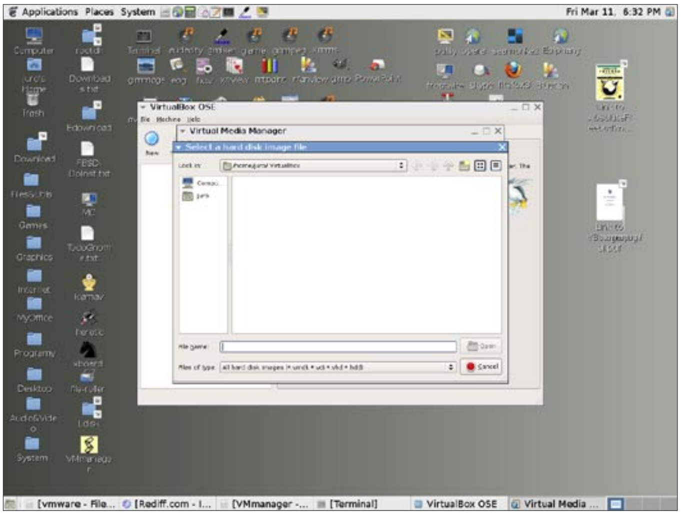


Figure 5. *As you see on the picture (Files of type at the bottom), VirtualBox supports VMware's VMDK virtual disks*

```
qemu-img convert win2kpro.vmdk -O qcow win2kpro.img
```

VMware

Running VMware on FreeBSD as a host is possible, but only older versions of VMware Workstations run. I recommend using VirtualBox.

Is it possible to convert VMware disk image (VMDK) to Xen?

Yes. You will find many websites that deal with this issue. However, there may be some little problems, so check them out. For the conversion you need the vmware-vdiskmanager tool that comes with VMware Server (for example, VMware Server 1.0). Check the *http://wiki.xensource.com/xenwiki/VMDKImage* website.

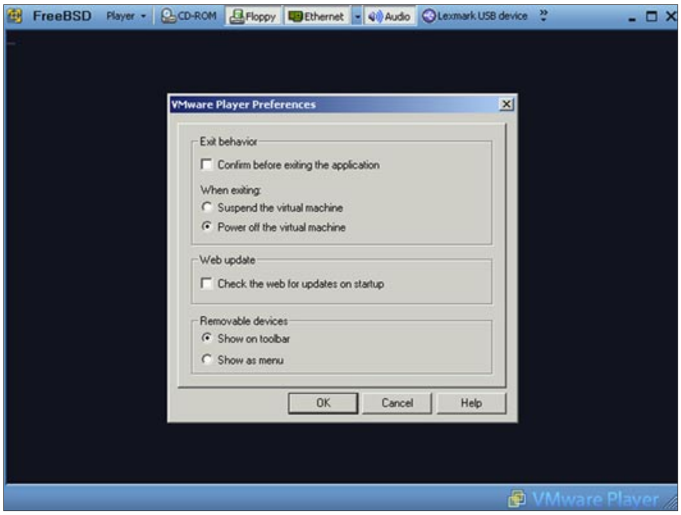
How to play your VMware images in Linux/Windows?

Download VMware Player, VMware Server, or VirtualBox. All alternatives are free and available for several platforms. To use your VMware image with Windows, edit the VMX file in notepad and change the syntax for CD-ROM and floppy devices (if you need them). Change:

```
ide1:0.fileName = „/dev/acd0” to ide1:0.fileName = „H:”
floppy0.fileName = „/dev/fd0” to floppy0.fileName = „A:”
```

Use the real letter (H: or I: or G:) that Windows assigned to the CD-ROM device.

In Linux the VMX file must describe these devices as they are known to the hosting Linux system. If your CD-ROM device is `/dev/sr0` in Linux, you must have the following line in your VMX file: `ide1:0.fileName = „/dev/sr0”`.



VMware Player Preferences

Exit behavior

□ Confirm before exiting the application

When exiting:

○ Suspend the virtual machine

● Power off the virtual machine

Web update

□ Check the web for updates on startup

Removable devices

● Show on toolbar

○ Show as menu

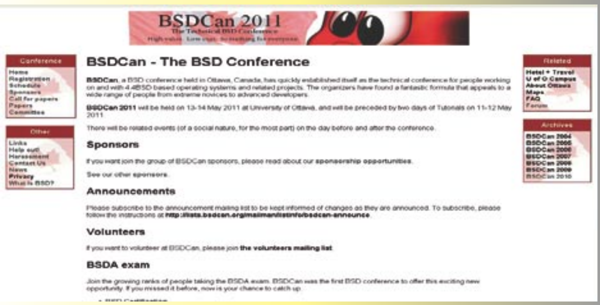
OK Cancel Help

Figure 6. *In the VMware Player's Preferences choose "Suspend the virtual machine"; then close the Player by clicking on its closing x in the upper right corner and the snapshot is made*

Conferences

BSDCan 2011

May 13-14
Ottawa, Canada



BSDCan - The BSD Conference

BSDCan, a BSD conference held in Ottawa, Canada, has quickly established itself as the technical conference for people working on and with 4.4BSD based operating systems and related projects. The organizers have found a fantastic formula that appeals to a wide range of people from extreme novices to advanced developers.

BSDCan 2011 will be held on 13-14 May 2011 at University of Ottawa, and will be preceded by two days of Tutorials on 11-12 May 2011.

There will be related events (of a social nature, for the most part) on the day before and after the conference.

Sponsors

If you want join the group of BSDCan sponsors, please read about our **sponsorship opportunities**.

See our other sponsors.

Announcements

Please subscribe to the announcement mailing list to be kept informed of changes as they are announced. To subscribe, please follow the instructions at **http://lists.bsdcan.org/mailman/listinfo/bsdcan-announce**.

Volunteers

If you want to volunteer at BSDCan, please join **the volunteers mailing list**.

BSDA exam

Join the growing ranks of people taking the BSDA exam. BSDCan was the first BSD conference to offer this exciting new opportunity. If you missed it before, now is your chance to catch up.

- BSD Certification

http://www.bsdcan.org/2011/

Open Source Business Conference

May 16-17
San Francisco, USA

COMPUTERWORLD
OSBC
SAN FRANCISCO
OPEN SOURCE BUSINESS CONFERENCE
Building Your Big Data Future with Open Source
May 16-17, 2011
Hilton San Francisco Union Square
San Francisco, California
Home
Agenda
Sponsors
Registration
Venue
Contacts

Register Today!

OSBC 2011
May 16-17, 2011
Hilton San Francisco Union Square
San Francisco, CA

Building Your Big Data Future With Open Source

- Open source strategies are driving data-driven businesses today
- Enterprise software infrastructure is increasingly supporting the consumerization of IT
- Open source is influencing the economic impact on customers in terms of storage, pricing and query time
- Open source is driving tomorrow's technologies by providing infrastructure for Cloud, Big Data and more
- Open source is creating competitive advantage

Why You Should Attend

The Computerworld Open Source Business Conference (OSBC) is the premier forum for business and technology leaders looking for an insightful discussion of how open source technology is changing the way we do business today. With a rich and deep agenda built around the concept of emerging business models and the best strategies for incorporating the strategies for open source software into growing your business, OSBC makes the argument that every enterprise is, or should be, a data-driven business today. As the IT industry's only forum for discussing how to reap profits from using open source software, OSBC brings together a vibrant group of the industry's top practitioners, venture capitalists, lawyers and thought leaders for two days of in-depth presentations and lively discussions and panels. By being the leading conference for educating top tier executives on the value of the open source market place, OSBC provides the latest in cutting-edge open source thinking. OSBC offers the change to connect with the developers, users and companies behind the most significant open source Big Data technologies, teaching attendees the strategies to making your business more effectively data-driven.

New Highlights

For the 2011 OSBC, the agenda is built around four main areas: Big Data, Big Cloud, Big Legal Problems and Big Money. In today's continuing tough economy, open source has led the way

SPEAKING OPPORTUNITIES

If you are interested in speaking at OSBC 2011, please contact:
Matt Asay
Program Chair

SPONSORSHIP OPPORTUNITIES

Sponsoring OSBC is a great way to reach open source technology decision-makers. For 2011 sponsorship opportunities and information please contact: Mike Dutton, Sponsorship Sales, 801-358-4231.

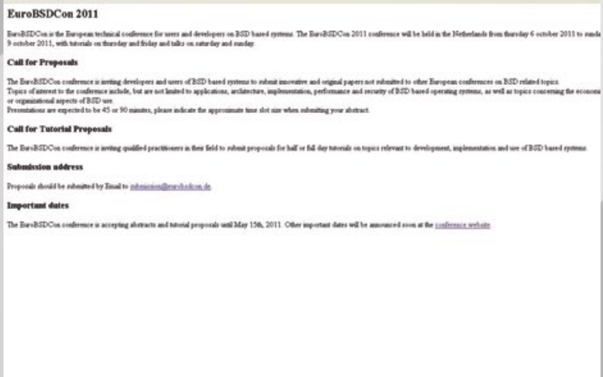
Who Should Attend

- IT managers implementing, managing and integrating open source software
- Developers actively developing and modifying open source software
- Legal professionals overseeing the rights of open source projects

http://www.eiseverywhere.com/

EuroBSDCon 2011

October 6-9
Netherlands



EuroBSDCon 2011

EuroBSDCon is the European technical conference for users and developers on BSD based systems. The EuroBSDCon 2011 conference will be held in the Netherlands from thursday 6 october 2011 to sunda 9 october 2011, with tutorials on thursday and friday and talks on saturday and sunday.

Call for Proposals

The EuroBSDCon conference is inviting developers and users of BSD based systems to submit innovative and original papers not submitted to other European conferences on BSD related topics.
Topics of interest to the conference include, but are not limited to applications, architecture, implementation, performance and security of BSD based operating systems, as well as topics concerning the econom or organizational aspects of BSD use.
Presentations are expected to be 45 or 90 minutes, please indicate the approximate time slot size when submitting your abstract.

Call for Tutorial Proposals

The EuroBSDCon conference is inviting qualified practitioners in their field to submit proposals for half or full day tutorials on topics relevant to development, implementation and use of BSD based systems.

Submission address

Proposals should be submitted by Email to submission@eurobsdcon.de.

Important dates

The EuroBSDCon conference is accepting abstracts and tutorial proposals until May 15th, 2011. Other important dates will be announced soon at the conference website.

http://2011.eurobsdcon.org/CfP.html

How to use the Internet?

The le (or em) driver handles networking. I configured my VMware image with a bridged networking and used the `dhclient` command to connect to the Internet (`dhclient le0`). It worked.

How to run X?

Before you backup your physical FreeBSD partition, install the Xorg VMware driver. It is in `/usr/ports/x11-drivers/xf86-video-vmware`. You can do this even after copying your physical partition to the virtual image. Running the command `Xorg -configure` in your virtual computer is necessary, too. Depending on the version of Xorg, sometimes adding the command `Option "AllowEmptyInput" "Off"` to your newly generated `xorg.conf` is necessary for your mouse to work. Then just run the `startx` command and enjoy.

Are VMware tools necessary?

The company does not presently support use of VMware tools (a special package of VMware enhancements) for FreeBSD and they are actually not needed at all for networking or X.

How to put a virtual computer back to a physical partition?

You must use the same method but reversely (the `dd` command, Acronis, Norton Ghost, g4u, etc.) – back up your virtual computer. To do so, you only need another (empty) virtual disk or just an FTP storage. After making a copy of your virtual disk you must make a bootable CD/DVD with it. Use the same standard way you use for restoring physical partitions.

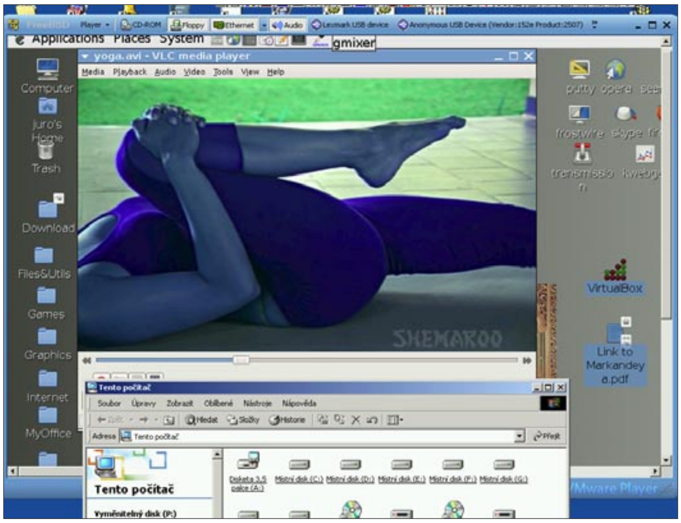


Figure 7. *FreeBSD 8.1 transferred from a physical disk running under Windows 2000 in VMware Player*

Are there any special tweaks recommended?

Reduce `kern.hz` (*Kernel Timer Frequency*), as the FreeBSD's default value of `kern.hz` is set to a relatively high number. High kern.hz value is beneficial on real hardware. To check your present `kern.hz` value on the system, type: `sysctl kern.hz` and if it is over 100, make your virtual FreeBSD a little bit faster by adding the following line to your `/boot/loader.conf: kern.hz=45`.

How about sound and video?

I did not make any changes and the sound and video worked (ogg, mp3 files, avi, mpg, etc.).

May I use snapshots with FreeBSD?

Yes. Snapshots, a very valuable feature of every good virtualization software today, save your time, because you immediately restore your virtual system to the point you left it the last time (starting your virtual machine again but without booting it), but they bring a very big security problem (see Figure 6). Anybody who has a physical access to your virtual computer's snapshot may (after he or she restores it) read your emails (if a snapshot was made with Thunderbird active on your desktop), or perhaps grab your passwords. Use snapshots in physically secure environment only (do not carry them on portable USB disks that you may lose). Although you may argue that anybody can boot a virtual FreeBSD machine in a single user mode, you are right, but this article is not primarily about security. When someone opens your VM in single user mode it is still a bit harder for him or her to access other encrypted stuff that is not left as an easy prey on your desktop (seen immediately after a snapshot restoration). However, the rule of thumb is that the same security measure you apply to your car keys do also apply to your virtual machines and keep them always away from prying eyes (see Figure 7).

JURAJ SIPOS

Juraj lives in Slovakia and he works in a library in an educational institute (with focus on school psychology, disabled children, hipotherapy – movements of a horse as treatment for the disabled, dyslexia, etc.). Some time in the past he was fortunate enough to travel around the world and he spent a bit of time in India and Australia. He worked with the HP 3000 series computers in the 1980's. Juraj's hobbies are computers, mostly Unix, but also spirituality. He has also translated several books from English, for example – Zen Flesh, Zen Bones by Paul Reps. He started with FreeBSD in 1997. He wrote the Xmodmap Howto „http://tldp.org/HOWTO/Intkeyb/" In addition to computers, he is very interested in Hinduism but not really the guru side of things, but more-so freedom and self actualization. His website has more information: http://www.freebsd.nfo.sk/

Looking for help, tip or advice?
Want to share your knowledge with others?
BSD MAGAZINE
Give us your opinion about the magazine's content
and help us create the most useful source for you!

Build appliances with QEMU and OpenBSD

OpenBSD is the slimmest desktop OS. It is complete, functional and usable on any computer as long as your expectations are that of an engineer as opposed to a user.

What you will learn…

- Qemu and building appliances with OpenBSD for networking applications

What you should know…

- Basic OpenBSD knowledge

If you wish to use OpenBSD for a specific purpose then there is no mandate that it has to be run on desktop machines.

With QEMU you can run multiple instances of the OS even on a modest desktop hardware or laptop. I use QEMU much of my product development work since QEMU gives me enormous flexibility. Using Qemu saves me from wasting optical media since I don't have to burn CD/DVD to test my new ISO images. I can boot off an ISO image and install to a hard disk without using optical media.

No real hard disk, no optical medium. How? QEMU allows us to simulate everything. QEMU allows us to dump the entire OS block by block into a flat file and you can boot off that file.

You call that flat file as the OS image. Once you make your changes into the flat file (it is like making changes to a file on a hard disk) and after you are done developing just dd(1) the file to a flash memory on the embedded system you are working on and your product is ready.

I consider this process cooking. Essentially what I do is compress the hard disk image using p7zip, then create a bootable ISO image which anyone can download from the Internet, install the hard disk image to a hard disk,USB stick or DOM(Disk on Memory) module and he gets to use the product.

The product in question could be a firewall, router, layer III switch, link concentrator whatever. It does not matter.

All you do is develop the whole thing with QEMU. Once you are satisfied cook it to make an ISO image.

In this article I shall explain in detail what I really mean by creating products with QEMU. You need to know something about OpenBSD pf, some networking concepts and so on. Let us get started.

Brief intro To QEMU

QEMU is an extremely fast processor emulator that can be used for doing sophisticated things even when run as an ordinary user. Root privileges are not required in most cases. Having a really fast virtualization technology is a great convenience since even on modest hardware it runs remarkably fast. This is what attracts me to it. It also helps me do anything I want without using any physical hardware box.

You can emulate multiple network cards, multiple hard disks and you can even run TCP or UDP servers on the guest if you run it as root using QEMU bridging.

It is quite fantastic how you can develop the entire appliance without wasting time and money on real physical devices.

Anyway in this article we will look at some of the ways in which QEMU has been used by myself in doing my work.

The simplest use of QEMU is when you have an ISO image you download from the Internet. Just typing:

```
$ qemu -cdrom install48.iso
```

would open a new window when you run this from X and inside that window you will get this Figure 1.

Inside the QEMU window, you can type as if you use an application. Now this is not very interesting. We want a way to install the OpenBSD OS into a hard disk. You go about it like this.

```
$ qemu-img create ex.bin 2G
Formatting 'ex.bin', fmt=raw, size=2097152 kB
$ qemu -hda ex.bin -cdrom install48.iso
```

Now you can install OpenBSD into the hard disk ex.bin the usual way. Once you are done, just boot off the hard disk using:

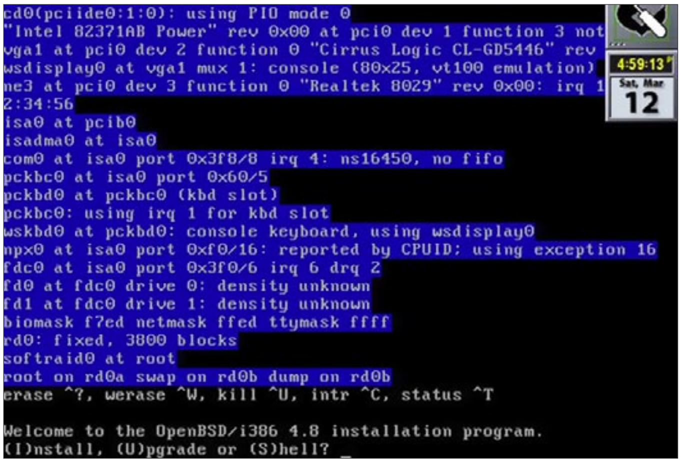
```
$ qemu ex.bin
```

That is it. Simple and straight forward. Now the problem with this is that when you run as a normal user QEMU does not support full networking. You need to have a bridge connection to the host machine in order to run a server inside the QEMU guest.

Please do not get confused with words like host and guest. The machine on which you run QEMU is the host machine and the virtual images (*ex.bin* above) is the guest OS. Normally the host OS provides support for emulation in the form of QEMU which runs under the host and the guest runs as a separate process on the host CPU. But this process includes both kernel and userland. QEMU can also be used with remote machines without X in -nographic mode. For this you have to have this in `boot.conf`.

```
# cat /etc/boot.conf
set tty com0
```

And you can start QEMU with :



```
cd0(pciide0:1:0): using PIO mode 0
"Intel 82371AB Power" rev 0x00 at pci0 dev 1 function 3 not
vga1 at pci0 dev 2 function 0 "Cirrus Logic CL-GD5446" rev
wsdisplay0 at vga1 mux 1: console (80x25, vt100 emulation)
ne3 at pci0 dev 3 function 0 "Realtek 8029" rev 0x00: irq 1
2:34:56
isa0 at pcib0
isadma0 at isa0
com0 at isa0 port 0x3f8/8 irq 4: ns16450, no fifo
pckbc0 at isa0 port 0x60/5
pckbd0 at pckbc0 (kbd slot)
pckbc0: using irq 1 for kbd slot
wskbd0 at pckbd0: console keyboard, using wsdisplay0
npx0 at isa0 port 0xf0/16: reported by CPUID; using exception 16
fdc0 at isa0 port 0x3f0/6 irq 6 drq 2
fd0 at fdc0 drive 0: density unknown
fd1 at fdc0 drive 1: density unknown
biomask f7ed netmask ffed ttymask ffff
rd0: fixed, 3800 blocks
softraid0 at root
root on rd0a swap on rd0b dump on rd0b
erase ^?, werase ^W, kill ^U, intr ^C, status ^T

Welcome to the OpenBSD/i386 4.8 installation program.
(I)nstall, (U)pgrade or (S)hell? _
```

Figure 1. *Qemu snapshot*

```
$ qemu -nographic ex.bin
```

This will start QEMU in serial port emulation mode thereby avoiding X or SDL. This is nice since you can run QEMU on remote machines without using VNC or any graphics across the Internet.

You can run any TCP server like web server or a mail server inside QEMU even on really modest hardware by running QEMU as root.

```
# qemu -net nic -net tap ex.bin
```

Now this will start QEMU in bridge mode network and with that you can run just about software and no one will know that you are running on a virtual machine. In fact this is a simple way to run a VPS node. I simulate multiple network cards for some of my products using:

```
$ qemu -net nic,model=lance -net nic,model=pcnet -net
                    nic,model=rtl8139 ex.bin
```

You now have three NIC cards inside the guest. You can also get full audio support using SDL. If you want to simulate multiple hard disks for testing RAID or something you can use

```
$ qemu -hda first.bin -hdb second.bin
```

to use two hard disk images `first.bin` and `second.bin`. In fact the `.bin` extention is customary. You can name the file just about anything. QEMU can also emulate various sound cards just the same way you emulate network cards. I do not develop any audio products with OpenBSD. So I don't have much familiarity with it. But it allows you to run skype on Linux since OpenBSD does not have skype. As you can see QEMU can help you in much more ways than what this article covers. Just get familiar with it and you will automatically know what to do.

QEMU allows you a lot more possibilities than covered in this article but we will stop here now. Once you are satisfied with the results, just do the following to copy your OS into a USB stick from OpenBSD.

```
# dd if=ex.bin of=/dev/rsd0c bs=16k
```

That is it!

GIRISH VENKATACHALAM

Girish has close to 15 years of UNIX experience and he loves OpenBSD more than he loves anything else in the technology world.

Drupal on FreeBSD

Part 5

Continuing the series on the Drupal Content Management System, we will look at adding discrete PHP and Javascript code to our pages.

What you will learn…

- How to integrate PHP / JS and write a basic drupal module

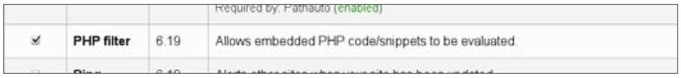
What you should know…

- Basic BSD / PHP skills and how to install / administer Drupal CMS (Parts 1, 2, 3 & 4))

Apart from the fact that Drupal is an efficient CMS that is based on PHP and MySQL, one of the major attractions of the software is the availability of many modules that extend the functionality of Drupal beyond that of a basic CMS. What happens though when no module is available that meets your requirements? Drupal provides numerous hooks that allow the developer to include custom code within the page itself or easily add a custom module where necessary.

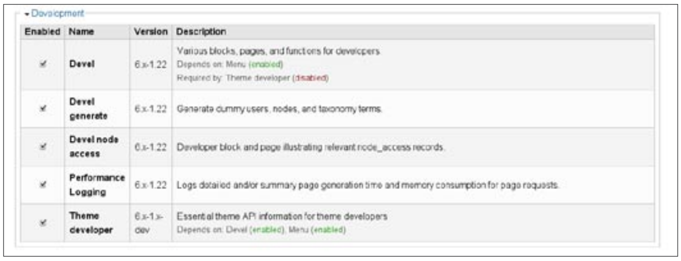
Installing and activating the modules

You will need the PHP filter, Devel and `devel_themer` modules installed and activated. The Devel and `devel_themer` modules will need to be downloaded from the Drupal website and extracted into the sites/all/modules



			Required by: Pathauto (enabled)
✓	**PHP filter**	6.19	Allows embedded PHP code/snippets to be evaluated.

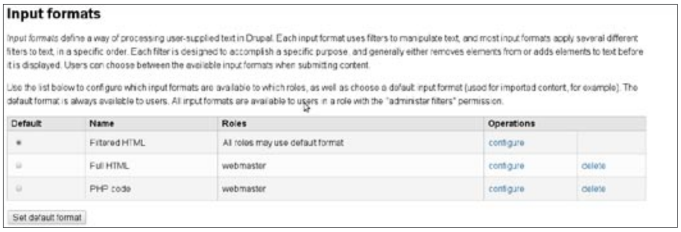
Figure 1. *PHP module enabled*



▾ Development

Enabled	Name	Version	Description
✓	**Devel**	6.x-1.22	Various blocks, pages, and functions for developers. Depends on: Menu (enabled) Required by: Theme developer (disabled)
✓	**Devel generate**	6.x-1.22	Generate dummy users, nodes, and taxonomy terms.
✓	**Devel node access**	6.x-1.22	Developer block and page illustrating relevant node_access records.
✓	**Performance Logging**	6.x-1.22	Logs detailed and/or summary page generation time and memory consumption for page requests.
✓	**Theme developer**	6.x-1.x-dev	Essential theme API information for theme developers Depends on: Devel (enabled), Menu (enabled)

Figure 2. *Developer tools and theme developer modules loaded*



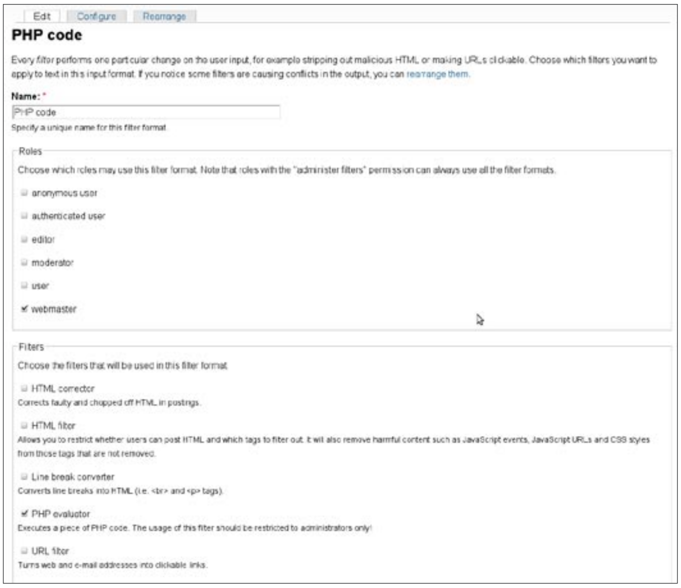
Input formats

Input formats define a way of processing user-supplied text in Drupal. Each input format uses filters to manipulate text, and most input formats apply several different filters to text, in a specific order. Each filter is designed to accomplish a specific purpose, and generally either removes elements from or adds elements to text before it is displayed. Users can choose between the available input formats when submitting content.

Use the list below to configure which input formats are available to which roles, as well as choose a default input format (used for imported content, for example). The default format is always available to users. All input formats are available to users in a role with the "administer filters" permission.

Default	Name	Roles	Operations	
●	Filtered HTML	All roles may use default format	configure	
○	Full HTML	webmaster	configure	delete
○	PHP code	webmaster	configure	delete

Figure 3. *Input filters*



PHP code

Every filter performs one particular change on the user input, for example stripping out malicious HTML or making URLs clickable. Choose which filters you want to apply to text in this input format. If you notice some filters are causing conflicts in the output, you can rearrange them.

Name: *
PHP code
Specify a unique name for this filter format.

Roles

Choose which roles may use this filter format. Note that roles with the "administer filters" permission can always use all the filter formats.

☐ anonymous user
☐ authenticated user
☐ editor
☐ moderator
☐ user
☑ webmaster

Filters

Choose the filters that will be used in this filter format.

☐ HTML corrector
Corrects faulty and chopped off HTML in postings.

☐ HTML filter
Allows you to restrict whether users can post HTML and which tags to filter out. It will also remove harmful content such as JavaScript events, JavaScript URLs and CSS styles from those tags that are not removed.

☐ Line break converter
Converts line breaks into HTML (i.e.
 and <p> tags).

☑ PHP evaluator
Executes a piece of PHP code. The usage of this filter should be restricted to administrators only!

☐ URL filter
Turns web and e-mail addresses into clickable links.

Figure 4. *PHP code filter*

directory as they are not part of the core system, unlike the PHP filter module (Figure 1 & 2).

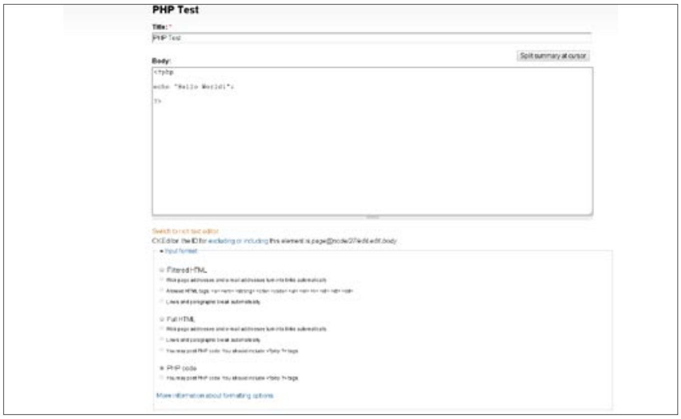
After enabling PHP filter, you will need to allow the relevant users access to the output filter that Drupal uses to display pages. Drupal is slightly unintuitive in that all code (HTML / PHP etc.) is parsed via an output filter, and while you may have embedded PHP in you page it may not execute if the correct filter is not set with the relevant permissions etc. As it is advisable that only administrators or trusted users have access to this module, ensure that the correct permissions are set (Figure 3 & 4).

If you are intending to access or manipulate database tables via your PHP code, it might be worthwhile to install webmin or phpmyadmin from the FreeBSD ports collection as this will allow you to access the DB via the browser rather than at the command line.

Our first PHP code snippet

In the true tradition, we will create a page that displays “Hello World!” using PHP. Create a new page, switch the plain text editor and change the input format to PHP code. Add the following code to the Body: and save the page (Figure 5, Listing 1).

Using the Drupal API, we can access the Node ID using a call to `node_load()`. Create a new page and add the following code: see Listing 2.


Split summary at cursor
Switch to rich text editor
Input format
Filtered HTML
Full HTML
PHP code
More information about formatting options

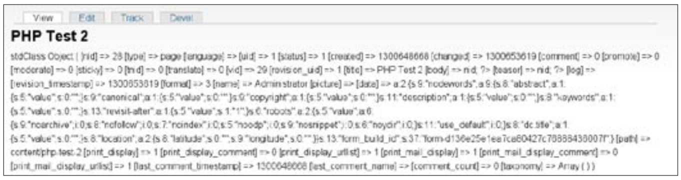
PHP Test

Title: *
PHP Test

Body:

```
<?php
echo "Hello World!";
?>
```

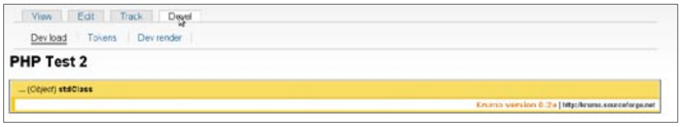
Figure 5. *Our first PHP code snippet*



PHP Test 2

```
stdClass Object ( [nid] => 28 [type] => page [language] => [uid] => 1 [status] => 1 [created] => 1300648668 [changed] => 1300653619 [comment] => 0 [promote] => 0
[moderate] => 0 [sticky] => 0 [tnid] => 0 [translate] => 0 [vid] => 29 [revision_uid] => 1 [title] => PHP Test 2 [body] => nid; ?> [teaser] => nid; ?> [log] =>
[revision_timestamp] => 1300653619 [format] => 3 [name] => Administrator [picture] => [data] => a:2:{s:9:"nodewords";a:9:{s:8:"abstract";a:1:
{s:5:"value";s:0:"";}s:9:"canonical";a:1:{s:5:"value";s:0:"";}s:9:"copyright";a:1:{s:5:"value";s:0:"";}s:11:"description";a:1:{s:5:"value";s:0:"";}s:8:"keywords";a:1:
{s:5:"value";s:0:"";}s:13:"revisit-after";a:1:{s:5:"value";s:1:"1";}s:6:"robots";a:2:{s:5:"value";a:6:
{s:9:"noarchive";i:0;s:8:"nofollow";i:0;s:7:"noindex";i:0;s:5:"noodp";i:0;s:9:"nosnippet";i:0;s:6:"noydir";i:0;}s:11:"use_default";i:0;}s:8:"dc.title";a:1:
{s:5:"value";s:0:"";}s:8:"location";a:2:{s:8:"latitude";s:0:"";s:9:"longitude";s:0:"";}}s:13:"form_build_id";s:37:"form-d136e25e1ea7ca60427c78888438007f";} [path] =>
content/php-test-2 [print_display] => 1 [print_display_comment] => 0 [print_display_urllist] => 1 [print_mail_display] => 1 [print_mail_display_comment] => 0
[print_mail_display_urllist] => 1 [last_comment_timestamp] => 1300648668 [last_comment_name] => [comment_count] => 0 [taxonomy] => Array ( ) )
```

Figure 6. *Array $node*



PHP Test 2

... (Object) stdClass

Krumo version 0.2a | http://krumo.sourceforge.net

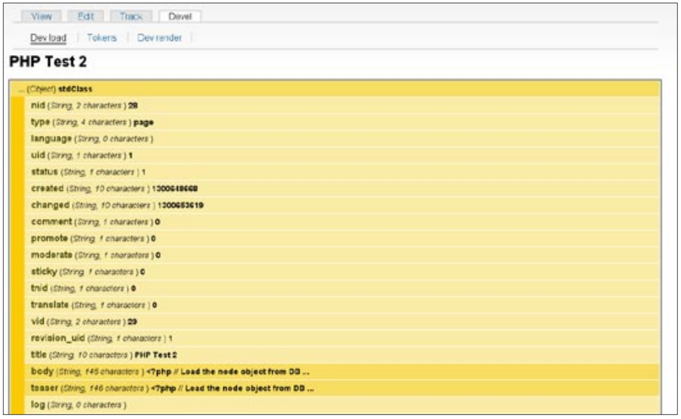
Figure 7. *Drilling down through the Devel tab*

This will dump the array `$node` (Figure 6). So rather than having to use print statements or referring to the API, how can we easily access the names of the variables and objects? Using the Devel tab we can examine the stdClass Object and using the `devel_theme` module we can click on a field and interrogate it’s theme properties (see Figure 7-13). Change the code to read: see Listing 3.

You will now get a page like Figure 14. Note that node 28 really is the full path for the page, if friendly URL’s are enabled this will be hidden.

Javascript

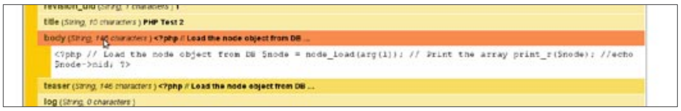
If the JS snippet is fairly short, it can be embedded in the body field: see Listing 4. This will create a JS alertbox when the page is accessed. If an external JS file is required, the following code could be used: see Listing 5.



PHP Test 2

```
... (Object) stdClass
    nid (String, 2 characters ) 28
    type (String, 4 characters ) page
    language (String, 0 characters )
    uid (String, 1 characters ) 1
    status (String, 1 characters ) 1
    created (String, 10 characters ) 1300648668
    changed (String, 10 characters ) 1300653619
    comment (String, 1 characters ) 0
    promote (String, 1 characters ) 0
    moderate (String, 1 characters ) 0
    sticky (String, 1 characters ) 0
    tnid (String, 1 characters ) 0
    translate (String, 1 characters ) 0
    vid (String, 2 characters ) 29
    revision_uid (String, 1 characters ) 1
    title (String, 10 characters ) PHP Test 2
    body (String, 145 characters ) <?php // Load the node object from DB ...
    teaser (String, 146 characters ) <?php // Load the node object from DB ...
    log (String, 0 characters )
```

Figure 8. *Using Devel to view stdClass*



```
    title (String, 10 characters ) PHP Test 2
    body (String, 146 characters ) <?php // Load the node object from DB ...
      <?php // Load the node object from DB $node = node_load(arg(1)); // Print the array print_r($node); //echo
      $node->nid; ?>
    teaser (String, 146 characters ) <?php // Load the node object from DB ...
    log (String, 0 characters )
```

Figure 9. *Body values shown in*

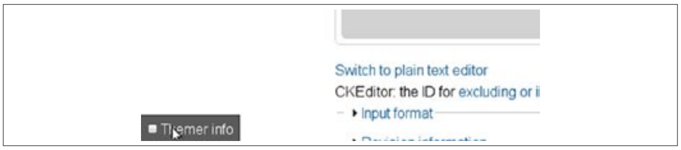
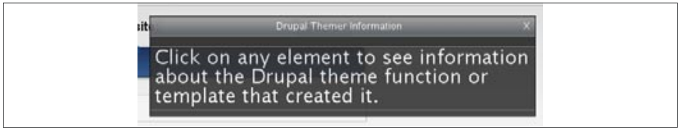

Switch to plain text editor
CKEditor: the ID for excluding or i
Input format
Themer info

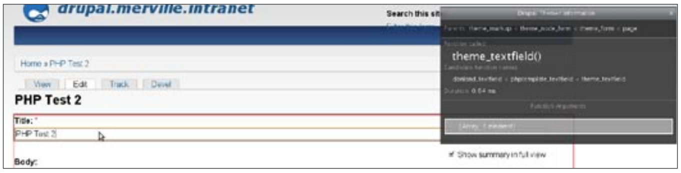
Figure 10. *The devel_theme control*



Drupal Themer Information

Click on any element to see information about the Drupal theme function or template that created it.

Figure 11. *Drupal Themer prompt*



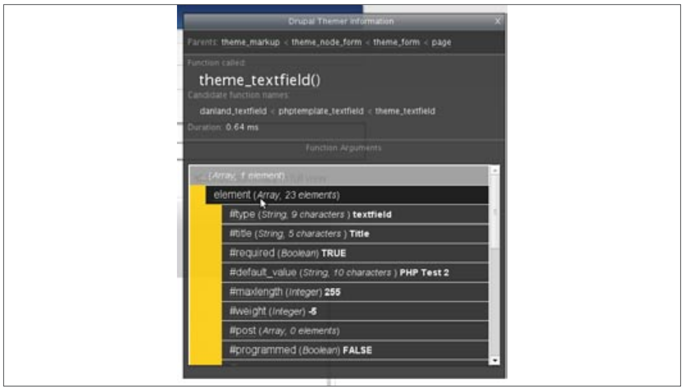
PHP Test 2

Title: *
PHP Test 2

Body:

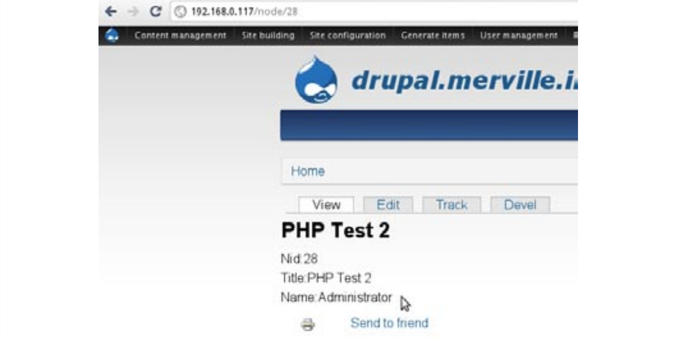
theme_textfield()

Figure 12. *The title field in theme_devel*



```
Parents: theme_markup < theme_node_form < theme_form < page
theme_textfield()
daniand_textfield < phptemplate_textfield < theme_textfield
Duration: 0.64 ms
(Array, 1 element)
element (Array, 23 elements)
#type (String, 9 characters ) textfield
#title (String, 5 characters ) Title
#required (Boolean) TRUE
#default_value (String, 10 characters ) PHP Test 2
#maxlength (Integer) 255
#weight (Integer) -5
#post (Array, 0 elements)
#programmed (Boolean) FALSE
```

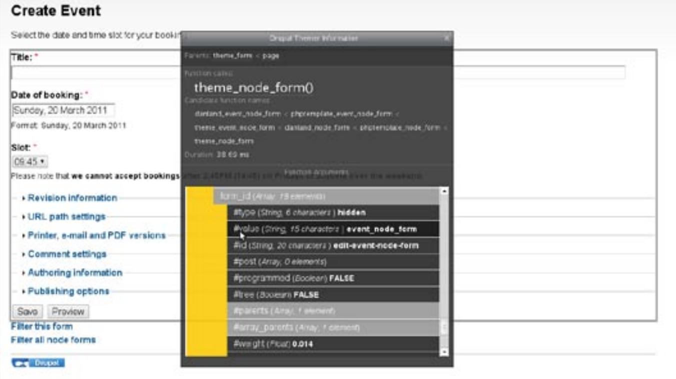
Figure 13. *The title field expanded in theme_textfield()*



PHP Test 2

Nid 28
Title PHP Test 2
Name Administrator

Figure 14. *Node 28*



Create Event

Select the date and time slot for your booki

Title: *

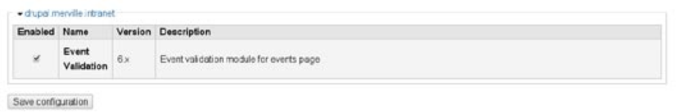
Date of booking: *
Sunday, 20 March 2011
Format: Sunday, 20 March 2011

Slot: *
09:45

Please note that **we cannot accept bookings**

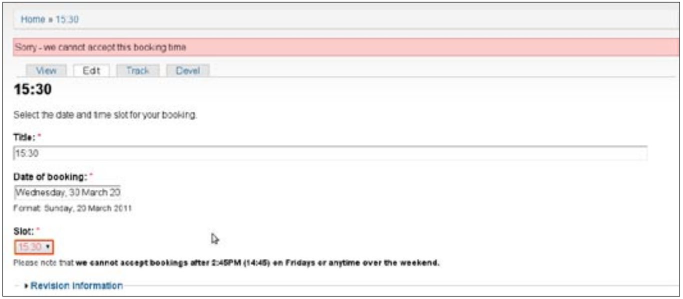
```
Parents: theme_form < page
theme_node_form()
danland_event_node_form < phptemplate_event_node_form <
theme_event_node_form < danland_node_form < phptemplate_node_form <
theme_node_form
Duration: 38.69 ms
form_id (Array, 13 elements)
#type (String, 6 characters ) hidden
#value (String, 15 characters ) event_node_form
#id (String, 20 characters ) edit-event-node-form
#post (Array, 0 elements)
#programmed (Boolean) FALSE
#tree (Boolean) FALSE
#parents (Array, 1 element)
#array_parents (Array, 1 element)
#weight (Float) 0.014
```

Figure 15. *Using devel_theme to locate form_id*



Enabled	Name	Version	Description
✓	Event Validation	6.x	Event validation module for events page

Figure 16. *New custom module added*



Sorry - we cannot accept this booking time

15:30

Select the date and time slot for your booking.

Title: *
15:30

Date of booking: *
Wednesday, 30 March 20
Format: Sunday, 20 March 2011

Slot: *
15:30

Please note that **we cannot accept bookings after 2:45PM (14:45) on Fridays or anytime over the weekend.**

Figure 17. *Slot failed validation*

Modules

In the last tutorial, we *hacked* a module to prevent an option allowing a user to create a duplicate booking from appearing. A better solution would be to write a custom module for validation of the field.

We will create a module called `event_validate.module` that will query the time selected, if 15:30 is selected, the user will not be allowed to save the form.

First create a directory called `event_validate` under *sites/all/modules*. Create two files there – `event_validate.info` and `event_validate.module` with the following code: see Listing 6.

`event _ validate.info` (see Listing 7).

```
event_validate_module
```

Enable the module. Create an event page and and save an event with a timeslot of 15:30 – it should fail (See screenshot 15-17).

Background

Using the Drupal API and the hooks `form_validate` and `form_alter`, we can easily customise our code to validate against certain logic. To reuse this code with your own forms, the module name, `form_id` and field values need to be changed as highlighted below: see Listing 8.

To Do

Add date logic to fail validation on all weekends and only after 15:30 on Fridays.

Debugging

While the Devel and Devel Themer modules are useful for highlighting variables passed during the page rendering process, for serious development work a more systemic approach is required. While Drupal provides extensive

Listing 1. *PHP hello World example*

```
<?php
  echo "Hello World!<br />";
?>
```

Listing 2. *Code to dump contents of $node*

```
<?php
    // Load the node object from DB
    $node = node_load(arg(1));
    // Print the array
    print_r($node);
?>
```

documentation about the API, dedicated Drupal developers will require a deeper understanding of the processes, hooks and flows within Drupal. One of the criticisms of the software is the learning curve that comes with it, and while a lot of this has been addressed from a GUI perspective in version 7.0, the Drupal developer will still need patience if they are exploit the full potential of coding themes, and modules etc. While it is relatively easy to perform on the fly modifications (e.g. adding or modifying a theme or module from the Drupal website) developing a theme or module from scratch is not such a straightforward process, especially if you want to harness the full power of the CMS.

Drupal developers tend to evolve through many stages, from adding echo or print commands in code through printing arrays at various stages of rendering the to a full blown development environment such as Quickstart. In reality, to get to know Drupal better *under the hood* from a developers perspective, a debugger and / or and IDE is an essential part of the tool-kit. Using this method, not only can you follow program flow from core to 3rd party

Listing 3. *Node 28 code example*

```
<?php
 // Load the node object from DB
 $node = node_load(arg(1));
 // Print out the values
 echo "Nid:" . $node->nid . "<br />";
 echo "Title:" . $node->title . "<br />";
 echo "Name:" . $node->name . "<br />";
?>
```

Listing 4. *Javascript example*

```
<?php
 // Load the node object from DB
 $node = node_load(arg(1));
 // Print out the values
 $js_alert_msg = "The page " . $node->title . " is
                   node " . $node->nid;
 drupal_add_js('alert("'.$js_alert_msg.'")',
                   'inline');
?>
```

Listing 5. *Adding an external javascript file*

```
<?php
  print drupal_add_js('path-to/javascript.js', 'theme');
?>
```

Listing 6. *event_validate.info*

```
;$Id$
name = "Event Validation"
description = "Event validation module for events page"
version = "6.x"
package = "drupal.merville.intranet"
```

Listing 7. *event_validate.module*

```
<?php
// $Id$
/**
* @file
* Module to run some basic validation on an event
*/

/*
* Implementation of hook_form_alter()
*
*/

function event_validate_form_alter(&$form, $form_state,
                   $form_id) {

  if ($form_id == 'event_node_form') {

     $form['#validate'] = array('_event_validate_event_
                   node_form_validate');


  }


  return $form;
}

/*
* Validates the event - throws an error if validation fails
*
*/

function _event_validate_event_node_form_validate($form,
                   &$form_state) {

  if($form_state['values']['field_slot'][0]['value'] == "15:30"){
     form_set_error('field_slot', t('Sorry - we cannot
                   accept this booking time'));
  }


}
```

modules, but you can also more easily understand the Drupal philosophy and coding / programming standards.

Xdebug

If you intend to perform any serious debugging, Xdebug is absolutely essential. Running as a PHP module on the web server, it provides stack and function traces, memory allocation, profiling information and code coverage analysis of PHP scripts. Xdebug can be used with a number of different IDE's as a remote debugger, and as such care must be taken to ensure that it is not run in a production environment accessible via the web, as critical information such as usernames and passwords etc. could be easily extracted by hostile users. Xdebug should therefore only be used in a development environment that is separate from and live systems. Xdebug supports a number of remote desktop editors/IDE's including Eclipse, Emacs, Komodo, NetBeans, PHPEclipse, Protoeditor and VIM. Installation on FreeBSD is just a matter of compiling and installing the PHP extension and adding a couple of lines to the php.ini file then restarting Apache to pick up the new module. See *www.xdebug.org* for more details.

Listing 8. *Sample module with module, form and value highlighted*

```
function event_validate_form_alter(&$form, $form_
                    state, $form_id) {
if ($form_id == 'event_node_form') {
$form['#validate'] = array('_event_validate_event_
                    node_form_validate');
}
return $form;
}
/*
* Validates the event - throws an error if validation fails
*
function _event_validate_event_node_form_
                    validate($form, &$form_state) {
if($form_state['values']['field_slot'][0]['value'] == "15:
                    30"){
form_set_error('field_slot', t('Sorry - we cannot accept
                    this booking time'));
}
}
event_validate = MODULE NAME
event_node_form = Form ID (Use devel themer module to
                    interrogate)
['field_slot'][0]['value'] = Value of field to validate (Use
                    Devel to interrogate)
field_slot = Field to highlight as error
```

On the 'Net

- *http://api.drupal.org/api/drupal/6* – Drupal API reference
- *http://drupal.org/* – Drupal website
- *http://drupal.org/node/337959* – A beginners guide to snippets

Additional modules used

```
Devel
devel_themer
```

Emacs, Vim, Netbeans and Eclipse are available for either the non FreeBSD environment or via ports and packages if you are running FreeBSD as your desktop.

Using VIM and and Xdebug

There are a number of scripts available on *www.vim.org* that support Xdebug, download and extract into your Vim plugins directory. You will need to toggle between Vim and your browser to start the debug session.

Using Xdebug and Netbeans

The Netbeans IDE supports Xdebug but watches and balloon help is known to be unstable. The current version 6.9.1 is available from *netbeans.org*. Details on how to configure Xdebug is available at *wiki.netbeans.org*.

Using Xdebug and Eclipse IDE

The Eclipse environment is available from *www.eclipse.org* and instructions for configuring Xdebug can be found in the PDF file XdebugGuide.pdf available on the website.

Using the Drupal Quickstart Virtual Box Image

If you have Virtualbox installed on your PC and sufficient resources (at least 2Gb RAM free) Quickstart is fully configured development environment out of the box. Supporting both Drupal 7 and Drupal 6 it provides Apache, MySQL, XDebug, Firefox (with developer plugins), Netbeans and Eclipse, XHProf/Webgrind profilers and version control (cvs, subversion, git and bazaar). Running on top of a Ubuntu 10.10 desktop, once you have downloaded the image you can be up and running and debugging in a separate development environment in minutes. Quickstart is available via bit torrent from the Drupal website at *http://drupal.org/project/quickstart*.

ROB SOMERVILLE

Rob Somerville has been passionately involved with technology both as an amateur and professional since childhood. A passionate convert to *BSD, he stubbornly refuses to shave off his beard under any circumstances. Fortunately, his wife understands him (she was working as a System/36 operator when they first met). The technological passions of their daughter and numerous pets are still to be revealed.

Mutt On OS X

Part II

Last time (BSD Magazine 02/2011), we installed Mutt on OS X and read and sent mail from a Gmail account. This month, we'll get one step closer to replacing Mail.app by learning a way to handle multiple accounts and how to search our Mac's Address book from within Mutt.

What you will learn…

- a method of configuring mutt to use multiple mail accounts
- how to search your Mac Address Book from mutt

What you should know…

- how to use macports to install programs
- it's recommended you read part I first, see BSD Magazine 02/2011

No one these days has 1 email account, so to make Mutt more useful let's get straight to configuring it so that it can handle more than one account. There are, of course, many ways one can go about this. In this case, we're going to use Mutt's account and folder *hooks*. The account-hooks will be for connection parameters such as username and password, and the folder-hooks will be for setting other parameters that might change on a per-account basis.

When using hooks, it's recommended to begin with a default. Let's continue by editing the .muttrc file we wrote last time[1] by adding the following above the existing Gmail account configuration:

```
# Default hooks
account-hook . ‘unset imap_user; unset imap_pass ‘
folder-hook  . ‘set folder=~/Mail’
```

These default hooks will make sure that our IMAP username and password will be reset by default, and also set our folder to a local mail folder in the home directory. It's ok if the `~/Mail` folder doesn't exist – you can change it to a local folder that does exist, if you prefer.

The next step is to define the account hook for our gmail account. Using separate files to hold the hooks helps to keep the configuration easier to follow, so let's create a file named gmail-account-hook. You can put this file wherever you like, as long as it's in a safe location. I keep all of my Mutt configuration files in a directory named `~/.mutt`, and my hooks are in a subdirectory. Actually, I've gone another step and created subfolders for account and folder hooks, so in this instance, the path to my gmail account-hook file is `~/.mutt/hooks/accounthooks/gmail-account-hook`. Here are the contents:

```
# vim: set ft=muttrc:
set imap_user=”mygmailuser”
set imap_pass=`security find-internet-password -g -r imap \
   -s imap.gmail.com 2>&1 >/dev/null | cut -d\” -f2`
```

The comment line (beginning with `#` vim) is there to help vim determine what file type this account-hook text file is. It tells vim to color it using the muttrc syntax highlighting, which I find makes things easier to read. It's completely optional. As you can see, the rest of the file is simply the `imap _ user` and `imap _ pass` lines from our existing Gmail configuration. If you like, you can cut & paste the lines from the existing configuration. It's important that they appear only in the gmail-account-hook file.

Now we can create the folder-hook file. The path to mine is `~/.mutt/hooks/folderhooks/gmail-folder-hook`. The contents are as follows:

```
set use_from
set envelope_from
set smtp_url=”smtps://mygmailuser@smtp.gmail.com”;
set smtp_pass=`security find-internet-password -g -s
                smtp.gmail.com \
   2>&1 >/dev/null | cut -d\” -f2`’
```

Again, these are lines that we already had in our *.muttrc*, but we're moving them to the folder-hook file. There is more you can add to your folder hook, but for now we are going to keep it very simple. The next step is to edit the muttrc so that it will tell Mutt to source our hook files. To do that, add the following to the muttrc so that it looks like this:

```
# Account settings for Gmail
account-hook imaps://imap.gmail.com  \
   source ~/.mutt/hooks/accounthooks/gmail-account-hook

 folder-hook imaps://imap.gmail.com/INBOX \
       source ~/.mutt/hooks/folderhooks/gmail-folder-hook

    mailboxes imaps://imap.gmail.com/INBOX
```

At this point we still have only one account configured, but it's still only one account! We are all set to handle multiple accounts, however. All you need to do is make copies of your hook files and adjust them to the settings of your other account. For example, you might have an IMAP account named *work@workserver.com* – for this account you'd copy the gmail-account-hook and gmail-folder-hook files and change the account information so that the username, password, server name, and other relevant bits are those for the work server. Then you'd adjust the .muttrc and and additional account-hook and folder-hook lines, which will tell Mutt to source the new files. Don't forget to adjust the mailboxes line to include your second account Inbox! The complete sample configuration files are available for download, see the links below[2].

To change between accounts, you use the c command. This and the rest of Mutt's commands are thoroughly documented, and if you haven't been to the Mutt Wiki, it's about time! Check *http://wiki.mutt.org/?MuttGuide/UseIMAP* for details on what has been covered so far.

Now that we have multiple accounts configured, let's move on to querying our address book. Mutt offers it's own sort of address book, using what are called *aliases*, but since we're on a Mac, we have an app called Address Book, and there is a way to use it from Mutt! In order to make this work, you'll need to install The Little Brother's Database, lbdb (*http://www.spinnaker.de/lbdb/*). The Little Brother's Database is a great unix tool that allows us to query several different types of database, including LDAP, finger, and of course, the OS X address book. All you need to do is install lbdb (I recommend using macports to do this), and then add the following to your .muttrc:

On the 'Net

- *http://www.culmination.org/Mike/sample-muttrc [1]*
- *http://www.culmination.org/Mike/sample-mutt-configs.zip [2]*

```
# Use lbdbq for address queries
set query_command=”lbdbq ‘%s’”
```

Now when you want to look up an address when you're composing a new email, you can start by typing the first few letters of the address, then press CTRL-T and Mutt will query the address book for addresses that start with what you've entered so far. Then you choose from the list, and that's it!

While many wouldn't believe it, Mutt has the potential to replace Mail.app on your Apple computer, but it's not for the faint of heart. Advanced Mutt configuration can become very complex and varies depending on each user's needs. We're all different, and for this reason among others, it can be difficult to find a tutorial on the Internet that matches your needs specifically. In my last 2 articles I've attempted to give you just a small sample of what Mutt can do for you – there's no way to tell it all in this small space! Mutt is extremely flexible, and since it was designed as a unix tool it can work with many other unix tools to provide a complete and robust experience. Using multiple email accounts and searching your address book is just the beginning. With Mutt you can read and send encrypted mail, view HTML email with any browser you choose, open any attachment your mac can handle, and more! If you're interested, there are many resources available on the web, although most are not OS X specific. Luckily you can use Mail.app while you figure it all out. Is it necessary? Of course not! Is it fun? Absolutely! :)

MICHAEL HERNANDEZ

Mike is an IT consultant and web programmer. He lives in Brooklyn, New York, and he and his wife are celebrating their one year anniversary on February 14th. He also loves electronic dance music and commuting on his fixed gear bike, appropriately named Constance.

BSDCAN 2011

THE BSD EVENT OF 2011
http://www.bsdcan.org/

Ottawa, Canada

BSDCan 2011 – The event to be at this year

BSDCAN 2011

There's only one major BSD Event in North America in 2011: BSDCan

WHERE

Ottawa, Canada

WHEN

Early May 2011, with two days of tutorials before the conference. Exact dates to follow.

WHO

Contributors, developers, and users

VENUE

University of Ottawa
http://www.uottawa.ca/

AT FEES YOU CAN AFFORD

We plan to keep costs to a minimum. As such, the conference will be held at University of Ottawa and accommodation is available within the University residences. Hotels are also within close walking distance of the conference venue.

WHAT DOES IT COST?

Type	CAD
Regular	$195
Corporate	587
Additional Corporate	$120
Student	57
Tutorial	57

Comfortable accommodation is available on campus at very reasonable rates. See our website for details.

Take the BSDA Certification exam.
For details see
http://bsdcertification.org/

SCHEDULE OVERVIEW

Wednesday

4:00 pm	Sign-in desk opens at a local pub. Get your registration pack and have a drink.
8:00 pm	Sign-in desk closes.

Thursday

9:30 am	Opening words
11:00 am	First set of talks
12:00 pm	lunch
1:00 pm	Second set of talks
2:00 pm	break
2:30 pm	Third set of talks
3:30 pm	break
4:00 pm	Fourth set of talks
5:00 pm	Key Signing Party

Friday

10:00 am	First set of talks
11:00 am	break
11:30 am	Second set of talks
12:30 pm	lunch
1:30 pm	Third set of talks
2:30 pm	break
3:00 pm	Fourth Set of Talks
4:00 pm	Fifth Set of Talks
5:00 pm	Closing words

Sat

8:30 am	Breakfast
9:30-4:00	Tourist fun

TALKS FROM 2010

Please see the website for complete details.

- ClangBSD - Replacing gcc with clang
- Consideration for the BSD Professional Exam
- Security Implications of the Internet Protocol version 6 (IPv6)
- Puffy At Work -- Getting Code Right And Secure, The OpenBSD Way
- Everything you need to know about cryptography in 1 hour
- Networking from the Bottom Up: IPv6
- Porting dummynet to Linux and Windows
- Journaled Soft-Updates
- Porting hwpmc to non x86 platforms
- Maintaining a Customized FreeBSD Distribution
- Debuggers - Architecture and Implementation
- pfSense 2.0
- Networking from the Bottom Up: Ipv6
- The New VVorld
- A new packet scheduling architecture for FreeBSD

SOCIAL ACTIVITIES

It's not all work. Social activities play a major role in project development.

Wednesday

4:00 pm Drinks + registration at a local pub

Thursday

4:30 pm BOFs
7:00 pm Gathering at local eateries for dinner

Friday

4:30 pm Key signing party
7:00 pm Gathering at local pubs for drinks

Saturday

8:30 am Breakfast

afterwards: various tourist-type things

To stay informed, please join our announcement mailing list. Details at http://www.bsdcan.org/

2010 PLATINUM SPONSOR

OUR 2010 GOLD SPONSORS

The FreeBSD Foundation Logo is the trademark
of the FreeBSD Foundation

Realtime Weather Data

EMWIN on FreeBSD

Have ever run to the TV, turned on a radio, or browsed to a weather site, just to find out what the weather conditions are, or about to become? You can now have data delivered right to server, use in a web site, or sent as notifications to pagers via e-mail.

What you will learn…

- What is EMWIN is, and who uses it
- What Npemwin is.
- How to install and use the Npemwin client
- What kind of information does it make available

What you should know…

- How to edit text files with "vi" or other editors
- How to execute commands from the root CLI shell
- How to setup basic networking and Internet access for FreeBSD
- Use of a web browser

In this article, I'll show you how to install *npemwin*, an open source EMWIN client for FreeBSD. EMWIN (*Emergency Management Weather Information Network*) is a low-speed data service provided by NOAA and the National Weather Service. The data rate of the products is approximately 100k per minute, and will work well over even a dial-up modem. The family of products consists of text, graphic (GIF/JPEG) files. All priority watch/warning products are sent over the EMWIN network usually within 60 seconds from time of issuance.

Npemwin is a C application with embedded Tcl. Tcl is used for processing text files, and simple to extend for nearly any purpose. It is both a client and a server. It includes it's own web server (running on port 8016).

Installation

This example install will be based on FreeBSD 8.2-RELEASE i386 platform, but the directions are the similar if you are using FreeBSD amd64 version.

I use a browser on my PC, and a Terminal/xterm up logged into my FreeBSD server, to do the following:

The FreeBSD package system has a few problems that make installation a bit challenging, especially in the area of dependancies. Gnuplot for example, has a a broken dependency, in libpdf. We'll install the broken one, then uninstall it, and install a lite version, without such dependancies. This extra step pulls in needed packages for npemwin as well. You will need a broadband connection to install the packages.

Now, switch to the shell window on the FreeBSD box, su to root, and cd to some place to store the packages, like `/usr/src` for example.

In the browser window, browse over to *http://www.noaaport.net* site and grab two packages (gnuplot, and npemwin):

Click on:

- Downloads (left pane near top)
- `packages`
- `optional packages`
- `freebsd-8.2-i386`
- Right click, or control click to get the URL location of the `gnuplot-4.4.0.tbz` file
- type `fetch` and then paste the URL of the location, press enter to download it.

Click on:

- Downloads again in the browser
- `packages`
- `npemwin-2.1.2p1`
- `packages`
- `freebsd-8.2`

- `i386`
- and finally right/control click on `npemwin-2.1.2p1 _ 1.tbz` to copy the URL location
- in the terminal window type "*fetch and paste that path in..*
  fetch url...

Now, we are ready to install the packages (don't type the // and after)

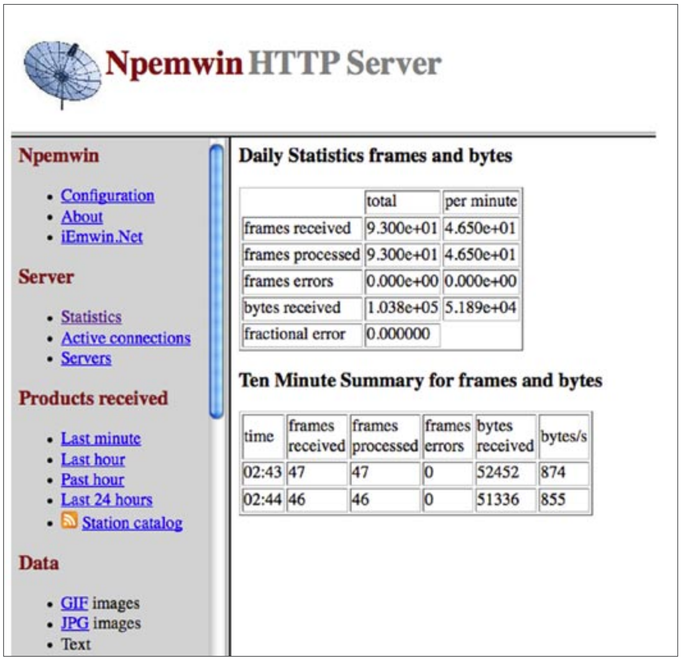
```
# pkg_add -r tcl86 tk86             // Pulls in a few of
                    the needed packages
# pkg_add -r gnuplot                // this will take a
                    while.. ignore the warnings.
# pkg_delete gnuplot-4.4.0          // delete the just
                    installed stock gnuplot pkg
# pkg_add ./gnuplot-4.4.0.tbz       // Install our "lite"
                    version of gnuplot
# pkg_add -r db48 tcllib unzip      // install the final
                    dependent packages
# pkg_add ./npemwin-2.1.2p1_1.tbz   // install npemwin
                    application
```

Configuration

At this point, you have installed the npemwin package, and all is left is configuring for client operation. Edit

```
/usr/local/etc/npemwin/bbserver.conf
```

copy and place the following line, if you just want to receive the emwin feed (client) and not be a server:



Daily Statistics frames and bytes

	total	per minute
frames received	9.300e+01	4.650e+01
frames processed	9.300e+01	4.650e+01
frames errors	0.000e+00	0.000e+00
bytes received	1.038e+05	5.189e+04
fractional error	0.000000	

Ten Minute Summary for frames and bytes

time	frames received	frames processed	frames errors	bytes received	bytes/s
02:43	47	47	0	52452	874
02:44	46	46	0	51336	855

Figure 1. *Npemwin Status Page*

```
set bbserver(privacy) „SPR";    # SPR for private, or SPU
                   for public
```

You'll need a public IP address, and bandwidth if you wish to serve the EMWIN data feeds. Add the text

```
"npemwind_enable=1" to your /etc/rc.conf
```

Note, you will need working DNS for your server, or some at least a reasonable entry in your `/etc/hosts` file, such as:

```
192.168.1.127yourhost.yourdomain.com yourhost
```

Start Your FEED

Start up the application by:

```
   /usr/local/etc/rc.d/npemwind start
```

Wait a minute or two, then point your browser at *http://server_ip:8016* where *server_ip* is your FreeBSD server ip address or FQDN (see Figure 1).

Your now receiving a realtime National Weather Service EMWIN feed, with the latest weather forecasts, station conditions, watch and warning information, as well as some amazing graphics. The majority of the EMWIN data products are *finished*, meaning they are typically useable as is, being plain text, or graphic files.

The products are organized by the destination FCC station identification. This is usually a nearby airport.

Npemwin is self maintaining, and will purge files automatically after 24 hours, so long as you have a couple of gigabytes free in your `/var` partition.

Dr. Jose Nieves, the author of Npemwin, has included some simple plots of the station data (METAR). Let npemwin run for a few hours, then click on the *Metar* item in the left pane. Npemwin will plot the accrued data for various atmospheric values, like pressure, wind, temperature, and humidity (Figure 2).

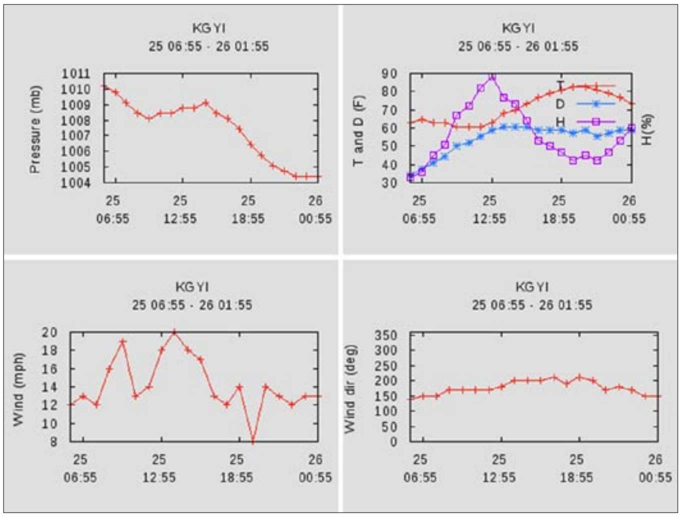


Figure 2. *Weather Station Conditions plots*

On the 'Net

- EMWIN: *http://www.weather.gov/emwin/user-intro.htm*
- Npemwin: *http://www.noaaport.net/*
- *http://www.iemwin.net*

It's worth noting that all times listed, from the status screens, to the plots, are in UTC time. Subtract your relevant difference (mine, in CDT is -5 hours for example).

Watches and warnings are issued as well by FCC station id. The products are named by defined list of product names by category. This list can be found at *http://www.weather.gov/emwin/windat.htm*.

One of my favorites is *AFDxxxx.TXT*. This is the Area Forecast Discussion, and usually has some very fascinating comments from Meteorologists, regarding what they believe is going to happen in regards to the weather. There are excellent plots for hurricane tracking as well as some beautiful full earth satellite images.

There is so much more that Npemwin can do, such as notifications, acting as a ByteBlaster server for other Npemwin/ByteBlaster clients. Dr. Nieves has produced some excellent documentation available at the application website, *http://www.noaaport.net*.

Check out my npemwin site at *http://www.wxpro.net:8016*. It's currently serving clients and in production. I include a few products on my main website as an example.

BILL HARRIS

Bill Harris is a Unix Systems Administrator and programmer, in the Texas area. His 20+ years of experience spans a variety of operating systems from DEC Ultrix, to modern BSD and Linux based varient in use today. He is also a General Class Amateur Radio Operator, and has a degree in Computer Science/Computer Information Systems.

Benchmarking Different Kind of Storage

In this article we will examine 2 types of storage: an iSCSI and a local hard drive.

What you will learn…

- Basic networking
- Something about benchmarking.
- What is „iSCSI" and how to use it.

What you should know…

- Very interesting benchmark results for an iSCSI and a local storage.
- The difference between the performance for the iSCSI and local storage

- What impacts the performance of the iSCSI traffic.

We will do all of our tests with a specific set of hardware and software to illustrate performance. Everything depends on the ethernet hardware, including Ethernet *network interface cards* (NICs) that incorporate a *TCP/IP Offload Engine* (TOE). TOE chips are designed to offload some or all of the work required to process TCP/IP data and this makes reduction of the workload on the local CPU. Another part of the puzzle that reduces the iSCSI traffic are the ethernet switshes. The ethernet switch introduces latency into iSCSI traffic, and this reduces performance. Many people suggest deploying high-performance ethernet switches that have low-latency ports. In order to improve the iSCSI performance we can separate our network logically and physically.

For example, iSCSI traffic should never be mixed with other ethernet user traffic from offices, datacenters and others. Let's proceed with the tests: see Fisting 1-20.

From what we see in the test results above, we can say that the local storage is basically faster than the iSCSI one and it is because of some limitations in the network connectivity. The iSCSI performance depends on the network throughput. Our primary target is to compare a usual local storage with a single hard drive with the same usual iSCSI storage with one hard drive. The benefit of this comparison will be to see the difference in the types of the storages. That is what we want.

Let's have a look at the benchmark results. The first parameter is `âeśTimeâeť`. This is the completion time for

Listing 1. *Results for 1 concurrent I/O threads*

```
,-------------------------------------------------------------------------------.
| Item                  | iSCSI Time |  iSCSI Rate  | Local Disk |  Local Disk  |
|                       |            |              |    Time    |     Rate     |
+-----------------------+------------+--------------+------------+--------------+
| Write           1 MBs |    0.3 s   |   3.648 MB/s |    0.0 s   |  58.473 MB/s |
| Random Write    4 MBs |    0.2 s   |  25.779 MB/s |    0.0 s   | 166.862 MB/s |
| Read            1 MBs |    0.0 s   | 539.084 MB/s |    0.0 s   | 536.193 MB/s |
| Random Read     4 MBs |    0.0 s   | 521.111 MB/s |    0.0 s   | 102.167 MB/s |
'---------------------------------------------------------------------------'
```

Listing 2. *Latency results*

```
,--------------------------------------------------------------------------------------.
| Item         |      iSCSI      |      iSCSI      |   Local Disk    |   Local Disk    |
|              | Average latency | Maximum latency | Average Latency | Maximum Latency |
+--------------+-----------------+-----------------+-----------------+-----------------+
| Write        |     0.026 ms    |     0.100 ms    |     0.021 ms    |    0.559 ms     |
| Random Write |     0.033 ms    |     0.485 ms    |     0.010 ms    |    0.569 ms     |
| Read         |     0.005 ms    |     0.034 ms    |     0.005 ms    |    0.033 ms     |
| Random Read  |     0.005 ms    |     0.021 ms    |     0.036 ms    |    30.544 ms    |
|--------------+-----------------+-----------------+-----------------+-----------------|
| Total        |     0.018 ms    |     0.485 ms    |     0.021 ms    |    30.544 ms    |
'--------------+-----------------+-----------------+-----------------+-----------------'
```

Listing 3. *Results for 4 concurrent I/O threads*

```
,-------------------------------------------------------------------------------.
| Item                  | iSCSI Time |  iSCSI Rate  | Local Disk |  Local Disk  |
|                       |            |              |    Time    |     Rate     |
+-----------------------+------------+--------------+------------+--------------+
| Write           4 MBs |    1.0 s   |   3.817 MB/s |    0.1 s   |  49.285 MB/s |
| Random Write   16 MBs |    1.5 s   |  10.720 MB/s |    0.2 s   |  64.680 MB/s |
| Read            4 MBs |    0.0 s   | 941.620 MB/s |    0.0 s   | 695.410 MB/s |
| Random Read    16 MBs |    0.0 s   | 345.518 MB/s |    0.0 s   | 778.486 MB/s |
'-------------------------------------------------------------------------------'
```

Listing 4. *Latency results*

```
,--------------------------------------------------------------------------------------.
| Item         |      iSCSI      |      iSCSI      |   Local Disk    |   Local Disk    |
|              | Average latency | Maximum latency | Average Latency | Maximum Latency |
+--------------+-----------------+-----------------+-----------------+-----------------+
| Write        |     0.055 ms    |    11.051 ms    |     0.020 ms    |     0.946 ms    |
| Random Write |     0.503 ms    |   483.575 ms    |     0.059 ms    |    61.949 ms    |
| Read         |     0.006 ms    |     0.158 ms    |     0.005 ms    |     0.044 ms    |
| Random Read  |     0.005 ms    |     0.028 ms    |     0.011 ms    |     8.010 ms    |
|--------------+-----------------+-----------------+-----------------+-----------------|
| Total        |     0.209 ms    |   483.575 ms    |     0.030 ms    |    61.949 ms    |
'--------------+-----------------+-----------------+-----------------+-----------------'
```

Listing 5. *Results for 8 concurrent I/O threads*

```
,-------------------------------------------------------------------------------.
| Item                  | iSCSI Time |  iSCSI Rate  | Local Disk |  Local Disk  |
|                       |            |              |    Time    |     Rate     |
+-----------------------+------------+--------------+------------+--------------+
| Write           8 MBs |    1.8 s   |   4.443 MB/s |    0.2 s   |  32.630 MB/s |
| Random Write   31 MBs |    4.7 s   |   6.619 MB/s |    0.6 s   |  53.960 MB/s |
| Read            8 MBs |    0.0 s   |1032.125 MB/s |    0.0 s   | 272.035 MB/s |
| Random Read    31 MBs |    0.0 s   |1087.108 MB/s |    0.1 s   | 262.387 MB/s |
'-------------------------------------------------------------------------------'
```

Listing 6. *Latency results*

```
,--------------------------------------------------------------------------------------.
| Item          |      iSCSI      |      iSCSI      |    Local Disk   |    Local Disk   |
|               | Average latency | Maximum latency | Average Latency | Maximum Latency |
+--------------+-----------------+-----------------+-----------------+-----------------+
| Write         |     0.046 ms    |     11.051 ms   |     0.078 ms    |     39.889 ms   |
| Random Write |     2.507 ms    |   2461.516 ms   |     0.199 ms    |    385.079 ms   |
| Read          |     0.005 ms    |      0.053 ms   |     0.006 ms    |      0.454 ms   |
| Random Read   |     0.005 ms    |      0.111 ms   |     0.040 ms    |     76.700 ms   |
|--------------+-----------------+-----------------+-----------------+-----------------|
| Total         |     1.005 ms    |   2461.516 ms   |     0.104 ms    |    385.079 ms   |
'--------------+-----------------+-----------------+-----------------+-----------------'
```

Listing 7. *Results for 16 concurrent I/O threads*

```
,-------------------------------------------------------------------------------.
| Item                  | iSCSI Time |  iSCSI Rate  | Local Disk |  Local Disk  |
|                       |            |              |    Time    |     Rate     |
+-----------------------+------------+--------------+------------+--------------+
| Write           16 MBs |    3.0 s   |   5.297 MB/s |    0.6 s   |  28.942 MB/s |
| Random Write    62 MBs |    6.2 s   |  10.015 MB/s |    3.5 s   |  17.888 MB/s |
| Read            16 MBs |    0.0 s   | 697.989 MB/s |    0.0 s   | 417.591 MB/s |
| Random Read     62 MBs |    0.1 s   | 955.249 MB/s |    0.1 s   | 824.821 MB/s |
'-------------------------------------------------------------------------------'
```

Listing 8. *Latency results*

```
,--------------------------------------------------------------------------------------.
| Item          |      iSCSI      |      iSCSI      |    Local Disk   |    Local Disk   |
|               | Average latency | Maximum latency | Average Latency | Maximum Latency |
+--------------+-----------------+-----------------+-----------------+-----------------+
| Write         |     0.059 ms    |     32.998 ms   |     0.040 ms    |     20.877 ms   |
| Random Write |     3.535 ms    |   1554.787 ms   |     2.147 ms    |   1178.787 ms   |
| Read          |     0.006 ms    |      2.112 ms   |     0.005 ms    |      0.044 ms   |
| Random Read   |     0.010 ms    |     47.205 ms   |     0.011 ms    |     61.249 ms   |
|--------------+-----------------+-----------------+-----------------+-----------------|
| Total         |     1.418 ms    |   1554.787 ms   |     0.863 ms    |   1178.787 ms   |
'--------------+-----------------+-----------------+-----------------+-----------------'
```

Listing 9. *Results for 32 concurrent I/O threads*

```
,-------------------------------------------------------------------------------.
| Item                  | iSCSI Time |  iSCSI Rate  | Local Disk |  Local Disk  |
|                       |            |              |    Time    |     Rate     |
+-----------------------+------------+--------------+------------+--------------+
| Write           32 MBs |    6.7 s   |   4.800 MB/s |    1.4 s   |  22.645 MB/s |
| Random Write   125 MBs |   24.7 s   |   5.060 MB/s |    1.7 s   |  75.039 MB/s |
| Read            32 MBs |    0.1 s   | 558.357 MB/s |    0.1 s   | 293.390 MB/s |
| Random Read    125 MBs |    0.1 s   | 953.973 MB/s |    0.1 s   | 845.229 MB/s |
'-------------------------------------------------------------------------------'
```

Listing 10. *Latency results*

```
,--------------------------------------------------------------------------------------.
| Item         |      iSCSI      |      iSCSI      |   Local Disk    |   Local Disk    |
|              | Average latency | Maximum latency | Average Latency | Maximum Latency |
+--------------+-----------------+-----------------+-----------------+-----------------+
| Write        |        0.099 ms |      124.861 ms |     0.110 ms    |     68.097 ms   |
| Random Write |       16.599 ms |    15118.814 ms |     0.282 ms    |   1298.109 ms   |
| Read         |        0.012 ms |       55.172 ms |     0.007 ms    |      5.553 ms   |
| Random Read  |        0.008 ms |       88.986 ms |     0.020 ms    |    115.280 ms   |
|--------------+-----------------+-----------------+-----------------+-----------------|
| Total        |        6.622 ms |    15118.814 ms |     0.132 ms    |   1298.109 ms   |
'--------------+-----------------+-----------------+-----------------+-----------------'
```

Listing 11. *Results for 1 concurrent I/O threads*

```
,-------------------------------------------------------------------------------.
| Item                  | iSCSI Time |  iSCSI Rate  | Local Disk |  Local Disk  |
|                       |            |              |    Time    |     Rate     |
+-----------------------+------------+--------------+------------+--------------+
| Write          10 MBs |    1.6 s   |   6.187 MB/s |   0.5 s    |  20.358 MB/s |
| Random Write    4 MBs |    0.8 s   |   4.733 MB/s |   0.5 s    |   7.118 MB/s |
| Read           10 MBs |    0.0 s   | 521.594 MB/s |   0.1 s    | 134.882 MB/s |
| Random Read     4 MBs |    0.0 s   | 567.686 MB/s |   0.0 s    | 218.715 MB/s |
'-----------------------+------------+--------------+------------+--------------'
```

Listing 12. *Latency results*

```
,--------------------------------------------------------------------------------------.
| Item         |      iSCSI      |      iSCSI      |   Local Disk    |   Local Disk    |
|              | Average latency | Maximum latency | Average Latency | Maximum Latency |
+--------------+-----------------+-----------------+-----------------+-----------------+
| Write        |     0.020 ms    |     0.081 ms    |     0.072 ms    |    20.651 ms    |
| Random Write |     0.013 ms    |     0.070 ms    |     0.012 ms    |     0.934 ms    |
| Read         |     0.006 ms    |     0.072 ms    |     0.027 ms    |    23.674 ms    |
| Random Read  |     0.005 ms    |     0.017 ms    |     0.016 ms    |    10.580 ms    |
|--------------+-----------------+-----------------+-----------------+-----------------|
| Total        |     0.012 ms    |     0.081 ms    |     0.039 ms    |    23.674 ms    |
'--------------+-----------------+-----------------+-----------------+-----------------'
```

Listing 13. *Results for 4 concurrent I/O threads*

```
,-------------------------------------------------------------------------------.
| Item                  | iSCSI Time |  iSCSI Rate  | Local Disk |  Local Disk  |
|                       |            |              |    Time    |     Rate     |
+-----------------------+------------+--------------+------------+--------------+
| Write          40 MBs |   10.6 s   |   3.775 MB/s |    1.4 s   |  27.796 MB/s |
| Random Write   16 MBs |    4.4 s   |   3.521 MB/s |    2.2 s   |   7.259 MB/s |
| Read           40 MBs |    0.1 s   | 643.956 MB/s |    0.1 s   | 526.371 MB/s |
| Random Read    16 MBs |    0.0 s   | 447.503 MB/s |    0.0 s   | 513.052 MB/s |
'-------------------------------------------------------------------------------'
```

Listing 14. *Latency results*

```
,-------------------------------------------------------------------------------------.
| Item         |      iSCSI      |      iSCSI      |    Local Disk   |    Local Disk   |
|              | Average latency | Maximum latency | Average Latency | Maximum Latency |
+--------------+-----------------+-----------------+-----------------+-----------------+
| Write        |     0.076 ms    |    38.357 ms    |     0.068 ms    |    40.944 ms    |
| Random Write |     0.059 ms    |    34.362 ms    |     0.027 ms    |    32.242 ms    |
| Read         |     0.014 ms    |    33.012 ms    |     0.013 ms    |    20.829 ms    |
| Random Read  |     0.010 ms    |    16.731 ms    |     0.006 ms    |     0.212 ms    |
|--------------+-----------------+-----------------+-----------------+-----------------|
| Total        |     0.042 ms    |    38.357 ms    |     0.034 ms    |    40.944 ms    |
'--------------+-----------------+-----------------+-----------------+-----------------'
```

Listing 15. *Results for 8 concurrent I/O threads*

```
,-------------------------------------------------------------------------------.
| Item                   | iSCSI Time |  iSCSI Rate  | Local Disk |  Local Disk  |
|                        |            |              |    Time    |     Rate     |
+------------------------+------------+--------------+------------+--------------+
| Write           80 MBs |   16.2 s   |   4.952 MB/s |    2.8 s   |  28.289 MB/s |
| Random Write    31 MBs |   13.3 s   |   2.347 MB/s |    4.6 s   |   6.782 MB/s |
| Read            80 MBs |    0.1 s   | 776.933 MB/s |    0.1 s   | 792.660 MB/s |
| Random Read     31 MBs |    0.0 s   | 798.498 MB/s |    0.1 s   | 317.801 MB/s |
'-------------------------------------------------------------------------------'
```

Listing 16. *Latency results*

```
,-------------------------------------------------------------------------------------.
| Item         |      iSCSI      |      iSCSI      |    Local Disk   |    Local Disk   |
|              | Average latency | Maximum latency | Average Latency | Maximum Latency |
+--------------+-----------------+-----------------+-----------------+-----------------+
| Write        |     0.142 ms    |    90.817 ms    |     0.120 ms    |   101.526 ms    |
| Random Write |     0.083 ms    |    64.169 ms    |     0.036 ms    |    41.964 ms    |
| Read         |     0.021 ms    |    42.344 ms    |     0.018 ms    |    57.642 ms    |
| Random Read  |     0.013 ms    |    22.091 ms    |     0.022 ms    |    90.336 ms    |
|--------------+-----------------+-----------------+-----------------+-----------------|
| Total        |     0.072 ms    |    90.817 ms    |     0.058 ms    |   101.526 ms    |
'--------------+-----------------+-----------------+-----------------+-----------------'
```

Listing 17. *Results for 16 concurrent I/O threads*

```
,-------------------------------------------------------------------------------.
| Item                   | iSCSI Time |  iSCSI Rate  | Local Disk |  Local Disk  |
|                        |            |              |    Time    |     Rate     |
+------------------------+------------+--------------+------------+--------------+
| Write          160 MBs |   33.1 s   |   4.836 MB/s |    5.7 s   |  27.841 MB/s |
| Random Write    62 MBs |   64.1 s   |   0.974 MB/s |   11.1 s   |   5.621 MB/s |
| Read           160 MBs |    0.2 s   | 792.951 MB/s |    0.2 s   | 803.451 MB/s |
| Random Read     62 MBs |    0.1 s   | 627.032 MB/s |    0.1 s   | 712.722 MB/s |
'-------------------------------------------------------------------------------'
```

Listing 18. *Latency results*

```
,-------------------------------------------------------------------------------------.
| Item          |      iSCSI      |      iSCSI      |   Local Disk    |   Local Disk    |
|               | Average latency | Maximum latency | Average Latency | Maximum Latency |
+---------------+-----------------+-----------------+-----------------+-----------------+
| Write         |    2.201 ms     |   5282.627 ms   |    0.741 ms     |   3248.578 ms   |
| Random Write  |    0.138 ms     |    148.243 ms   |    0.070 ms     |     97.927 ms   |
| Read          |    0.037 ms     |     88.114 ms   |    0.034 ms     |    110.471 ms   |
| Random Read   |    0.014 ms     |     32.601 ms   |    0.016 ms     |     39.086 ms   |
|---------------+-----------------+-----------------+-----------------+-----------------|
| Total         |    0.826 ms     |   5282.627 ms   |    0.291 ms     |   3248.578 ms   |
'---------------+-----------------+-----------------+-----------------+-----------------'
```

Listing 19. *Results for 32 concurrent I/O threads:*

```
,--------------------------------------------------------------------------------.
| Item                   | iSCSI Time |  iSCSI Rate  | Local Disk |  Local Disk  |
|                        |            |              |    Time    |     Rate     |
+------------------------+------------+--------------+------------+--------------+
| Write          320 MBs |   77.4 s   |   4.132 MB/s |   12.2 s   |  26.153 MB/s |
| Random Write   125 MBs |  169.9 s   |   0.736 MB/s |   22.6 s   |   5.542 MB/s |
| Read           320 MBs |    0.3 s   | 930.887 MB/s |    0.4 s   | 800.845 MB/s |
| Random Read    125 MBs |    0.2 s   | 708.745 MB/s |    0.2 s   | 742.664 MB/s |
'--------------------------------------------------------------------------------'
```

Listing 20. *Latency results*

```
,-------------------------------------------------------------------------------------.
| Item          |      iSCSI      |      iSCSI      |   Local Disk    |   Local Disk    |
|               | Average latency | Maximum latency | Average Latency | Maximum Latency |
+---------------+-----------------+-----------------+-----------------+-----------------+
| Write         |   15.472 ms     |  46694.484 ms   |    2.279 ms     |  10767.881 ms   |
| Random Write  |    0.225 ms     |    306.176 ms   |    0.130 ms     |    184.111 ms   |
| Read          |    0.065 ms     |    226.841 ms   |    0.087 ms     |    182.375 ms   |
| Random Read   |    0.029 ms     |    134.505 ms   |    0.030 ms     |    121.424 ms   |
|---------------+-----------------+-----------------+-----------------+-----------------|
| Total         |    5.622 ms     |  46694.484 ms   |    0.873 ms     |  10767.881 ms   |
'---------------+-----------------+-----------------+-----------------+-----------------'
```

the particular test. We can see that in all the tests, the time is increasing when the storage is iSCSI and when the storage is a local hard disk it remains at significantly lower values.

Also we can also notice that the latency of the tests is usually higher in the iSCSI storage and lower in the local storage. This is the effect of the network throughput and performance. So, if we have better network equipment we would probably get better results at this test.

SVETOSLAV CHUKOV

Svetoslav Chukov/Chukich is a system administrator with experience in BSD and Linux. Some of the primary interests for him are: system security, firewalls, improving performance of the servers, filesystem optimizations, benchmarks, high availability and some others... He enjoys benchmarking huge storage servers, or if they aren't available, he also likes to play with „more simple" 2 nodes clusters.

Content Management Made Easy

The Open Source Way!

We take a look at the open-source Content Management Systems available for your enterprise website.

What you will learn…

- Picking the (potentially) best CMS for your needs

What you should know…

- General info about website creation, content management and enterprise resource planning

You need to have a web presence. Indeed, you can't expect to do great things if you don't have a website. And when it comes to the web, the days of boring, static pages are long gone. Nowadays, all that matters is the content and how you manage it (pun unintended). To cut the long story short, you need to choose the perfect *Content Management System* (CMS) for your web presence. This choice is important in the sense that the ultimate look and feel of your website relies almost entirely on the CMS you decide to employ.

And when it comes to choosing the ideal CMS, the power of open source can not be underestimated. Over 95% of the CMSes are open source, and that number says it all! Wordpress, Joomla, Drupal – you name it! No matter what sort of site you are running, open source ensures that you are in safe hands. The obvious benefit of opting for open source CMS solution is the cost reduction and easy customizability, thereby saving time, efforts and funds.

So, how do we choose the best CMS? The good thing is that there are numerous free CMS options available. The first thing to consider is the type of website that you are planning to build. If it is an e-commerce site, your requirements will be different from that of someone running a forum. Once, you've decided on the type or genre of your website, you may get to the available CMS options, along with each other's pros and cons.

What To Bear In Mind When Choosing a CMS?

Flexibility

Simply put, a good CMS should allow you to effectively manage your website with the least possible efforts. A WYISWYG Editor is a must, along with a plethora of plugins. You should be able to customize your website the way you want, not the way your CMS wants.

Nowadays, most CMSes come bundled with plugins and modules that allow you to do virtually anything that you may want.

However, not all plugins are alike, and certain CMSes like Wordpress and Joomla score over the rest in terms of plugins' repository for the simple reason that their developers' community is more active (and perhaps older too, as in case of Joomla).

Adaptability

Anyone running a website knows that it is crucial that time is spent setting up the website, not the CMS. If you are running a collaborative site with many contributors, ease of use of the CMS becomes all the more important. It is here that the adaptability features of the CMS come in handy. Needless to say, it will be a sure disadvantage if you end up *teaching* yourself how to deal with the CMS. Most CMSes offer user friendly interfaces now, though many still lag behind.

Functionality

A good CMS offers way more than just creation of pages. It all comes down to the user to get the most out of it. For example, if you are running a blog with hundreds of posts and just two pages, Wordpress, Textpattern and even Dotclear will serve your purpose, but Pixelpost won't. In case of an enterprise website, Magento will be the obvious choice for e-commerce solutions and not Drupal, simply because Magento is custom-built for such tasks and thus offers better functionality.

Plugins, Addons & Extensions

Most CMSes employ plugins and extensions to manage data and present it. For instance, Wordpress has a world-renowned plugin to combat spam named Akismet. On a similar tone, a newer plugin called Defensio offers the same feature set as Akismet. Though it now has become a matter of choice, most businesses tend to go with Akismet's commercial license rather than Defensio's free service for the sake of reliability. In CMS like Joomla, even themes and templates are managed as addons.

Such plugins/extensions are usually installed using the user interface or by uploading the required files via FTP. Quite obviously, it is a wise choice if you ensure that the CMS you are going to bank upon is backed by the requisite plugins and extensions. This will save you time and efforts as most of the repetitive coding will be taken care of by plugins and addons. Also, just because a CMS has millions of plugins to its credit doesn't essentially denote its worth. In any given repository, half of the projects are abandonware, and same applies to plugins.

Security

No one can underestimate the importance of securing one's online presence. While most CMSes secure the management end (i.e. admin, control panel, etc.), not all of them may do the same with the directory and site folders. For instance, while running my photo blog, I noticed that Pixelpost had left my .htaccess file with public visibility and editable. In that case, you may need to manually mark your crucial files as *read-only*.

Majority of the CMS softwares though now offer extensions that take care of the security front for you, unless you opt for a really obsolete CMS with rare updates.

Update Mechanism

The world of technology is ever changing, and you'd surely not want your enterprise to be stuck with floppy diskettes in the age of USB. Same goes with CMS softwares. While some offer fort-nightly updates (like Wordpress), others take their sweet time in bringing out an update (say, Zikula). Its not just the count of updates, but also the security patches that needs to be taken into account.

What Not To Do When Choosing a CMS?

Going Geek-friendly

This is one of the most important things that you need to bear in mind. In all probability, your website will have more than one contributors, and not all of them will share the same level of technical expertise. A good CMS will cater to all and sundry, and if the CMS isn't idiot-friendly, its time to dump it at the earliest.

The Bigger, the Better

Never judge a book by its cover – replace *book* with CMS. Bluntly put, just because a CMS has zillions of forum threads and promises to give you *gorgeous templates*, it doesn't become the best choice for your website. A wiser choice is to opt for a simpler solution and adhere to it, rather than work with a CMS that makes your coffee too.

Trying them all

A better choice will be to shortlist a few CMSes, check out the demos, and then finally settle with one. This will not only save time but also help you remain focused with no undue interference.

Installing ALL plugins under the sun

Let's face it. The availability of so many extensions and addons is hard to resist. But you need to have a clear idea of the modules or plugins that you want. If it is spam that you are fighting, you should consider Akismet. If you're on

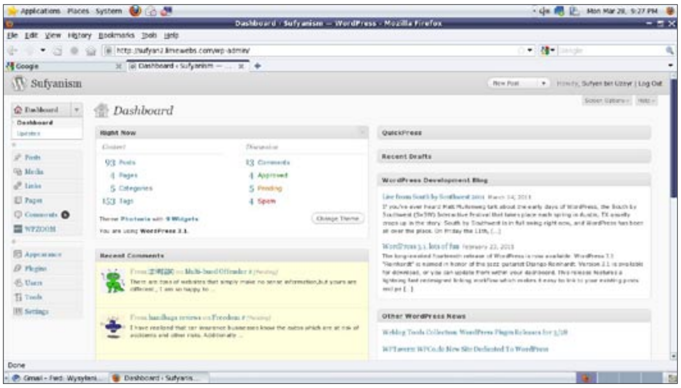


Figure 1. *Wordpress is an easy-to-use yet powerful CMS that is specially-built keeping the bloggers' needs in mind*

a budget, Defensio will be a good choice. Last, you can dump both and stick to the captcha system. The choice is yours, but as you'd notice, you'll have to employ any one, not all.

With the Do's and Dont's out of the way, lets take a quick look at some of the major open source CMS options.

Some CMSes You Just Can't Ignore!

General Purpose

Zikula: Zikula is a Web Application Toolkit, which allows you to run impressive websites and build powerful online applications. It focuses on security and customizability. Though relatively newer, Zikula will definitely be a great bet for running your website owing to the fact that it gives you total control in your hands, not in someone else's.
Pros: Secure, large scope for customization, excellent update mechanism.
Cons: Not very user-friendly

www.zikula.org

Wordpress: The grand-daddy of blogging platforms, Wordpress has now established itself as the *de-facto* standard when it comes to content-rich websites that are easy to maintain.
Pros: Extremely popular, ease of use
Cons: Security issues/bugs though none so grave

Joomla: Joomla is by far one of the most popular and widely used open-source CMS. You can use it to develop almost any genre of website, be it a portal or a blog, or even a full-fledged social arena.
Pros: Highly extendible by means of plugins, huge community support
Cons: WYSIWYG editor needs improvement

www.joomla.org

Geeklog: Initially developed as a blog engine, Geeklog has support for comments, trackbacks, multiple syndication formats, spam protection, and many other features. Needless to emphasise, it gives you all that you can desire to set up your site.
Pros: Gives you total control over the website, in the absolute sense
Cons: May seem overwhelming for a beginner

www.geeklog.net

Drupal

Drupal is another open-source content management system for building dynamic web sites offering a broad range of features and services including user administration, publishing workflow, discussion capabilities, news aggregation, metadata functionalities using controlled vocabularies and XML publishing for content sharing purposes.
Pros: plugins, themes and everything you'll ever need to build your website
Cons: Portability issues (difficult to migrate to other CMS if needed)

www.drupal.org

PHP-Nuke: PHP-Nuke is an automated news publishing and content management system based on PHP and MySQL. The system is fully controlled using a web-based user interface.
Pros: Ideal for news sites
Cons: Not the best choice for websites that don't publish on an hourly basis

www.phpnuke.org

XOOPS: XOOPS is an acronym of eXtensible Object Oriented Portal System. It can serve as a web framework for use by small, medium and large enterprise sites.
Pros: No-nonsense approach
Cons: Still a long way to go before it evolves as a true CMS

www.xoops.org

Typo3: Typo3 is a CMS for enterprise purposes on the web and in intranets. Its USP is the extreme ease of use it lends to multi-user management.
Pros: Flexible and extendable via modules
Cons: Bloated administrative interface

www.typo3.com

Concrete5: Concrete5 is perhaps one of the easiest CMS to use when it comes to developing a website. It is also backed by a huge number of plugins and modules.
Pros: Ease of use
Cons: Poor community support

www.concrete5.org

CMS Made Simple: CMS Made Simple allows you to fully customize your website using a table based layout, or a fully valid XHTML/CSS layout. Though initially targeted at personal and small business websites, CMS Made Simple is now widely employed as an enterprise CMS solution.
Pros: Very easy to use and extend
Cons: Lesser number of plugins and modules

www.cmsmadesimple.org

Forums

So, you're probably planning to run a forum on your website? Though CMSes like Zikula and Concrete5 provide support for forums, it is nevertheless a wiser option to install a separate and dedicated CMS for the purpose, in a directory of its own (it is a good practice to install each CMS in its own directory).

phpBB: phpBB is the most widely used open source bulletin board system in the world. PhpBB has all the features you will need to setup a large bulletin board for a corporate website.
Pros: Ease of use, good support
Cons: Administrative interface needs improvisation

www.phpbb.com

bbPRESS: bbPress is forum software from the creators of WordPress. bbPress concentrates on ease of use and speed. If you're planning a forum for your enterprise site but are short of time (or too lazy to deal with the zillions of features provided by phpBB), bbPRESS may very well be the solution for you.
Pros: Customizability, easy to pick up
Cons: Not as robust feature set as phpBB

www.bbpress.org

e-Commerce

Most enterprise websites may require e-Commerce features. Again, you have the option of employing the standard CMS like Joomla, but owing to the simple fact that open source CMSes are generally free (or available at a negligible price), it is a better option to go for a dedicated solution.

Zencart: Zen Cart is the most popular open-source e-commerce CMS for enterprises. It gives you a plethora of options along with simple interface. Just be prepared to update it manually (the default update mechanism is prone to bugs).
Pros: ease of use, popular
Cons: Poor update mechanism

www.zen-cart.com

osCOMMERCE: osCommerce provides out-of-the-box features to setup e-commerce solutions. It has several extensions and a near-perfect approach to e-Commerce, but it lags behind Zencart when it comes to community support.
Pros: Customizable
Cons: Poor support

www.oscommerce.com

Visit our website
You will find here:
materials for articles-listings, additional documentation, tools
the most interesting articles to download
current information on the upcoming issue
www.bsdmag.org

Magento: Magento is a CMS based on the Zend framework. It is one of the fastest growing e-commerce solutions.
Pros: Rapid progress, versatile, easily extendible
Cons: Relatively newer
www.magentocommerce.com

ERP (Enterprise Resource Planning)

Orange HRM: OrangeHRM is the world's leading open source HRM solution for small and medium sized enterprises (SMEs).
Pros: Universal usage, ease of use
Cons: Difficult to customize (good technical skills needed)
www.orangehrm.com

Front Accounting: FrontAccounting is an accounting system for small companies. Though it may not be able to handle your accounts if you are as big a firm as Microsoft, it suffices pretty well for small and medium businesses.
Pros: Simple, no-nonsense and minimal approach
Cons: Feature set needs improvement
www.frontaccounting.com

Customer Support

We can't forget our customers, can we? The general practice nowadays is to setup a customer support system using a CMS dedicated for the purpose. It not only leaves your main mechanism uncluttered, but also facilitates abstraction.

HESK: Hesk allows you to setup a web based ticket support system for your website. Hesk also provides an easy-to-use web interface to manage customer support requests.
Pros: Uncluttered interface
Cons: Poor multi-lingual support
www.hesk.com

osTICKET: osTicket is another open source support ticket system that integrates inquiries created via email and web-based forms into one web interface. The idea behind osTICKET is simple. It lets your users send mails or fill forms, and compiles the data entered in the manner you deem fit.
Pros: Ease of use
Cons: Poor update mechanism
www.osticket.com

phpSUPPORT: phpSupport allows users to register to open trouble tickets and keep track of their tickets. If you wish to make life easy for the end-users (and difficult for the administrator), phpSUPPORT will serve your cause.
Pros: Speed, ease of use
Cons: Chaotic administrative back-end
http://phpsupport.jynx.net

So, The Bottom Line?

Content Management Systems have become an indispensable element of web development today (unless of course if you are designing absolutely static pages with no flavor). Choosing the perfect CMS isn't the world's easiest job.

Considering the features of a CMS is a good idea indeed, especially those related to security, updates, extendibility, customization, etc. However, even a general glance at the feature sets will show you that almost all of them offer identical set of features (seriously, which CMS doesn't provide RSS Feed support, for example?)

What can be done in such a case? Well, to begin with, the list of *requirements* should be different from that of *wishes*. Keeping a plain, simple yet emphatic list of requirements always means that you can back it up, to suit your business' perspectives. At the same time, it is advisable to bear in mind the changing trends of technology. While you would not want to be overwhelmed by features that may or may not help your cause, you will also not want to be locked down by a CMS that fails to survive the pangs of time. For instance, Wordpress has plugins for every second job on earth – be it displaying songs' lyrics to your logged-in users or helping them discuss your posts. Not all of them will serve your purpose. If yours is not a debate/journal website, having a dedicated plugin for comments isn't extremely vital. In many cases, even the default feedback mechanism will suffice. The idea here is to make a list of addons that your enterprise really requires, and then install them.

SUFYAN BIN UZAYR

Sufyan is a 20-year old freelance writer, graphic artist, programmer and photographer based in India. He writes for several print magazines as well as technology blogs. He is also the Founder and Editor-in-Chief at http://www.bravenewworld.co.nr He can be reached at http://www.sufyan.co.nr

FreeBSD Mall

Your FreeBSD & PC-BSD Resource

www.FreeBSDMall.com

FreeBSD 8.2 Jewel Case CD/DVD

Set contains:

- **Disc 1**: Installation Boot (i386)
- **Disc 2**: LiveFS (i386)
- **Disc 3**: Essential Packages (i386)
- **Disc 4**:Essential Packages (i386)

FreeBSD 8.2 CD	$39.95
FreeBSD 8.2 DVD	$39.95
FreeBSD 7.4 CDROM	$39.95
FreeBSD 7.4 DVD	$39.95

FreeBSD Subscriptions

Save time and $$$ by subscribing to regular updates of FreeBSD!

FreeBSD Subscription , start with CD 8.2	$29.95
FreeBSD Subscription, start with DVD 8.2	$29.95
FreeBSD Subscription, CD 7.4	$29.95
FreeBSD Subscription, DVD 7.4	$29.95

PC-BSD 8.2 DVD (Hubble Edition)

PC-BSD 8.2 DVD	$29.95
PC-BSD Subscription	$19.95

BSD Magazine

BSD Magazine	$11.99

The FreeBSD Handbook

The FreeBSD Handbook, Volume 1 (User Guide)	$39.95
The FreeBSD Handbook, Volume 2 (Admin Guide)	$39.95
★ Special: The FreeBSD Handbook, Volume 2 (Both Volumes)	$59.95
★ Special: The FreeBSD Handbook, Both Volumes, & FreeBSD 8.2	$79.95

The FreeBSD Bundle

Inside the Bundle, you'll find:

- FreeBSD Handbook, 3rd Edition, Users Guide
- FreeBSD Handbook, 3rd Edition, Admin Guide
- FreeBSD 8.2 4-disc set
- FreeBSD Toolkit DVD

★ Special: The FreeBSD CD Bundle	$99.95
★ Special: The FreeBSD DVD Bundle	$99.95

The FreeBSD Toolkit DVD	$39.95
FreeBSD Mousepad	$10.00
FreeBSD Caps	$20.00
PC-BSD Caps	$20.00

For *MORE* FreeBSD & PC-BSD items, visit our website at FreeBSDMall.com!

CALL 925.240.6652 Ask about our software bundles!

Enterprise Open Source Storage: FreeNAS with ZFS has arrived...

✓ Reliable ↑ Scalable $ Affordable

1-855-GREP-4-IX | http://www.iXsystems.com/FreeNAS

iX systems®