

MAGAZINE

BSD

FOR NOVICE AND ADVANCED USERS

PANORAMIC PHOTOGRAPHY IN BSD

INSIDE

2012: BSD YEAR IN REVIEW BY DRU LAVIGNE

CONFIGURING THE QMAIL AUTORESPONDER TO IGNORE SPAM

POSTGRESQL: USERS AND PERMISSIONS MANAGERMENTS

FREEBSD PROGRAMMING PRIMER

VOL.7 NO.1
ISSUE 01/2013(42)
1898-9144



800-820-BSDI
<http://www.ixsystems.com>
Enterprise Servers for Open Source



✓ Increased Performance ✓ Impressive Energy Savings



TrueNAS™

UNIFIED. SCALABLE. FLEXIBLE.



Across all industries the demands of data infrastructure have soared to new heights.

As capacity requirements continue to rise at an ever-increasing rate, performance must not be compromised. The hybrid architecture and advanced software capabilities of the TrueNAS appliance enable users to be more agile, effectively manage the explosion of unstructured data and deploy a centralized information storage infrastructure. Whether it's backing virtual machines, business applications, or web services, there's a TrueNAS appliance suited to the task.

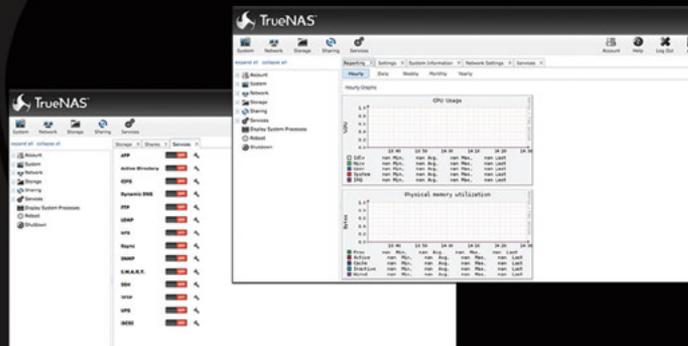
TrueNAS™ Storage Appliances: Harness The Cloud

iXsystems' TrueNAS Appliances offer scalable high-throughput, low latency storage

All TrueNAS Storage Appliances feature the Intel® Xeon® Processors 5600 series, powering the fastest data transfer speeds and lowest latency possible. TrueNAS appliances come in three lines: Performance, Archiver, & High Availability. High-performance, high-capacity ioMemory modules from Fusion-io are available in the TrueNAS Enterprise, Ultimate, and Archiver Pro models.

Key Features:

- One or Two Six-Core Intel® Xeon® Processors 5600 series
- Share Data over CIFS, NFS and iSCSI
- Hybrid storage pool increases performance and decreases energy footprint
- 128-bit ZFS file system with up to triple parity software RAID



*Optional component

	TrueNAS Pro	TrueNAS Enterprise	TrueNAS Ultimate	TrueNAS Fileshare	TrueNAS Archiver Pro	TrueNAS Pro-HA	TrueNAS Enterprise-HA	TrueNAS Ultimate-HA
	PERFORMANCE			ARCHIVER		HIGH AVAILABILITY		
Fusion-io Card		X	X		X			
Deduplication					X			
High Availability						X	X	X
Gigabit NICs	Quad	Dual	Dual	Dual	Dual	Six	Quad	Dual
10 Gigabit NICs		Dual*	Quad*		Dual*			Dual
Max Main Memory	48Gb	96Gb	192Gb	48Gb	192Gb	48Gb	96Gb	192Gb
Max Capacity	220TB	500TB	1.5PB	580TB	2.2PB	250TB	310TB	1.4PB
Rack Units	2U	2U/4U	4U	2U	4U	3U	3U	Dual 3U



Call iXsystems toll free or visit our website today!
1-855-GREP-4-IX | www.iXsystems.com



Intel, the Intel logo, and Xeon Inside are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

MAGAZINE **BSD**

Dear Readers,

The January issue opens another year with BSD Magazine. We hope you enjoyed the last few.

We start with a second part of our new light-hearted and non-technical column. Its aim is not to learn but to entertain and to inspire some reflection and (maybe) action.

In Developers Corner Dru Lavigne reviewed the 2012 of BSD. Read it to recall the most important events, publications and releases of the last year.

This time the flag article is about panoramic photography. You will have an opportunity to try it out with tools available under a BSD or any Unix like system, using only free open source software.

In the 'How To' section you have a chance to read another part of PostgreSQL series about users and permissions managements.

From the tutorials you will also learn how to ignore spam in 2013, since we get it more and more with each year.

Especially for administrators we start a new series, where Rob Somerville will look at the tools, processes and methods involved in writing software, including developing a Content Management System (CMS) which will run under an AMP stack on FreeBSD, OpenBSD and Linux.

We wish you a good read!

Post Scriptum

Many thanks to iXsystems Company and its employees for their support and contribution. Congratulations to authors, betatesters and proofreaders for their good job and the time they dedicate to the magazine. Finally, the big THANK YOU to all our readers for their passion to knowledge and interest in the subject. On our side we promise to put more effort into improving the magazine content to make it worth your attention and time.

Patrycja Przybyłowicz
Editor of BSD Magazine
& BSD Team

Editor in Chief:

Ewa Dudzic
ewa.dudzic@software.com.pl

Supportive Editor

Patrycja Przybyłowicz
patrycja.przybylowicz@software.com.pl

Contributing:

Dru Lavigne, Rob Somerville, Luca Ferrari, William Olson,
Carlos E. G. Carvalho

Top Betatesters & Proofreaders:

Barry Grumbine, Darren Pilgrim, Eric Geissinger,
Henrik Andersén, Imad Soltani, Luca Ferrari, Luiz Claudio
Pacheco, Mani Kanth, Rob Cabrera, Zander Hill

Special Thanks:

Denise Ebory

Art Director:

Ireneusz Pogroszewski

DTP:

Ireneusz Pogroszewski
ireneusz.pogroszewski@software.com.pl

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:

Ewa Dudzic
ewa.dudzic@software.com.pl

Advertising Sales:

Patrycja Przybyłowicz
patrycja.przybylowicz@software.com.pl

Publisher :

Software Media 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.

All trade marks presented in the magazine were used only for informative purposes. All rights to trade marks presented in the magazine are reserved by the companies which own them.

Mathematical formulas created by Design Science
MathType™.

Let's Talk

06 Keyboard Error... Press any Key to Continue

By Rob Somerville

Some pieces of technology are extra special, some utilitarian. Until the user interface develops to the point that we can get strong tactile feedback from a touch screen (via electrostatic shock perhaps?). The IBM Model M keyboard remains the standard by which the computer keyboard – indeed any input device – is defined. Even for the worst two finger typist on the Internet, that reassuring spring-back “click” let you know when a key had registered. Clipping keys on the way past with the edge of fingernails (that should have been trimmed long ago – like the almost mandatory Unix beard) was impossible. The feel of the key responding to the fingertip, the audible “clack” on a successful key-press, all added to the pleasure of writing code, letters, or deleting file-systems as appropriate...

Developers Corner

08 2012: BSD Year in Review

By Dru Lavigne

This article looks back on 2012 and provides an overview of what occurred in the BSD world. Read it to recall the last year's top news for FreeBSD, NetBSD, OpenBSD, DragonFlyBSD and PC-BSD. Check what projects were supported by FreeBSD and NetBSD Foundations. Learn the statistics for BSD Certification. Find out what were the best published books on BSD in 2012 and the editorial plans for 2013.

How To

12 Configuring the Qmail Autoresponder to Ignore Spam

By William Olson

This is one way to combat spam using autoresponder created by Ken Coar with qmail. With this method, when your users setup a vacation message, the autoresponder program will recognize and not reply to spam messages but will reply to messages that pass the spam filter. These instructions require your system to be setup using author's qmail guide on <http://freebsdocks.net>, or have qmail installed with maildrop, validrcptto with on-change and have per-user spam filtering enabled.



16 PostgreSQL: Users and Permissions Managements

By Luca Ferrari

This article considers PostgreSQL user and privilege management. PostgreSQL allows the definition of database users and groups, providing them permissions to access database objects (e.g., tables) and thereby allowing Database Administrators (DBAs) to enforce a security policy on data access. All the examples shown here have been tested on a PostgreSQL 9.1 cluster running on a FreeBSD 8.2-RELEASE machine; all the example source code are available in a GitHub repository.

26 Immersive 360x180° Panoramic Photography in BSD

By Carlos E. G., “Cartola” Carvalho

In this article the author is going to show you what 360x180° panoramic photography is along with a few of its possible applications. His aim is to prove you how fascinating this subject might be for BSD users. Panoramic photography is one of the most complete ways of experiencing a location without physically visiting. That is one of the reasons why it is also called “Immersive Photography”. The author hopes to inspire you to try it out with tools available under a BSD or any Unix like system, using only free open source software.

Admin

46 FreeBSD Programming Primer

By Rob Somerville

In this new series we will look at the tools, processes and methods involved in writing software, including developing a Content Management System (CMS) which will run under an AMP stack on FreeBSD, OpenBSD, Linux etc.

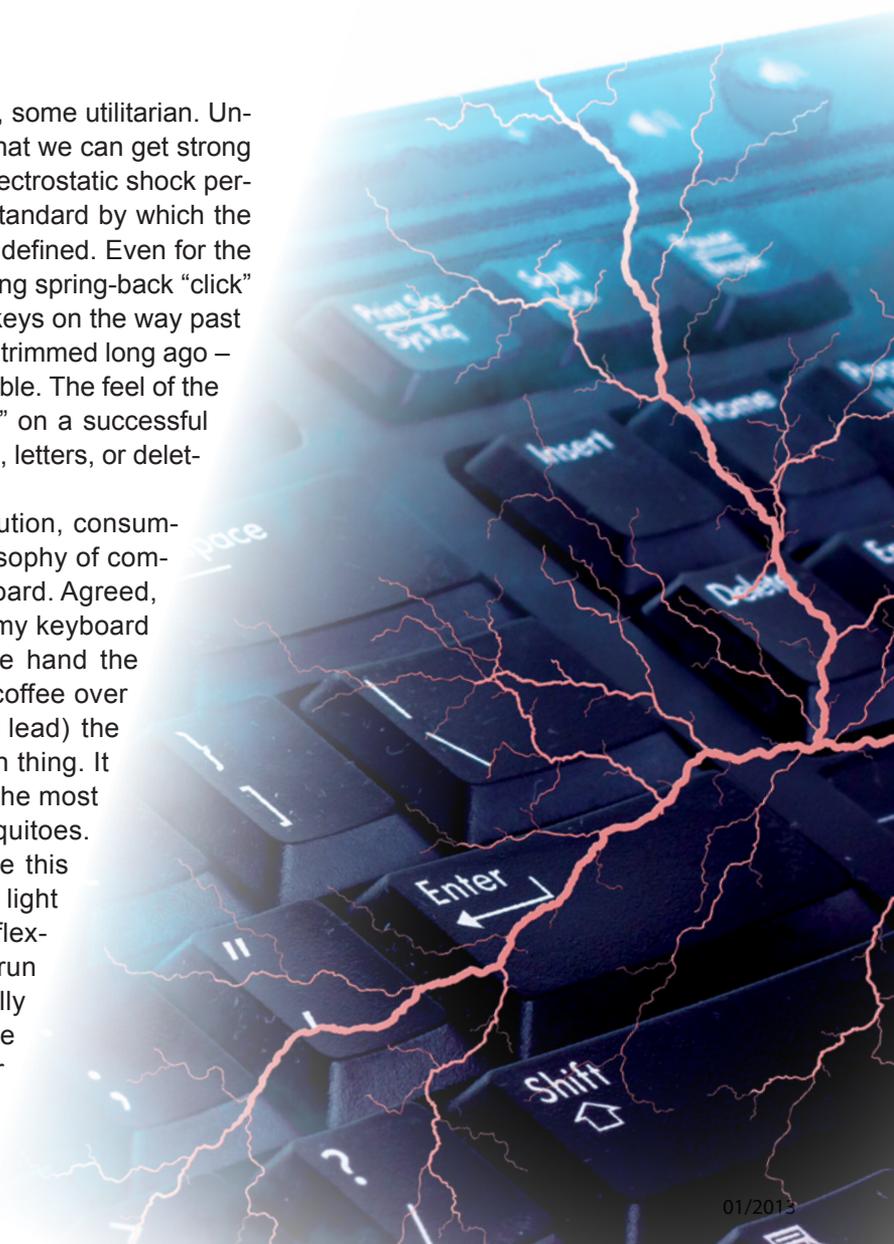
Keyboard Error... Press any Key to Continue

“The test of the machine is the satisfaction it gives you. There isn't any other test. If the machine produces tranquility it's right. If it disturbs you it's wrong until either the machine or your mind is changed”.

Robert M. Pirsig, Zen and the Art of Motorcycle Maintenance: An Inquiry Into Values

Some pieces of technology are extra special, some utilitarian. Until the user interface develops to the point that we can get strong tactile feedback from a touch screen (via electrostatic shock perhaps?), the IBM Model M keyboard remains the standard by which the computer keyboard – indeed any input device – is defined. Even for the worst two finger typist on the Internet, that reassuring spring-back “click” let you know when a key had registered. Clipping keys on the way past with the edge of fingernails (that should have been trimmed long ago – like the almost mandatory Unix beard) was impossible. The feel of the key responding to the fingertip, the audible “clack” on a successful key-press, all added to the pleasure of writing code, letters, or deleting file-systems as appropriate.

Alas, due to the intrusion of technological evolution, consumerism, hot-desking, common space and the philosophy of communitarian-ism, I have a corporate supplied keyboard. Agreed, rank has its privileges (rank = works in I.T.), and my keyboard is wireless. Yet for all the benefits of being able hand the keyboard over to co-workers without spilling my coffee over the desk (after physical contact with the trailing lead) the functionality is eclipsed by the quality of the damn thing. It rattles, squeaks, twangs and generates some of the most interesting audible entertainment known to mosquitoes. I'm sure dogs can hear it at 1000 paces. Despite this great leap forward in technology – it doesn't even light up. Not even a humble Numlock LED graces its flexible extruded plastic curves. It does however, run on batteries, and unless I keep a spare set of fully charged AAA's handy (as the manufacturers have chosen not to provide power-status software for my O/S of choice or via LED), I potentially run



the risk of bringing ridicule to the Free Software movement by giving the appearance that my Desktop has hung when in fact the keyboard batteries are no more. As to the keys themselves, I am sure that the condensation from my breath alone could dissolve the low quality paint printed on each key. At least with the IBM offering – if the color did wear away on the letters, the engraving could be filled in with indelible marker pen or the key cap swapped out. To add insult to injury, the key caps are an integral part of the switch, so cannot be replaced from a spare keyboard. A quick survey around the office demonstrated that I was not the only victim, the number of keyboards with a blank key where the “E” would be (the most common letter used in the English alphabet) was quite surprising.

As to the feedback, the designer of this keyboard obviously took his cues from fluid dynamics rather than mechanical engineering. There is no click, rather a disturbing feeling beneath the finger when a key is pressed, not unlike the sensation when depressing tepid Blu-Tack or bursting a particularly feeble carbuncle on bubble wrap. Quite often, I can press a key and not realize that I pressed another *en passant* – the favorite being catching the Caps Lock key when targeting Tab. This could be forgiven if I had fingers like sausages, but standing sideways I would disappear so I cannot blame genetics. I suspect the real motive cutting down the sound a proper keyboard makes is to reduce the cacophonous noise if large numbers were employed in an open plan office. Just think – Simon Travaglia (The renown author of the BOFH files) would struggle – no >clickety-click< rather a pathetic >squidgedy-plop<.

All of this has a serious side however. We are continually encouraged to re-use, renew, recycle but what chance do we have when a device is designed to be obsolescent and disposable? So often in our consumer driven world the curse of mass-production and poor quality design comes back to haunt us. Just-In-Time systems discourage manufacturers and suppliers from holding spares, and the corporate mentality that profit must be made on everything has spread to the point that original replacement parts have such a mark-up that it is often not cost effective to repair or recondition a broken device (especially if other expensive parts may fail later). This extends throughout the consumer and professional market from mobile phones to large goods and beyond. Even the few true innovators in the IT market lost the original vision. The IBM XT was designed to be modular, and indeed the original ISA bus design was so successful that it was copied by so many manufacturers that it is inconceivable to believe that the later generation IBM PC's used the proprietary Micro Channel architecture purely for performance

rather than intellectual property or profit reasons. As for Apple, replacing the cracked screen on one of the new devices really is a return to manufacturer repair – you need a heat gun to melt the glue holding the device together.

Brand loyalty is important to manufacturers, and one of the the principles of good design is to involve the user with the product to a such a degree that effectively a product is characterized and even humanized. Think Pilsbury Dough Boy. Better still, like a craftsman's used saw, a tool adapts over time and becomes personal to our ways of working. Yet there is a disconnect here – we have reached the point where objects too durable cannot compete in the marketplace as price and profit is everything. So while the marketeers engage in hyping and personalizing our experience, true long term value from an engineering perspective is eroded and when the product breaks, we just buy another.

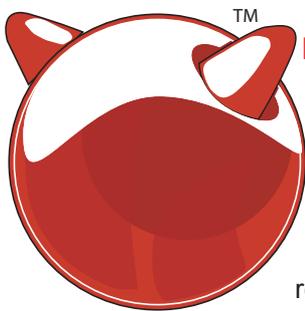
So how as designers, developers and architects can we continue to push the boundaries so that our creations are not mundane? Is that technical solution just another “band aid” and once again we have accepted compromise rather than striving for excellence? Are we willing to truly think outside the box and innovate rather than just accept the average? Do we have the freedom to ask awkward questions and challenge stale thinking? That's why the Open Source ethic is such a threat to the status quo – be it *BSD, Linux or the Internet itself. Once adopted and understood, it becomes irreplaceable, humanized, a part of us. Some would say even addictive. We then quickly see through the commercial realities, and start to appreciate true value, design and quality.

ROB SOMERVILLE

Rob Somerville has been passionate about technology since his early teens. A keen advocate of open systems since the mid eighties, he has worked in many corporate sectors including finance, automotive, airlines, government and media in a variety of roles from technical support, system administrator, developer, systems integrator and IT manager. He has moved on from CP/M and nixie tubes but keeps a soldering iron handy just in case.

2012: BSD Year in Review

This article looks back on 2012 and provides an overview of what occurred in the BSD world.



FreeBSD

The FreeBSD Project had three releases in 2012:

9.0

Was released on January 6. Some of the highlights of this release include: a new installer, SUJ (soft-updates journaling),

ZFS v2.8 which adds RAIDZ3 and deduplication, the HAST (highly available storage) framework, the Capsicum framework for sandboxing applications with minimal code changes, user-level DTrace, pluggable TCP congestion control framework with five new algorithms, NFSv4, high-performance SSH, LLVM and clang, and support for the Sony Playstation 3.

8.3

Was released on April 11. This is the fourth release from the 8-STABLE branch which improves on the functionality of FreeBSD 8.2. It introduced some 9.0 features to this branch (ZFSv2.8 and the pluggable TCP congestion control framework), provided updated drivers, and addressed several security advisories.

9.1

Was released on December 30. Some of its new features include: Intel GPU driver with GEM/KMS support, netmap(4) fast userspace packet I/O framework, POSIX2008 extended locale support, kernel support for the AVX FPU extension, and numerous improvements in IPv6 hardware offload support. Due to the November intrusion on the FreeBSD package building cluster, the FreeBSD Project's package building infrastructure is un-

dergoing a complete review and redesign. This means that the only packages available for 9.1 are those contained on the installation DVD. The Project will make an announcement once the rest of the packages become available.

The FreeBSD ports collection now contain 24,047 ports.

The FreeBSD Project held DevSummits at the following 2012 events: AsiaBSDCon in Tokyo, BSDCan in Ottawa, The University of Cambridge in August, EuroBSDCon in Warsaw, and MeetBSD California in Sunnyvale. The documentation team organized several IRC doc sprints as well as Documentation Working Groups during BSDCan, the Cambridge DevSummit, OpenHelp in Cincinnati, and EuroBSDCon.



NetBSD

The NetBSD Project also had three releases in 2012:

5.1.2

Was released on February 2.

This release was the second critical security update of the NetBSD 5.1 release branch and represented a subset of fixes deemed critical for security or stability reasons.

6.0.1

Was released on October 17 and is considered to be the current release. It is the first security bugfix update of the NetBSD 6.0 release branch and represents the subset of fixes deemed important for security or stability reasons.

5.2

Was released on December 3. This release provides bug and security fixes as well as updated drivers.

pkgsrc currently contains over 13,000 packages. The pkgsrc team released pkgsrc-2012Q1 in April, pkgsrc-2012Q2 in July, and pkgsrc-2012Q3 in October.



OpenBSD

Right on schedule, the OpenBSD Project had two releases in 2012. Each release includes its own artwork and theme song.

5.1

Was released on May 1. This release included: improved hardware support, network stack improvements, routing and PF improvements, improved locale support, OpenSSH 6.0, major performance and stability improvements in the package build process, and improved man pages and documentation.

5.2

Was released on November 1. The most significant change in this release is the replacement of the user-level utthreads by kernel-level rthreads, allowing multithreaded programs to utilize multiple CPUs/cores. Other features include the addition of nginx(8), and OpenSSH 6.1 which turns on pre-auth sandboxing sshd by default for new installs.

The OpenBSD ports collection now contains over 7,600 ports. The OpenBSD developers organized several hackathons in 2012: an rthreads hackathon in Paris, a general hackathon in Budapest, a network hackathon in Starnberg, Germany, a ports hackathon in Budapest, and a hackathon in Coimbra, Portugal.



DragonFly BSD

DragonFly BSD

The DragonFly BSD Project had several point releases and two release branches in 2012:

3.0.1

Was released on February 22. The giant kernel lock was removed from much of the system, increasing performance significantly on multi-core systems. Other features include: significant improvement in HAMMER throughput under heavy storage loads due to a new time domain multiplexing method, and a BSD-licensed tclplay(8) tool for creating and managing encrypted disk volumes which are 100% TrueCrypt compatible.

3.0.2

Was released on March 26.

3.0.2

Was released on August 29.

3.1.1

Was released on November 2.

3.2.2

Was released on December 16. This release added performance improvements under database load, using postgres benchmarking as a measure. FreeBSD's USB stack is available, but is not built by default.

Work on HAMMER2 continues, though it is not yet ready for production use. An update on its features and what still needs to be done can be found at <http://lists.dragonflybsd.org/pipermail/users/2012-December/017716.html>.



PC-BSD

The PC-BSD project had two releases in 2012:

9.0

Was released on January 10. This release added support for multiple window managers, a new Control Panel so that common administrative graphical tools are available from any desktop, support for freebsd-update using a graphical Update Manager, an improved graphical networking utility, an enhanced Life Preserver utility for scheduling backups, support for BootCamp partitions, and pre-installed virtualbox and vmware images.

9.1

Was released on December 18. This release added the following features: a newly designed installer to separate pre- and post-installation configuration tasks, many ZFS improvements in the installer, a newly designed Warden for managing FreeBSD, ports, and Linux jails, multiple boot environment support for ZFS, the ability to install a vanilla FreeBSD or TrueOS server, a new About utility, a new GDM configuration utility, a new sound configuration GUI, a new hardware compatibility GUI, a new Bluetooth pairing management utility, a new mount tray utility, an improved EasyPBI utility for converting FreeBSD ports to PBIs, a newly designed thinclient script for creating PXE boot servers or PXE installation servers, and USB live mode is now read-writable.

AppCafe now contains over 1,050 PBIs. pbidir.com has been deprecated and a new FreshPBIs.org website is being created to make it easy for users to track PBI information.



FreeBSD Foundation

The FreeBSD Foundation is a 501(c)(3), US based, non-profit organization dedicated to supporting and building the FreeBSD Project and community worldwide. The Foundation accepts donations, using them to fund and manage projects, sponsor FreeBSD events, and provide travel grants to FreeBSD developers and contributors.

The Foundation sponsored the following conferences in 2012: AsiaBSDCon, BSDCan, including its Developer Summit and Vendor Summit, BSDDay in Vienna, EuroBSDCon, the Cambridge Developer Summit, and the Bay Area Vendor Summit.

The Foundation sponsored the travel of twelve people to BSDCan, four to EuroBSDCon, one to OpenHelp, and one to MeetBSD California. Trip reports describing the benefits of sponsorship can be found at freebsdfoundation.blogspot.com.

The Foundation provided financial grants to the following development projects in 2012:

- **iSCSI Target project:** the goal of this project is to create a native, high performance, iSCSI target facility for FreeBSD. While configuration and connection setup and teardown will still be handled by a userland daemon, all data movement will be performed in the kernel. The iSCSI target will be fully integrated with the CAM target layer meaning that volumes can be backed by files or any block device. The hardware offload capabilities of modern network adapters will also be supported.
- **Capsicum Component Framework:** this project will develop a comprehensive userspace framework for writing Capsicum-based applications. It will include a Capsicum runtime linker and component library providing sandboxed versions of key higher-level system libraries.
- **Growing Filesystems Online:** this project adds GEOM and filesystem changes that are necessary to increase the size of both UFS and ZFS filesystems while the filesystem is mounted read-write. This project will provide the additional benefit of online provisioning of virtual instances.

- **IPv6 Performance Analysis:** this project will carry out a detailed performance analysis, benchmarking IPv6 to IPv4. It will identify the origins of differences in performance, and where possible, directly address them or identify areas of future work. Having initial benchmark numbers will allow changes to be evaluated by re-running the measurements and quantifying the improvements.
- **Implementing auditdstd:** the goal of this project is to securely and reliably distribute audit records over the TCP/IP network from a local auditdstd daemon to a remote auditdstd daemon. In case of source system compromise, the attacker's activity can be analysed using data collected by the remote system, as only the remote system's audit logs can still be trusted.
- **NAND Flash Support:** this project will enable FreeBSD to natively manage NAND flash devices, satisfying a crucial requirement for many applications needing access to fast, reliable, non-volatile storage.
- **Porting FreeBSD to Efika ARM platform:** this project will make it possible to run X11 applications on FreeBSD on the Efika MX SmartBook laptop and Smart-Top nettop devices, with full support for sound and networking. It will also make it much easier to support other devices, such as some Android tablets, that ship with the i.MX515 SoC.

The Foundation's fund-raising goal for 2012 was \$500,000. Thanks in part to a slashdot post, donations from companies and users of FreeBSD blew away that goal. As of this writing, the donations are still being logged, but it is expected that the final total will be over \$700,000.

NetBSD Foundation

The NetBSD Foundation is a registered 501(c)(3) non-profit organization which serves as the legal entity which owns several of the NetBSD Project servers, handles donations of money, services, hardware or time to the project, and administers NetBSD copyrights.

Donations to the Foundation allow the NetBSD Project to make major improvements to the code base. The NetBSD 6.0 Fund Drive is targeting \$60,000 and will be used to fund development in various areas of the system, including:

- Improving network stack concurrency and performance.
- Development of modern file systems and improvement of existing ones.

- Features which are useful in embedded environments, for example, high resolution timers and execute in place (XIP) support.
- Automatic testing and quality assurance.

As of this writing, the Foundation still needs help in meeting this goal. See www.netbsd.org/donations/ if you would like to assist the NetBSD Project financially.

bsdevents.org

bsdevents.org launched this year. This website provides the following features:

- Blog: announces CFPs, upcoming conferences, presentations, and user group meetings about a week in advance. Blog posts are also tweeted to [@bsdevents@](https://twitter.com/bsdevents).
- Calendar: lists upcoming conferences with BSD related presentations and/or booths.
- Presentations: known BSD presentations are organized by year and event. Uses icons to indicate at a glance the type of presentation media.
- User groups: list of known BUGs (or UUGs) categorized by country, state/province, and city.

The website contains a submission form if you have information about a missing user group or presentation or if you would like to submit the details for an upcoming presentation or BSD booth. A subscribe button is provided for receiving updates in various formats.



BSD Certification

The BSD Certification Group organized 27 paper-based exam events in 14 countries in 2012. Computer-based exams are also available in 20 countries throughout the world. There are currently 187 certified BSDAs in 25 countries.

The Group released the details for the BSDA re-certification policy: <http://www.bsdcertification.org/certification/certification/recertification-policy>. Certificants who expire in 2013 will also receive an email in January reminding them of the re-certification policy.

The latest release of the BSDA Study DVD was released on December 10. Even if you are not planning on taking the BSDA exam, the DVD provides a handy installation media and documentation reference for FreeBSD,

NetBSD, OpenBSD, and DragonFly BSD. DVD sales are used to fund the ongoing psychometric work required by the BSDA certification.

Books

The following books were published in 2012:

- FreeBSD Device Drivers by Joseph Kong: nostarch.com/bsd/drivers.htm.
- SSH Mastery: OpenSSH, PuTTY, Tunnels and Keys by Michael W. Lucas: www.amazon.com/SSH-Mastery-OpenSSH-PuTTY-Tunnels/dp/1470069717.
- The Kindle version of the FreeNAS 8.3.0 Users Guide is available for purchase. This Guide is also freely available in several formats at doc.freenas.org.
- The PC-BSD 9.0 Users Handbook is available in book format from FreeBSD Mall as well as in Kindle format. The Handbook is also freely available in several formats at wiki.pcbsd.org.

The following books are in the pipeline for 2013:

- Absolute OpenBSD, 2nd Edition by Michael W. Lucas. This book will become available in April and pre-orders are available now from nostarch.com/openbsd2e.
- The FreeBSD documentation team is working hard to update the FreeBSD Handbook so that an up-to-date version can be published in book format. The current goal is to have a publishable edition by BSDCan 2013 with the book format available from the FreeBSD Mall. An epub format will also be available, with sales going directly to the FreeBSD Foundation.
- The Best of FreeBSD Basics by Dru Lavigne is currently being re-written for FreeBSD 9.x. The new edition should be available by end of 2013.

DRU LAVIGNE

Dru Lavigne 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.

Configuring the Qmail Autoresponder to Ignore Spam

This is one way to combat spam using autoresponder created by Ken Coar with qmail. With this method, when your users setup a vacation message, the autorespond program will recognize and not reply to spam messages but will reply to messages that pass the spam filter.

What you will learn...

- How to install autorespond from source and the Mime-Tools from ports
- How to create the autorespond file
- How to fix vpopmail permissions

What you should know...

- An understanding of the FreeBSD ports system
- FreeBSD 8.x or 9.x is recommended. I386 or AMD64
- How qmail works with John Simpsons latest qmail patch
- How apache works (Apache 2.2 or better with SSL encryption is recommended)

These instructions require your system to be setup using my qmail guide on <http://freebsdocks.net>, or have qmail installed with maildrop, validrcptto with onchange and have per-user spam filtering enabled.

For starters we need to install the dependencies for the autoreponder source:

```
# cd /usr/ports/mail/p5-MIME-Tools
# make install
```

What is qmail?

qmail is a mail transfer agent (MTA) that runs on Unix. It was written, starting December 1995, by Daniel J. Bernstein as a more secure replacement for the popular Sendmail program. qmail's source code is in the public domain, making qmail free software.[2]

From <http://en.wikipedia.org/wiki/Qmail>

Now we install autorespond via the source since there is no FreeBSD port: Listing 1.

Provided you don't get any errors at the end you should be fine Listing 2.

Now run 'make install'.

```
# make install
```

When it's done you should see:

```
Installing applications...
Installing documentation...
autoresponder 1.16.7 installed.
```

Next we need to create the `/home/vpopmail/bin/autoresponder` file using your favorite text editor. I'm using vi:

Listing 1. Extracting and running configure on the autoresponder source

```
# cd ~root
# fetch http://meepzor.com/packages/autoresponder/autoresponder-1.16.7.tar.gz
# tar autoresponder-1.16.7.tar.gz
# cd autoresponder-1.16.7
# ./configure
```

```
# cd ~vpopmail/bin
# vi autorespond
```

Insert the code shown in Listing 3 into the file. This code is a script for the autorespond call so vpopmail will be able to determine whether this is spam. If the headers in the email equal X-Spam-Status=yes then don't send a autorepond message and continue. Otherwise send a reply to the sender.

The code from Listing 4 runs a script called vfixpermissions that checks all permissions in the ~vpopmail folder. These are commands that are run at the console.

Next we recompile qmailadmin to call the autorespond program:

```
# cd /usr/ports/mail/qmailadmin
# make deinstall
# make clean distclean
# make extract
```

This will reinstall qmailadmin so the global autorespond program will be called when new users are created. These are commands that are run at the console.

If you get a configure screen when you run make extract, make sure the following options are checked:

```
CATCHALL
MODIFY_QUOTA
NOCACHE
SPAM_DETECTION
TRIVIAL_PASSWORD
USER_INDEX
```

When you run the patch you will get the following output:

```
Installing applications...
Installing documentation...
autoresponder 1.16.7 installed.
```

Listing 2. Configure output

```
loading cache ./config.cache
checking package version... 1.16.7
checking for a BSD compatible install... (cached) /usr/
    bin/install -c
checking for perl... (cached) /usr/bin/perl
checking for sendmail... (cached) /var/qmail/bin/
    sendmail

no
checking for perl... (cached) /usr/bin/perl
checking Perl module POSIX... found
checking Perl module MIME::Parser... found
checking Perl module Getopt::Long... found
checking Perl module MIME::Base64... found
checking Perl module Fcntl... found
creating ./config.status
creating autoresponder
Creating Makefile
```

Autoresponder has been configured with the following options.

```
    User binaries: /usr/local/bin
    Man pages: /usr/local/man
    sendmail: /var/qmail/bin/sendmail
Perl interpreter location: /usr/bin/perl
Files in distribution: MANIFEST
    INSTALL
    README
    LICENCE.txt
```

```
ChangeLog
TODO
acinclude.m4
configure
configure.in
Makefile.in
install-sh
autoresponder.in
autoresponder.1
```

Listing 3. Autorespond file

```
#!/bin/sh
/usr/local/bin/autorespond \
    --from=$USER@$DOMAIN \
    --ignore-from="MAILER-DAEMON.*" \
    --ignore-header="X-Spam-Status:^yes" \
    --subject="On Vacation (was: %s)" \
    --file=./vacation/message
```

Listing 4. Vpopmail permissions check

```
# cd /var/qmail/bin
# fetch http://qmail.jmsl.net/scripts/vfixpermissions
# chmod 755 vfixpermissions
# ./vfixpermissions
# cd ~vpopmail/bin
# chmod 6711 vchkpw
# chmod 0755 autorespond
# chown vpopmail:vchkpw vchkpw
```

Just look through the output in Listing 6 and make sure that the lines that start with "Hunk" say succeeded and NOT failed. Then proceed to make `qmailadmin` with the patches applied:

NOTE

The sections highlighted in blue will need to be changed to the actual path for `cgi-bin`, your `www` and `images` folders.

Listing 5. Patching `qmailadmin`

```
# cd work/qmailadmin-1.2.15/
# fetch http://qmail.jms1.net/vpopmail/qmailadmin-1.2.12-onchange.3.patch
# patch < qmailadmin-1.2.12-onchange.3.patch
```

Listing 6. Patch output

Hmm... Looks like a unified diff to me...

The text leading up to this was:

```
-----
|diff -ruN qmailadmin-1.2.12-factory/autorespond.c qmailadmin-1.2.12-patched/autorespond.c
|--- qmailadmin-1.2.12-factory/autorespond.c      2007-09-21 19:27:38.000000000 -0400
|+++ qmailadmin-1.2.12-patched/autorespond.c      2008-12-21 15:55:12.000000000 -0500
-----
```

Patching file `autorespond.c` using Plan A...

Hunk #1 succeeded at 186.

Hmm... The next patch looks like a unified diff to me...

The text leading up to this was:

```
-----
|diff -ruN qmailadmin-1.2.12-factory/command.c qmailadmin-1.2.12-patched/command.c
|--- qmailadmin-1.2.12-factory/command.c          2007-01-01 18:17:06.000000000 -0500
|+++ qmailadmin-1.2.12-patched/command.c          2008-12-21 15:26:17.000000000 -0500
-----
```

Patching file `command.c` using Plan A...

Hunk #1 succeeded at 309 (offset 3 lines).

Hmm... The next patch looks like a unified diff to me...

The text leading up to this was:

```
-----
|diff -ruN qmailadmin-1.2.12-factory/user.c qmailadmin-1.2.12-patched/user.c
|--- qmailadmin-1.2.12-factory/user.c             2007-11-03 13:44:12.000000000 -0400
|+++ qmailadmin-1.2.12-patched/user.c             2008-12-21 15:27:57.000000000 -0500
-----
```

Patching file `user.c` using Plan A...

Hunk #1 succeeded at 626 (offset 15 lines).

done

Listing 7. Configuring `qmailadmin`

```
# cd /usr/ports/mail/qmailadmin
# make CONFIGURE_ARGS="--enable-modify-spam=Y --enable-spam-command='|preline -f /usr/local/bin/mailedrop mailfilter'
--enable-cgibindir=/usr/local/www/apache22/cgi-bin --enable-htmldir=/usr/local/www/apache/www
--enable-imagedir=/usr/local/www/apache22/images --enable-help --enable-autoresponder-path=/usr/
home/vpopmail/bin --enable-modify-quota"
# make install clean
```



References

- <http://lifewithqmail.org>
- <http://goodcleanemail.com>
- <http://freebsdroids.net>

Now it's time to test. Log into qmailadmin and setup a vacation response message, then email that user to ensure you get the response.

To test a SPAM filter, create and send an email to your test user containing the following content:

```
XJS*C4JDBQADN1.NSBN3*2IDNEN*GTUBE-STANDARD-ANTI-UBE-TEST-  
EMAIL*C.34X
```

NOTE

This will give your email a spam score of 1000. If your qmail-scanner is set to delete spam above a certain score you will need to check your qmail-scanner logs to ensure it was received and deleted. In any case you should not receive the vacation message.

Conclusion

When this walkthrough is completed, your qmail system will be setup so when your customers/clients setup a vacation autoresponder they will no longer get bounces from unknown email addresses. When a message comes in and a vacation autoresponder is enabled, autorespond checks the header for the X-Spam status and one of two things happen:

- If X-Spam Status=yes then autorespond passes the message but does not send a vacation reply to the sender.
- If X-Spam Status=no then autorespond passes the message and does send a vacation message to the sender

If the user does not have the vacation message enabled then nothing happens. The message will pass through qmail normally.

WILLIAM OLSON

William Olson has been working with qmail since 1998. He is fluent in Windows and FreeBSD hardware and software and is currently an IT Admin for a distributor in Maine.

**BSD development
and consultancy**

Zabbix Monitoring

**Bacula enterprise
backup**

BSD Thin Client

**Corporate BSD
Desktop**

**Solution
management
with Puppet**

and more ...

www.mtier.org
contact@mtier.org

PostgreSQL

Users and Permissions Management

This article considers PostgreSQL user and privilege management. PostgreSQL allows the definition of database users and groups, providing them permissions to access database objects (like tables) and thereby allowing Database Administrators (DBAs) to enforce a security policy on data access.

What you will learn...

- How PostgreSQL organizes database users, groups and permissions
- the concept of roles and which privileges are available on database objects
- the commands to grant and revoke privileges

What you should know...

- basic shell commands
- basic PostgreSQL concepts
- database object creation (creating tables, indexes, and so on)

All the examples shown here have been tested on a PostgreSQL 9.1 cluster running on a FreeBSD 8.2-RELEASE machine; all the example source code are available in a GitHub repository.

The Concept of Role

In older versions of PostgreSQL the access methods were based on the concept of *users* and *groups*. With version 8 the concept of users and groups was deprecated and replaced by the concept of a *role*. This does not mean that PostgreSQL is no longer able to handle user and group credentials and privileges: in fact the concept of role is used by PostgreSQL to map both.

A role can be thought of as a set of credentials and permissions that can be used for a single login (as a user) or that can be shared among other roles, thus functioning as a user group.

When the cluster is initialized via `initdb` a special role is created (on FreeBSD it is the `pgsql`): this is a super-user role (i.e., it has all the privileges) and is used to create other roles and do the “bootstrap” maintenance of a new cluster. PostgreSQL allows DBAs to define as many super users as they need, as well as regular users or groups. It is good practice to define a new user for each new database, to be used for regular maintenance. It is worth noting that PostgreSQL users and groups have nothing to do with those used by the operating system,

even if a direct association can be kept. It is also worth noting that if not explicitly specified, the `psql` command tries to automatically connect to a database named as the operating system user and with the operating system username.

Basic Role Management

Roles can be created with the SQL command `CREATE ROLE` and can be deleted with the `DROP ROLE` command; it is also possible to modify an existing role using the `ALTER ROLE` special command. In order to make it easier for script automation or command line management, the `createuser(1)` and `dropuser(1)` shell commands are also installed along with the cluster; these commands are wrappers for the SQL commands, so there is no difference in using the SQL or the shell ones. However, it is worth noting that if the shell command `createuser(1)` does not get enough parameters from the command line, it executes in interactive mode asking the user which properties of the new role have to be set, making it simpler than the SQL counterpart.

Before handling practical role creation it is worth listing some important features of a role that can be specified at the time the `CREATE ROLE` command is executed (or that can be set later via `ALTER ROLE`); please note that each setting has a corresponding `NO` value that disables the setting:

- `LOGIN` | `NOLOGIN` specifies if the role can be used to connect to the database or not. Creating `NOLOGIN` roles is useful for defining groups of privileges.
- `PASSWORD` allows the definition of the user password (either in plain text or encrypted text with the prefix `ENCRYPTED` | `NOENCRYPTED`), useful when the `pg_hba.conf` specifies a password base authentication (see previous articles in this series).
- `SUPERUSER` | `NOSUPERUSER` determines if the role will act as a superuser role, having all privileges on the database or not.
- `CREATEDB` | `NOCREATEDB` specifies if the new role will be able to issue a `CREATE DATABASE` command to create new databases.
- `CREATEROLE` | `NOCREATEROLE` specifies if the new role will be able to create other roles issuing another `CREATE ROLE` command.
- `REPLICATION` | `NOREPLICATION` indicates if the role can be used to connect to the cluster to get streaming replication data flow (see the previous article on replication).
- `CONNECTION LIMIT` specifies how many connections the role can open simultaneously to the backend (defaults to no limit).
- `INHERIT` | `NOINHERIT` specifies if the role will inherit privileges from other roles it belongs to (more on this later).

There are other parameters that can be specified, but the above are the most commonly used.

In order to see the role management in action, consider the `bsdmagdb` database that has a `bsdmag` superuser already set up and running, and let's create three roles corresponding to users `editor`, `reviewer_a` and `reviewer_b`. Top half of Listing 1 shows the creation of the roles within

Listing 1. Creating roles and inspecting them

```

Bsdmagdb# CREATE ROLE editor
WITH
LOGIN PASSWORD 'editor'
NOREPLICATION
NOSUPERUSER
NOCREATEROLE
CONNECTION LIMIT 2;

Bsdmagdb# CREATE ROLE reviewer_a
WITH
LOGIN PASSWORD 'reviewer_a'
NOREPLICATION
NOSUPERUSER
NOCREATEROLE
CONNECTION LIMIT 1;

Bsdmagdb# CREATE ROLE reviewer_b
WITH
LOGIN PASSWORD 'reviewer_b'
NOREPLICATION
NOSUPERUSER
NOCREATEROLE
CONNECTION LIMIT 1;

bsdmagdb# \x
bsdmagdb# SELECT * FROM pg_roles WHERE rolname like
        'reviewer%';
-[ RECORD 1 ]-----
rolname      | reviewer_a
rolsuper     | f
rolinherit   | t
rolcreaterole | f
rolcreatedb  | f
rolcatupdate | f
rolcanlogin  | t
rolreplication | f
rolconndefault | 1
rolpassword  | *****
rolvaliduntil | 
rolconfig    | 
oid          | 129165
-[ RECORD 2 ]-----
rolname      | reviewer_b
rolsuper     | f
rolinherit   | t
rolcreaterole | f
rolcreatedb  | f
rolcatupdate | f
rolcanlogin  | t
rolreplication | f
rolconndefault | 1
rolpassword  | *****
rolvaliduntil | 
rolconfig    | 
oid          | 129166

```

the database itself; please note that almost every important option has been specified even if not strictly necessary (it depends on its default value), but it is a good practice when creating roles to explicitly state all the options that the role will initially have. The bottom half of Listing 1 shows how it is possible to view a summary of the roles from the system catalog `pg_roles` (as readers can see, the password is never shown in plain text). Another way of getting the list of available roles, in a prettier formatted output, is using the `psql(1)` special command `\du` as shown in Listing 2.

As already explained, roles can be changed after their creation via the `ALTER ROLE` command. To simulate a real case, it is possible to grant to the `editor` role the permission to create other roles and to exclude the `reviewer_b` from connecting to the database since it had become an inactive reviewer. Listing 3 shows the application of the above changes and the result on the affected roles.

Who am I?

PostgreSQL stores in a special session value, `current_user`, the role the user is connected as, and therefore the

following simple query provides the information of the running role:

```
bsdmagdb=> SELECT current_user;
current_user
-----
editor
```

It is possible to dynamically change the role a connected user is using with the `SET ROLE` command, as in the following example:

```
SET ROLE uploaders_role;
```

Of course, to avoid security risks, a role can be changed only to another role that is tied to the former one (as a group, see later).

Roles as Groups

In the previous example two pretty much identical roles have been created: `reviewer_a` and `reviewer_b`. As readers can see, any user can belong to the “reviewer” group,

Listing 2. Viewing the available roles with the special command `\du`

```
bsdmagdb=# \du

                List of roles
-----+-----+-----
 Role name | Attributes                                     | Member of
-----+-----+-----
 bsdmag   | Superuser, Replication                         | {}
 postgres | Superuser, Create role, Create DB, Replication | {}
 editor   | 1 connection                                  |
 reviewer_a | 1 connection                                  | {}
 reviewer_b | 1 connection                                  | {}
```

Listing 3. Modifying roles properties

```
bsdmagdb# ALTER ROLE editor WITH CREATEROLE;
bsdmagdb# ALTER ROLE reviewer_b WITH NOLOGIN;
bsdmagdb=# \du

                List of roles
-----+-----+-----
 Role name | Attributes                                     | Member of
-----+-----+-----
 bsdmag   | Superuser, Replication                         | {}
 editor   | Create role                                    | +| {}
 reviewer_a | 2 connections                                  |
 reviewer_b | 1 connection                                  | {}
 reviewer_b | Cannot login                                   | +| {}
 reviewer_b | 1 connection                                  |
```

so it is worth creating a role that defines such group and grants any privilege common to any reviewer to the reviewer users.

User roles are assigned to group roles using the `GRANT SQL` command and are removed using the `REVOKE` command; virtually any user role can belong to as many group roles as needed. It is worth emphasizing again that since there is no distinction from a user role and a group role, a user role can be assigned to another user role and so on. The distinction is pretty much only semantic for the DBA, but internally PostgreSQL does not distinguish between the two kinds of roles. Nevertheless, usually group roles do not have login privileges, but this is not mandatory.

Intra-role assignments demonstrate the `INHERIT` property of a role itself: when this property is enabled (this is the default) the role will inherit all privileges from the other roles it is assigned to. If the inherit property is disabled, the enclosed role will need to explicitly change its status to get advantages of the enclosing role privileges. To better understand the inheritance mechanism, consider the creation of the groups `reviewers_group` and `uploaders_group`; the former will have read-only access, while the latter write only access. Then two other user roles are created: `reviewer_c` (which will explicitly inherit permissions) and `reviewer_d` (which will not inherit permissions) as shown in Listing 4. The groups are created without the login property and with some more restrictions just to be sure that assigning them roles will be safe. Lastly, all the reviewers' roles are assigned to the `reviewers_group`, while only the `reviewer_c` and `reviewer_d` are assigned also to the `uploaders_group`.

Therefore in this example there are two group roles, `reviewers_group` and `uploaders_group` that contain respectively four and two user roles, where only the `reviewer_d` doesn't have the inherit property. Please note that the distinction "user role" and "group role" is kept only for the sake of clarity, but as shown in Listing 4 there is no difference between the two concepts.

It is time to apply some privilege policies; the following examples will handle the `articles_to_review` table shown

Box 1.

Setting up a simple table to test privileges

The following SQL excerpt creates and populates the `articles_to_review` table used in the examples:

```
CREATE TABLE articles_to_review(
pk serial NOT NULL,
title text NOT NULL,
abstract text,
download_url text,
PRIMARY KEY( pk )
);

INSERT INTO articles_to_review( title, abstract, download_url
)
VALUES( 'FreeBSD userland',
'An article on FreeBSD Userland',
'http://bsdmag.org/download/art1.pdf' );

INSERT INTO articles_to_review( title, abstract, download_url
)
VALUES( 'OpenBSD kernel',
'An article on OpenBSD kernel development',
'http://bsdmag.org/download/art2.pdf' );
```

Listing 4. Creation of a group role and of two other user roles

```
bsdmagdb# CREATE ROLE reviewers_group
WITH
NOLOGIN
NOREPLICATION
NOSUPERUSER
NOCREATEROLE;

bsdmagdb# CREATE ROLE uploaders_group
WITH
NOLOGIN
NOREPLICATION
NOSUPERUSER
NOCREATEROLE;

bsdmagdb# CREATE ROLE reviewer_c
WITH
```

```
LOGIN PASSWORD 'reviewer_c'
CONNECTION LIMIT 1
INHERIT;

bsdmagdb# CREATE ROLE reviewer_d
WITH
LOGIN PASSWORD 'reviewer_d'
NOINHERIT
CONNECTION LIMIT 1;

bsdmagdb# GRANT reviewers_group TO reviewer_a;
bsdmagdb# GRANT reviewers_group TO reviewer_b;
bsdmagdb# GRANT reviewers_group TO reviewer_c;
bsdmagdb# GRANT uploaders_group TO reviewer_c;
bsdmagdb# GRANT reviewers_group TO reviewer_d;
bsdmagdb# GRANT uploaders_group TO reviewer_d;
```

Listing 5. Defining permissions for the `articles_to_review` table

```
bsdmagdb# REVOKE ALL PRIVILEGES ON articles_to_review FROM public;
bsdmagdb# GRANT SELECT ON TABLE articles_to_review TO reviewers_group;
bsdmagdb# GRANT ALL PRIVILEGES ON articles_to_review TO editor;
bsdmagdb# GRANT ALL PRIVILEGES ON articles_to_review_pk_seq TO editor;

bsdmagdb# GRANT INSERT ON TABLE articles_to_review TO uploaders_group;
bsdmagdb# GRANT USAGE ON SEQUENCE articles_to_review_pk_seq TO uploaders_group;
```

Listing 6a. Trying different operations on the `articles_to_review` table from different users

```
bsdmagdb=> SELECT current_user;
current_user
-----
reviewer_a

bsdmagdb=> SELECT pk,title,abstract FROM articles_to_review;
pk | title | abstract
-----+-----+-----
 1 | FreeBSD userland | An article on FreeBSD Userland
 2 | OpenBSD kernel | An article on OpenBSD kernel development

bsdmagdb=> INSERT INTO articles_to_review( title, abstract )
bsdmagdb-> VALUES( 'Bad article', 'N/A' );
ERROR: permission denied for relation articles_to_review

- ##### another connection
bsdmagdb=> SELECT current_user;
current_user
-----
editor
bsdmagdb=> INSERT INTO articles_to_review( title, abstract )
VALUES( 'Bad article', 'N/A' );
bsdmagdb=> SELECT pk, title, abstract FROM articles_to_review;
pk | title | abstract
-----+-----+-----
 1 | FreeBSD userland | An article on FreeBSD Userland
 2 | OpenBSD kernel | An article on OpenBSD kernel development
 3 | Bad article | N/A

- ##### another connection
bsdmagdb=> SELECT current_user;
current_user
-----
reviewer_c
(1 row)

bsdmagdb=> INSERT INTO articles_to_review( title, abstract )
VALUES( 'A new uploaded article', 'N/A' );
```

in Box 1. Privileges on database objects are assigned using the `GRANT` command and removed using the `REVOKE` command. PostgreSQL recognizes the special role `PUBLIC` as a way to refer to any role available at any time; moreover instead of requiring the DBA to manually specify a long list of privileges, the special clause `ALL PRIVILEGES` refers to any permission available on the object. A simple example of a security policy to start with is to remove any access to anybody to the `articles_to_review` table, granting then the `SELECT` permission to the role (group) `reviewers_group` and granting all privileges to the role (user) `editor` as shown in Listing 5. The grant for the sequence is required since the revoke of all privileges cascades to the sequence too.

As Listing 6 shows, the user `reviewer_a` is allowed to do a `SELECT` but not to alter the table, while the `editor` role can do both operations. The user role `reviewer_c` can perform an `INSERT` since it belongs to the `uploaders_group`, as well

as the `reviewer_d`. (see the bottom half of Listing 6). The difference between the `reviewer_c` and `reviewer_d` is that the former is enabled from the beginning to use the write privileges from the group `uploaders_group`, while the latter has to explicitly state that it wants to use the `uploaders_group` privileges by changing the current role. This explains the difference between having the `inherit` property or not: if inheritance is enabled, then the enclosed role will have its own permission set added by the set of permission of all the enclosing roles; if disabled, the enclosed role requires an explicit role change to use enclosing role privileges.

The role inheritance is broken at the first role in the chain that does not “re-export” the inheritance itself. As an example consider the role `reviewer_a`, that by default has the inheritance, and assign to it the `reviewer_d` role that does not. Therefore the final result is that the `reviewer_a` is in the group `uploaders_group`, as is the `reviewer_d`, but

Listing 6b. Trying different operations on the `articles_to_review` table from different users

```
bsdmagdb=> SELECT pk,title, abstract FROM articles_to_review;
pk |          title          |          abstract
-----+-----+-----
 1 | FreeBSD userland      | An article on FreeBSD Userland
 2 | OpenBSD kernel       | An article on OpenBSD kernel development
 3 | Bad article          | N/A
 4 | A new uploaded article | N/A

- ##### another connection
bsdmagdb=> SELECT current_user;
current_user
-----
reviewer_d
(1 row)

bsdmagdb=> INSERT INTO articles_to_review( title, abstract )
VALUES( 'A reviewer_d article', 'N/A' );
ERROR: permission denied for relation articles_to_review
bsdmagdb=> SET ROLE uploaders_group;
SET
bsdmagdb=> SELECT current_user;
current_user
-----
uploaders_group
(1 row)

bsdmagdb=> INSERT INTO articles_to_review( title, abstract )
VALUES( 'A reviewer_d article', 'N/A' );
INSERT 0 1
```

the inheritance is broken at `reviewer_d`, and therefore `reviewer_a` does not get the `uploaders_group` permissions without issuing a `SET ROLE`, as shown in Listing 7.

Therefore, the rule of thumb is that a role gets privileges of other granted roles as follows:

- all the privileges reachable through an *inheritance* path of roles;
- all the privileges of granted roles using an explicit `SET ROLE` command.

Assigning and Inspecting Privileges

Each database object is assigned to a set of permissions related to a role. Such permissions, commonly called also *ACL (Access Control List)* are stored as a text string in the `pg_class` catalogue. The format of the `acl` string is as follows:

```
targetRole=permissions/grantedByRole
```

where:

Listing 7. A broken inheritance example

```
bsdmagdb=# GRANT reviewer_d TO reviewer_a;
- ##### another connection
bsdmagdb=> SELECT current_role;
current_role
-----
reviewer_a
bsdmagdb=> INSERT INTO articles_to_review( title, abstract )
VALUES ( 'A reviewer_a article', 'N/A' );
ERROR: permission denied for relation articles_to_review
bsdmagdb=> SET ROLE uploaders_group;
SET
bsdmagdb=> INSERT INTO articles_to_review( title, abstract )
VALUES ( 'A reviewer_a article', 'N/A' );
INSERT 0 1
```

Listing 8. Inspecting permissions on the `articles_to_review` table

```
bsdmagdb=# SELECT c.relname, array_to_string( c.relacl, E'\n' )
FROM pg_class c
WHERE relname='articles_to_review';
```

```
relname      | array_to_string
-----+-----
articles_to_review | bsdmag=arwdDxt/bsdmag +
                  | reviewers_group=r/bsdmag+
                  | editor=arwdDxt/bsdmag +
                  | uploaders_group=a/bsdmag
```

```
bsdmagdb=# \dp articles_to_review;
```

Access privileges				
Schema	Name	Type	Access privileges	Column access privileges
public	articles_to_review	table	bsdmag=arwdDxt/bsdmag + reviewers_group=r/bsdmag+ editor=arwdDxt/bsdmag + uploaders_group=a/bsdmag	

- `targetRole` is the role to which permissions are applied;
- `permissions` is the permission string;
- `grantedByRole` is the role that granted the above permission string to the `targetRole`.

The permission string is very similar to the one shown by the file system command `ls(1)`, and therefore is made by 'r', 'w', 'x' and other letters. The meaning of each letter will be explained in the following. As an example of `acl` consider the following:

Table 1. PostgreSQL ACL letters meaning

ACL Letter	GRANT/REVOKE permission name	Meaning	Applies to	Allows for	Note
r	SELECT	Read permission	Table	SELECT, COPY TO	
			Sequence	curval(..)	
w	UPDATE	Write permission	Table	UPDATE	Often requires also 'r' permission for the columns in the WHERE clause or for SELECT...FOR UPDATE and SELECT...FOR SHARE statements.
			Sequence	nextval(..), setval(..)	
a	INSERT	Append permission	Table	INSERT, COPY TO	
d	DELETE	Delete permission	Table	DELETE	
x	REFERENCES	Refence permission	Table (columns)	FOREIGN KEY	Required on both sides of the foreign key.
t	TRIGGER	Trigger permission	Table	CREATE TRIGGER	
c	CONNECT	Connection permission	Database	\connect	The permission to connect to a database is checked after the <code>pg_hba.conf</code> has been validated.
C	CREATE	Create permission	Database	CREATE SCHEMA	
			Schema	General creation of objects into the schema (CREATE TABLE, CREATE SEQUENCE, CREATE INDEX, ...)	
			Tablespace	CREATE INDEX ... WITH TABLESPACE CREATE TABLE ... WITH TABLESPACE	
D	TRUNCATE	Truncate permission	Table	TRUNCATE	
X	EXECUTE	Execute permission	Stored procedure	Procedure call	
			Operator	Operator usage	
U	USAGE	Usage permission	Schema	Lookup of objects within the schema (i.e., "entering" the schema.	
			Sequences	nextval(..), curval(..)	
			Procedural language	Use the language and its facilities.	
T	TEMP	Temporary permission	Database	CREATE TEMPORARY TABLE	

```
editor=arwdDxt/bsdmag
```

where *editor* is the target role, that is the permissions are applied to the *editor* role; *bsdmag* is the granting role, that is permissions have been set up by *bsdmag* for *editor*; finally *arwdDxt* is the permission string that specifies which permissions *bsdmag* granted to *editor* on the current database object.

Permissions can be set up on any database objects and, in the case of tables, even on every column. Permissions for a database object are stored as an array of acl strings in the `pg_class.relacl` field, while permission for single table attributes are stored into the `pg_attribute.attacl` field. It is possible to query the system catalogue

or use the special command `\dp` of the terminal `psql(1)`, as shown in Listing 8.

A database superuser can always act on an object with all the permissions, even if a more restrictive string has been applied to its role.

Table 1 summarizes the ACL letters in the permission string and their meaning as well as different objects to which they can be applied. The permission string *arwdDxt* represents the `ALL PRIVILEGES` for a database table. As shown in Listing 8, the `uploader_group` can insert new tuples in the table (permission *a*) while the role `reviewers_group` can only select tuples (permission *r*).

Once an object is created the catalogue does not store any permission string, that is the `pg_class.relacl`

Listing 9. Automatically generated ACL strings for owner as of executing the first permission assignment

```
bsdmagdb=# CREATE TABLE p_test( id integer );
bsdmagdb=# \dp p_test
```

Access privileges				
Schema	Name	Type	Access privileges	Column access privileges
bsdmag	p_test	table		

```
bsdmagdb=# GRANT INSERT ON p_test TO uploaders_group;
GRANT
bsdmagdb=# \dp p_test
```

Access privileges				
Schema	Name	Type	Access privileges	Column access privileges
bsdmag	p_test	table	bsdmag=arwdDxt/bsdmag	+
			uploaders_group=a/bsdmag	

Listing 10. Setting up an author column and revoking permissions for non editor users

```
bsdmagdb=# ALTER TABLE articles_to_review ADD COLUMN author text;
bsdmagdb=# UPDATE articles_to_review SET author = 'Luca Ferrari';
bsdmagdb=# GRANT ALL PRIVILEGES (author) ON TABLE articles_to_review TO uploaders_group;
bsdmagdb=# REVOKE ALL PRIVILEGES ON TABLE articles_to_review FROM reviewers_group;
bsdmagdb=# GRANT SELECT (pk, title, abstract, download_url) ON TABLE articles_to_review TO reviewers_group;
bsdmagdb=# SET ROLE reviewers_group;
bsdmagdb=> SELECT author FROM articles_to_review;
ERROR: permission denied for relation articles_to_review
bsdmagdb=> SELECT title, abstract FROM articles_to_review;
```

title	abstract
FreeBSD userland	An article on FreeBSD Userland
OpenBSD kernel	An article on OpenBSD kernel development
A reviewer_d article	N/A
A reviewer_a article	N/A

is empty. This means that there are default permissions for such objects (that is all privileges for the owner and some other privileges for the *PUBLIC* – all other roles); once a `GRANT` or `REVOKE` command is executed on the object, the system places a permission string for covering also the owner permissions. As an example, in Listing 9 a new table `p_test` is created and the permissions just after the creation are empty; then a `GRANT` is executed for the role `uploaders_group` and the situation is that the system, in order to keep track of the owner permissions and of the `uploaders_group` ones, have to insert not only the latter permission string, but also an automatically generated string for the owner role `bsdmag`.

As already explained, permissions are assigned using the `GRANT` command and are removed using the `REVOKE` command. Since the `GRANT/REVOKE` commands are used in conjunction with different cases, there is not only one syntax. The general syntax is as follows:

```
GRANT <permission> <permission> ... ON <object-type>
      <object-name> TO <role-name>
REVOKE <permission> <permission> ... FROM <object-type>
      <object-name> FROM <role-name>
```

where the `REVOKE` can also have the `CASCADE` option to remove permission also to connected objects. Both commands accept the `PUBLIC` special role as a role indicating any user, as well as `ALL PRIVILEGES` to apply all the available privileges for the object. Moreover, it is possible to use the special clauses `ALL TABLES IN SCHEMA` and `ALL`

Box 2. psql(1) and Operating System Users

The command line PostgreSQL terminal `psql(1)` uses the operating system username as both the default PostgreSQL role name to connect with and the database to connect to if no option has been specified.

Therefore, assuming the operating system user is *luca*, a role named *luca* is defined into PostgreSQL as well as a database, the following statements are equivalent:

```
psql -U luca luca
psql -U luca
psql luca
psql
```

and connect the PostgreSQL role *luca* to the database *luca*.

On The Web

- PostgreSQL official Web Site: <http://www.postgresql.org>
- ITPUG official Web Site: <http://www.itpug.org>
- GitHub Repository containing the source code of the examples: <https://github.com/fluca1978/fluca-pg-utils>

`SEQUENCES IN SCHEMA` to apply permission management to all tables/sequences.

Permissions on Table Columns

PostgreSQL allows a fine grain permission definition: not only can permissions be granted or revoked on a per-table basis, but also on each individual column of tables. This can be really useful since it allows prevention of malicious or buggy applications from reading private data.

As an example, consider adding an *author* column on the `articles_to_review` table. Such a column should be of course be accessible to the `uploaders_group` users, that are going to set the value of the author, but should not be available at all to the `reviewers_group` users in order to not influence the reviewing process. Therefore, using a sequence of `GRANT` and `REVOKE`, it is possible to provide access to such column to the `uploaders_group` while revoking it to the `reviewers_group`. It is important to note that, since the latter group has unconditional read access to the table, it is required to remove such privilege and specify exactly which column will be allowed to be referenced in a `SELECT` statement. In other words, permission at the level of a table implies the same permission on each column. As shown in Listing 10, the `reviewers_group` cannot access the *author* column while it can access (in read mode) all the others.

Summary

This article presented the way PostgreSQL manages users, groups and the permission related to database objects, with particular regards to the table permissions. PostgreSQL allows DBAs to set up a very fine grain security policy on pretty much any aspect of the database.

LUCA FERRARI

Luca Ferrari lives in Italy with his wife and son. He is an Adjunct Professor at Nipissing University, Canada, a co-founder and the vice-president of the Italian PostgreSQL Users' Group (ITPUG). He simply loves the Open Source culture and refuses to log-in to non-Unix systems. He can be reached on line at <http://luca1978.blogspot.com>

Immersive 360x180°

Panoramic Photography in BSD

In this article I am going to show you what 360x180° panoramic photography is along with a few of its possible applications. I hope I can show you how fascinating it is. Panoramic photography is one of the most complete ways of experiencing a location without physically visiting. That is one of the reasons why it is also called “Immersive Photography”. In the following sections I hope I can inspire you to try it out with tools available under a BSD or any Unix like system, using only free open source software.

What you will learn...

- what 360x180° panoramic photography is,
- how to start making your own panoramic photographs,
- the main tips and tricks to transition completely to 360x180°.

What you should know...

- know how to use GIMP [1] or an image editor to correct small imperfections in an image,
- preferentially know how to use a command line,
- preferentially have the recommended equipment.

If you like photography and have a hacker spirit, which qualifies most BSD enthusiasts, you probably have already tried to join two or three pictures together to make a panoramic image. If you haven't yet, you probably

imagine that this is possible? What about Google Street View, do you know it. Figure 1 and Figure 2.

Well, anyone can guess what a panoramic image is.. One possible definition is that it is a picture that extends



Figure 1. Full Sphere Panoramic Image in a Stereographic Projection

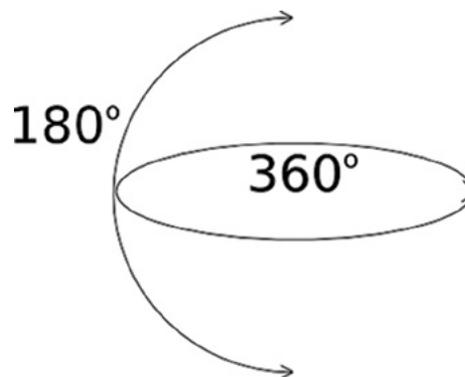


Figure 2. The Angles of a Full Sphere Panorama

the lens or the human eye field of view. The most accepted definition is that it is a question of the field of view degrees, but sometimes it is hard to discover the angle used in a ready made picture, so we finish with the image aspect. It is generally appropriate to call any picture with an aspect of 2:1, or wider, a panoramic photo. Most of the times it is achieved by joining pictures together to make a new one with a wider coverage angle. The "immersive" or "360x180°" or "full sphere" or "spherical panorama" is the panoramic picture that extends this to the widest possible angle, covering all visible areas. As it is shown in Figure 2.

To see a full sphere picture you will need special software or the image will have some distorted parts. The best way to look at them is interactively, in a computer or any other device, preferentially with a big screen. These pictures are usually not very much adequate for printing, although you can convert them to some projections that are better for that, like the equirectangular or stereographic. This last is also called "little planet" in an allusion to the classic "The Little Prince". Figure 1 shows an image in a stereographic projection.

After this little introduction it is a good idea to have an immersive experience. Follows some links below where you can see examples of 360x180° photography:

- <http://www.cartola.org/360/> – my blog, where I post the images I create and some how-to's. As my native language is Portuguese, there will probably have

always more posts in this language than in English. Sorry for that,

- <http://www.360cities.net/> – 360x180° pictures community with pictures from all over the world,
- <http://www.airpano.net/> – spectacular gallery from a non commercial group of Russian photographers.

Try to experiment with the immersive photographs, clicking and dragging to every side, up and down too, and zoom in and out. Google street view doesn't care about perfect quality and even this way their images cause a good impression. The possibility to have a point of view as if we were in the place and be able to chose what to look at is really amazing, isn't it?

Google Street View is available from the Google Maps [5] system but they have not shot everywhere yet. Small cities and countries might not have it. With it you can discover the place without going there physically. Try to see some streets in New York, Rio de Janeiro, London or Tokyo. It is totally amazing. Probably every BSD user already know how to use it, but anyway, you need to put the map over where you want, click and drag the orange little guy figure over the zoom rule on the left side and release the mouse button when the guy is over some street. The streets with street view available will turn to blue when you start dragging the orange guy. You will need flash plugin to see this.

By now you can probably imagine some of the possibilities for these images: hotel sites, tourism, real state, res-



Figure 3. Google Street View Man Figure

taurants, parks... Any enterprise or individual that want to show any place can take advantage of this. Even those crazy guys that only want to keep a family memory. Do you think it can be made totally with free software? Surely! The free software available to make panoramic images has existed for some time. Hugin, the main free software I use to stitch my photos, has its first files at source force dated from 2003. And it is only a GUI to other tools, like PanoTools, that existed before. The German physician and mathematician Dr. Helmut Dersch has began to make PanoTools in 1998. In 2004 I created the first Hugin and its dependencies ports to FreeBSD. In 2000 I have participated in a project to build an Internet Brazilian tourism portal that used 360° images to present hotels. It was a big news at that time, but the picture qualities were much poorer than today and also the computer and Internet capacities were inferior, so it was difficult to make higher resolution images. Nowadays it is possible to make images with very high resolution and commercial quality using only free software. So let's start talking about the process and free software tools that can be used to make such images. I have already made some speeches and workshops and I usually like to split the process into steps, like this:

- Shooting
- Joining images (stitching)
- Post editing (to correct possible joining imperfections)
- Publishing

Each step has its own details and secrets to achieve a good result at the end and I will try to talk a little about each. Before that, however, let me try to show you a simpler process to mount a partial panorama with just a few images.

Mounting a Simple Panorama

For a beginner it is surely easier to start mounting a partial panorama. For that you just need some images taken with some overlap between them, like 30 to 50% for example. There is no need of specific cares about the way you shoot and probably also won't have no need of post editing.

Hugin site has a very good English tutorial on it [6]. By the tutorial, one can become acquainted with the tool and start recognizing the menus and tabs. A good aspect of the tutorial is that it teaches you how to do the steps manually. It is not so important when stitching just two pic-

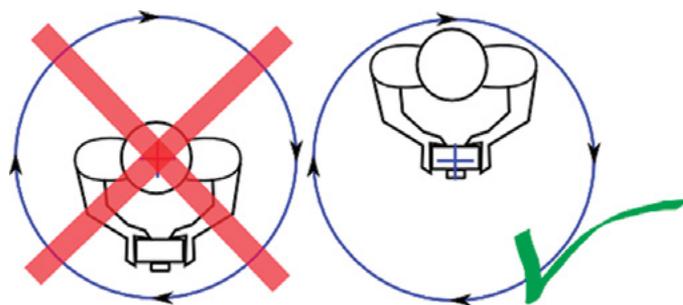


Figure 5. Turning without a tripod



Figure 4. Final image with many parallax errors and color differences

tures, but can be necessary to join a full sphere panorama, so it is a good way to start learning to do a full sphere.

Mounting a Complete Panorama – Step 1. Shooting

First of all we need to take the pictures (obviously). It is possible to use any camera to do a spherical panorama, but it is highly recommended that it has “Manual Mode” and that you use a panoramic tripod head in a tripod. It is also possible to make it not using all this, but it can put the newbie in such a difficult situation that they might give up without been able to finish their first panorama. Without the manual mode the color and luminosity difference between images will be very difficult to solve, although yet possible. One will also have a hard time to correct parallax errors without using an adequate tripod head. Nevertheless, my very first stitch had all these problems – and it never had a good final quality. It is shown in Figure 4. It has been made with a common point and shoot camera (Olympus C-120) using a common tripod and automatic mode. I have made 42 pictures to cover the full sphere. It is easy to see each original picture in the final stitched image.

For the beginners I strongly recommend making the previous step, mounting a simple panorama to get accu-

tomed to Hugin. Patience will certainly be of great value in your learning journey, because mounting a complete sphere panorama can take some time. If you are going to try to shoot without a tripod, some important tips are:

- turn using the machine as the rotation axis and not you. (as shown in Figure 5),
- overlap image pairs using a more or less 30% (1/3) factor,
- choose a scene with less "human lines" or "expected lines": prefer abstract nature than city,
- choose a scene with the subject as far as possible.

Those tips will make your first try much easier. Using a tripod would make it easy? Yes, but the best would be using a tripod with an appropriated head for panoramic shooting. Rotate the camera around the *No Parallax Point* makes a *huge difference* in the stitching step and it can be made using this tripod head I mentioned.

There are many models of panoramic tripod heads and you can even make your own one at home. There is a page in the Panotools Wiki [7] that shows many commercial models of them and also many home made ones in the section called "Self Made".



Figure 6. First picture for a panorama



Figure 7. Second picture for a panorama

But what is this thing called No Parallax Point? Even if you can't buy or make your own panoramic head, it can help a lot if you know what it is. This will decrease the perspective errors, usually called parallax errors, making it easy to stitch and decreasing the need for post editing.

No Parallax Point

In a few words it is the point that, if used as the axis for rotating the camera, won't cause any perspective errors



Figure 8. Joined panorama with some problems



Figure 9. DSLR on a panoramic head



Figure 10. Point and shoot on a panoramic head

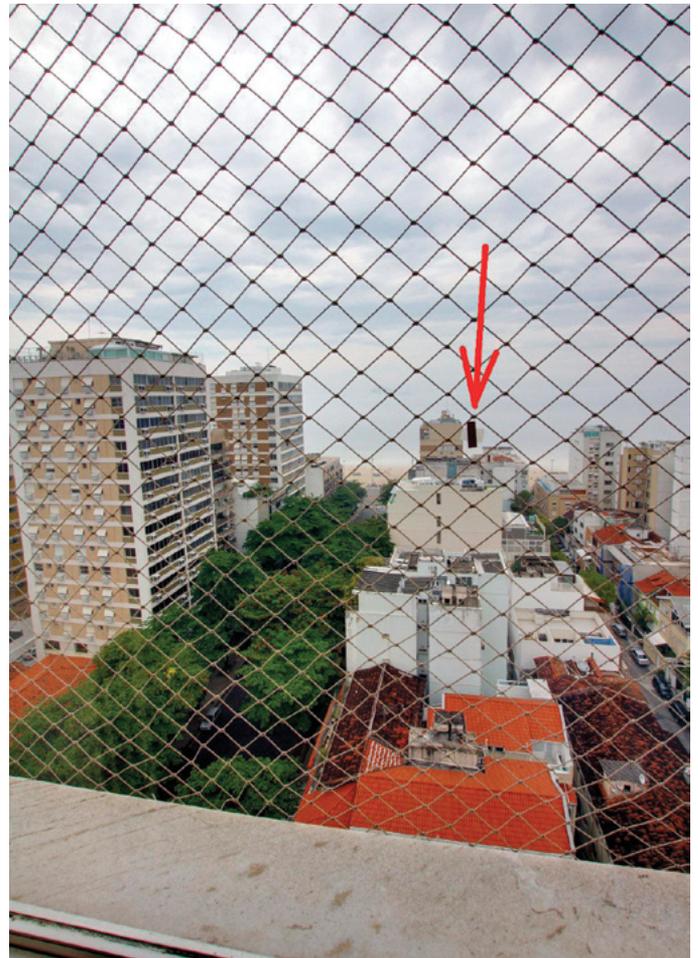


Figure 11. Left image with a tape to set the NPP

between the objects in the scene. Usually it is located in front of the camera, many times in the middle of the lens. In Figures 9 and 10 we can see some examples of panoramic tripod head being used to turn the camera using the no parallax point (NPP).

The No Parallax Point is sometimes confused [8] with the Nodal Point [9], which is an optical concept and not necessarily is the same point where one would be able to shoot pictures with no parallax errors [10]. Well, what I want to tell you here is that there is one special point where you can put the rotation axis of the camera and this way you will be able to take pictures rotating the cameras and have no parallax problems. And what is a parallax problem?

Between Figures 6 and 7 it is possible to see a parallax problem. Notice the big difference of position between elements like the island and the window, the buildings and the window and so on. An object that was behind the other can be seen in the next picture. The relative position be-

tween objects changes as we rotate. The resulting image will have probably some problems. In Figure 8 I tried to join the two pictures. We can see lines not joining (1), a building that was distorted (2) probably to make it possible to join another part of the image and an object showing twice (3).

If we use an appropriated tripod head and rotate the camera around the NPP the objects would be exactly in the same relative position, making it much easier to stitch pictures then.

And how do we use the NPP? In Figures 9 and 10 we can see two different cameras using a professional tripod head that allows horizontal and vertical rotation using the NPP allowing a full sphere shooting. The rotation axis is marked by the green line. The PanoTools Wiki [11], which is a great panoramic reference, has an article about tripod heads [12].

Besides a lot of commercial models we can also see some “Self Made” heads, where two of mine are also listed. I am also making a tutorial for a simple self made head very efficient[13]. By the time of this writing it was only in Portuguese, but the many images there can give you a good idea of the suggested ideas of simplification.

It is important to find the NPP of your camera+lens and for this it is necessary to follow some steps. One very interesting method is the tape on the window. In Figures 11 and 12 we can see two images showing a tape fixed in the glass of a window.

The camera is not perfectly positioned in the NPP, as we can see by the small different position of the tape in relation to the buildings behind. However, the position is already good enough to reduce drastically the parallax errors in a final stitched image. Look at the result in Figure 13.



Figure 12. Right image with a tape to set the NPP



Figure 13. Stitched image using images taken with a better NPP



Figure 14. *Image made with ISO 6400: too much noise*



Figure 15. *Image filtered with Neat Image*



Figure 16. *Image filtered with ImageMagick*

There are some very good references about setting the NPP, like the John Panos site [14]. It can take some time to set it up, but it is something that you can do only once. There is also a database of already known camera+lens at the PanoTools wiki [15]. For updated references besides other not shown here refer to my blog reference page [16].

Image Treatment

Yet in the photography phase I would include the image treatment. Almost all picture can be improved in an image editor and the full sphere panoramic images usually can take advantages of some techniques like HDR, noise reduction, color correction, exposure fusion, focus stack and others. Here are some free software tools I would recommend, not all of them directly available for BSDs. Specifically the “Neatimage” has a Linux version that runs easily with Linux emulation under my FreeBSD.

UFRaw [17], RawTherapee [18] or Darktable [19]

If you don't know what a raw image format is you can just ignore it and work with the JPG, jumping this topic. These are free softwares that will make the initial RAW treatment in case you have shoot in RAW format. This image format is available in many cameras and is capable of storing much more information than the traditional JPG format, allowing more flexibility in the image treatment process. These three are available for installing from the FreeBSD port system. In NetBSD packages you will find the UFRaw as a GIMP plugin or you can try the FreeBSD ports on it. Some other softwares are able to deal with raw format, like Shotwell [20]. I just didn't list it above because it has limited edition capabilities, being more a picture organizer.

GIMP [1]

Probably a well known tool for many of the BSD world. The GIMP is, in a few words, the free software alternative for Photoshop. It is an image editor. With it we can make most of the image improvements like color, luminosity, focus and many more. As an old and mature tool it is available for all most famous BSD flavors, including DragonFly BSD.

Neatimage [21]

This is a very good noise reduction tool. It is not Open Source, but has a free version that deals with JPG images and is available for Linux, Mac and even Windows! As I have said, I run the Linux version with no problems in my FreeBSD desktop. In Figure 14 a closeup in a section of an example image with much noise. It used ISO 6400 and

with higher ISO values the noise gets stronger. In Figure 15 the same section treated with Neat Image.

ImageMagick [22]

What to say about ImageMagick? It can make almost everything with images in the command line. It is a very mature and powerful set of image tools to do many many things related to pictures. Here I mention it as an alternative tool to use as noise reduction filter. It can take a little more time to calibrate the parameters to use, but, as in panoramas we use many similar images, it can save you much more time using it in a script to process all the panorama images in the same way, as it needs to be done. I have tested the commands suggested in the ISONOISE script done by Fred [23]. The suggested command sequence is:

```
convert $infile -median $radius $tmp0
convert $infile $tmp0 -compose Difference -composite
    -threshold $thresh% $tmp1
convert $infile $tmp0 $tmp1 -compose src -composite $outfile
```

You must change `$tmp0` and `$tmp1` to temporary file names, `$infile` and `$outfile` for the input and output files and `$radius` and `$thresh%` with the radius and threshold values. Those last two are the ones you change until achieve the result desired. Notice that the parameters used as variables are ready to be used in a shell script or you can also download the ready made script from Fred's mentioned site, of course. Notice, however, that they are bash scripts prepared to use bash installed at `/bin/bash` and bash is not installed by default on BSD systems. I got the result shown in Figure 16 with a radius of 30 and a threshold of 0% using the commands directly, not the script (it made difference as the script doesn't do exactly those commands). Watching the pictures in a



Figure 17. Image showing well the outdoor

larger size in my computer I consider the resulting image using ImageMagick better than the one using Neat Image, but it can vary from image to image. Other tries can be made with the approaches done by Fred in his scripts called noise cleaner, denoise, statsfilt or others.

Enfuse [24]

This tool is able to make exposure fusion. Imagine a scene where the contrast is too high. In Figures 17 and 18, for example, I took a picture inside a living room with daylight outside. It is really difficult to make one only picture showing at the same time the details of the dark and the light parts of the scene. To solve this problem we can take pictures with different expositions and combine them with Enfuse. At the end we will have a picture showing good details in all areas, like I did in Figure 19. In full sphere panoramas the chances to have areas with different lights are obviously higher.

In the following table the result of a searching I have made of the availability of those tools in the most known BSD systems (Table 1).



Figure 18. Image showing well the indoor



Figure 19. Image generated by Enfuse, joining the two expositions

Table 1. The availability of tools in popular BSD systems

Application	DragonFly BSD	FreeBSD	NetBSD	OpenBSD
UFRaw	GIMP plugin	YES	GIMP plugin	YES
Raw Therapee	NO	YES	From FreeBSD	NO
Darktable	YES	YES	From FreeBSD	YES
GIMP	YES	YES	YES	YES
Enfuse	YES	YES (within enblend)	YES	YES

Mounting a Complete Panorama Step 2. Joining Images (Stitching)

After all image treatment is over it is finally time to join the pictures and make the panorama. The tool for this is Hugin [25]. It will do all the distortion, rotation and position job to make a perfect stitch with our images.

The stitching is probably the hardest part of making a full sphere panorama. It can be made in many different ways, even considering only the use of one tool, and there are many tools for it. Any way you choose to do it can lead you into some problems and to solve many of the possible problems you will need to understand some concepts understand better the tool used.

I will try to talk first about the automatic process, then will give you some hints to try to use a partial automatic process. The most guaranteed, although, is the manual process, but it will surely take more of your precious time. Take a chance to read the many tutorials available at Hugin's site and also all over the Internet.

We can use the tool with it's automatic wizard or we can use more manual steps and take control of what it does. Each choice has it's price. The automatic can fail to find common points between images or to align the panorama and the manual way is a little complex, which means that you will need some time to learn some concepts and try a lot before get experienced.

The wizard is available in the first tab, shown in Figure 20. To use the automatic way we need to press on-

ly three buttons to finish the panorama, generating a final equirectangular tiff image. For this automatic process to work it is very important that the pictures were very well done, with enough overlap and preferably using the NPP.

So you can try the three buttons:

- Load images...
- Align...
- Create panorama...

If you are lucky you can then jump to the step where I suggest how you can visualize the immersive panorama, at the end of this section.

Depending on your camera, the loading images step can show you a window to fill some values about your lens and sensor. It won't happen if your camera already writes those values in the image file, but if not, the window on Figure 21 will pop up. It is not very important to fill it correctly. The optimization steps will find the better values. One very important thing to set is the lens type. Be sure to set it to fisheye (any) or stereographic if you are using a fish eye lens or keep it with "Normal (rectilinear)" if you are using a normal lens, otherwise the optimization will fail, automatically and manually. If you don't know the other values, just keep "1" in the "Focal length multiplier", fill a reasonable angle in the "HFOV", which should be the horizontal field of view of your lens in angles and the "Focal length" will be calculated.

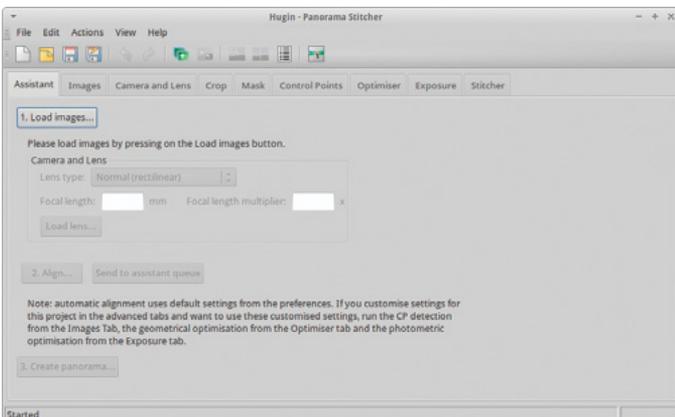


Figure 20. Hugin main window

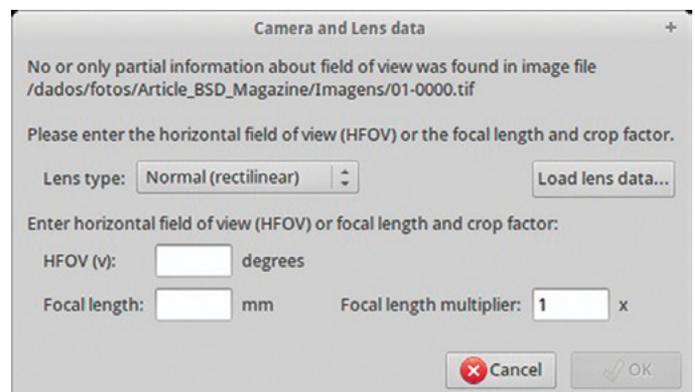


Figure 21. Camera and lens data window

If the automatic way fails we can try the manual, but before that there are a couple of things you can still try to make the automatic work. One first hint is to try to stitch one horizontal line of images first. Then you can insert other lines in next steps. Some scenes can be harder to stitch, like those made with narrow lens that can have images without visible common points. Also very hard are the images with moving objects, like clouds, trees in the wind, moving people and so on.

A second try to make the automatic work is to generate the called "key files" before aligning. The generation of each image "Control Points Candidates" is a high memory consumer process. The strategy is to save this information in files, so the step is done before trying to match Control Points between images. To do that, first load your images and save the Hugin project file. After that, open a command line. Supposing you are in a BSD system, do this in your csh/tcsh:

```
foreach i (`jot 10`)
cpfind -k $i project.pto
end
```

This commands suppose you have 10 images in your file and the Hugin file is called *project.pto*. That *jot* command will generate ten numbers starting at zero and the *foreach* will make a loop until the *end*. Adjust the steps to your case or simply run *cpfind -k N project.pto* for N between zero and the number of your images minus one. After that try again the automatic process.

Going to the manual mode, the first step is to find common points between images. Those are called Control

Points or just CPs. To find them it is mandatory that we had overlapped a little each image pair. If you did this, now it is time to show Hugin that, for example, that little rock in one image is the same as the rock in the other image. Using these CPs Hugin will be able to distort, roll and position the images to stitch the panorama. Some manual steps have been shown in the tutorial mentioned in section "Mounting a simple panorama". That tutorial lacks some other important concepts, so here follows some hints to add manual control points:

- there are normal CPs, vertical line CPs, horizontal line CPs and lines[26],
- the most important are the normal ones, the ones you set between two images that overlap,
- the lines are set in only one image, putting it in both sides of the panel, like in Figure 23,
- set at least 2 CPs between each image pair: I usually like to set 3 and the automatic process usually set around 20,
- add CPs distant one from another: I try to put one in the top, one in the middle and another one in the bottom of each overlap of two side by side images,
- don't set CPs over moving objects like leaves, clouds and people, among others,
- add at least one horizontal or vertical line at each two images: this is not something you must do, but will help to level the panorama.

In Figure 22 we can see the "Control Points" tab with a normal CP. Notice that it is set between two images. We

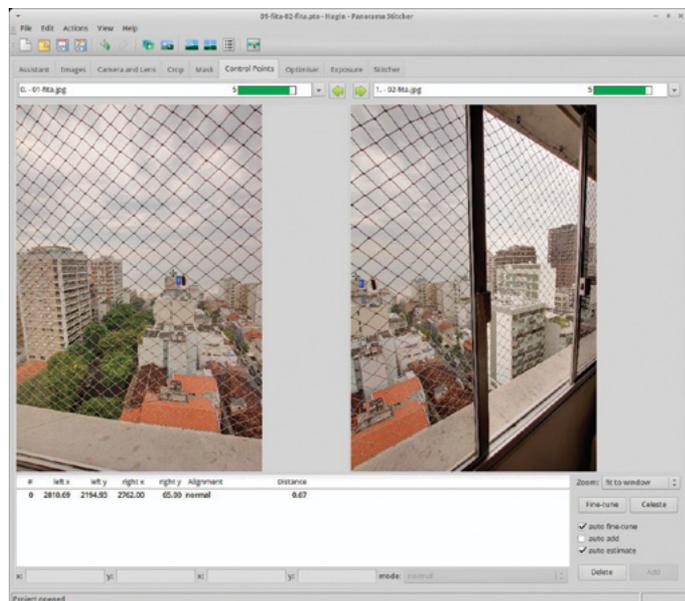


Figure 22. Normal Control Point set in the Control Points tab

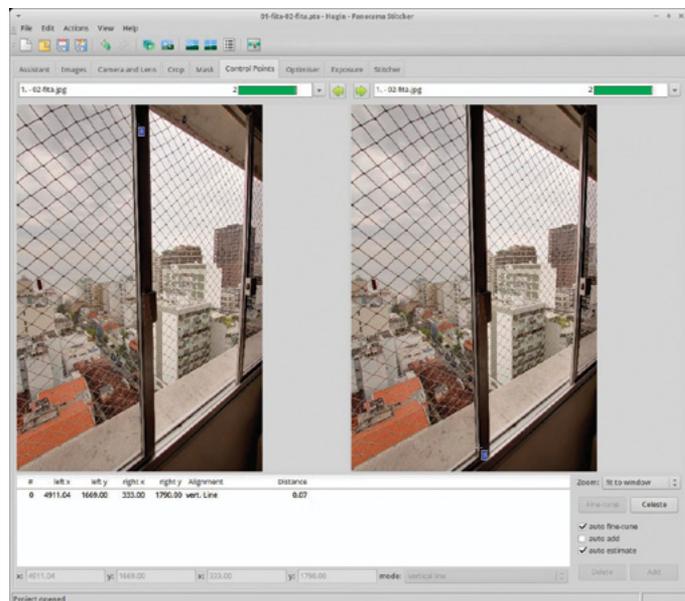


Figure 23. A line Control Point, set in one only image

put one image in the left and another in the right to add this CP. I have set more CPs between those images, but have shown only one in the picture to simplify.

To set normal CPs:

- choose the "Control Points" tab,
- choose one image in the left panel and the next image in the right panel
- optionally unset the options "auto fine-tune" and "auto estimate" (i like to unset them),
- try to look at a good candidate to a control point and click on it in one of the images (any of them),
- the software will zoom in the clicked image so you can optionally reposition the point,
- click on the correspondent point in the other image,
- the software will zoom in the other image too, so you can reposition the second point,
- optionally click the "Fine-tune" button or use the "f" keyboard shortcut to make an automatic fine adjustment,
- click the "Add" button or use the "a" keyboard shortcut.

Notice in Figure 22 the point defined by the small blue rectangle with a zero inside. This is the CP number. Right beside it there is a small cross, which is the specific place where the CP is located.

Set more CPs until you have the desired number of them between the two images. Remember to set at least two and to make them far from each other. You can navigate to the next pair of images using the small

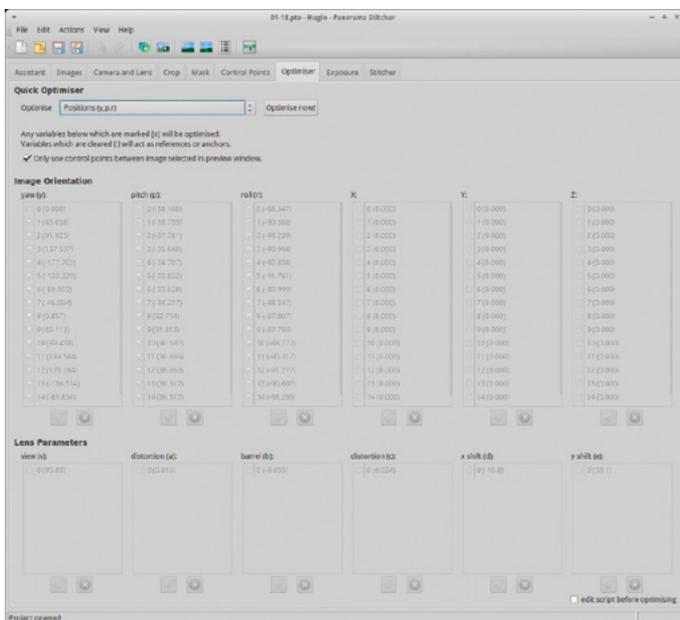


Figure 24. Hugin „Optimizer“ tab

green arrow between the image names. Don't forget to close the circle making CPs between the last and the first image.

About the other type of CPs, in Figure 23 we can see a line CP. Remember math classes? A line is defined by two points. To set a line we put the same image in the left and right panel and add the two points that define it. In Figure 23 there is one point of the line in the upper part of the window and another in the lower part.

Usually I set vertical lines at each two or three images. In a 10 images set, for example, you can try to set around 5 vertical lines. Sometimes it is not possible, because there is no vertical line. In nature pictures, for example, it happens. You can then try horizontal lines or don't use it at all, leveling the panorama by hand after optimizing it. I will show it in a while.

After setting up the CPs it is time to "Optimize". This is the process in which the tool makes many interactions to distort, roll and reposition images until the Control Points are positioned as good as possible. This means that a point, represented in one image, must be positioned right over the same point represented in the other image. Usually it is impossible to make all points get perfectly positioned with its pair. The measurement to see if the op-

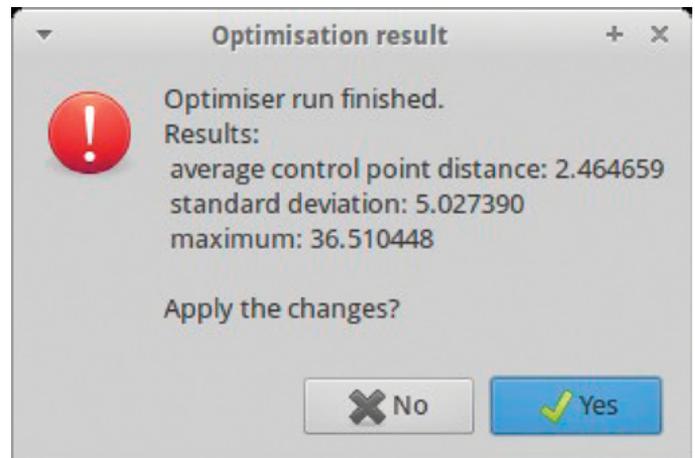


Figure 25. Optimization result

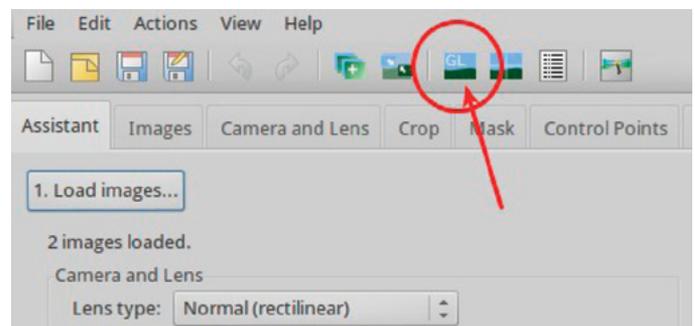


Figure 26. Fast preview panorama window button

timization went well is the maximum and average CPs distance. The utopian case would give us 0 distance for all CPs.

The "Optimizer" tab is shown in Figure 24. I usually do my optimizations in three steps. In any optimization the program takes the actual positions and distortions and try to change the requested parameters to achieve a better result. It means that it is a good idea to save the project without any optimization, so, if you want to try a different approach you just reopen the file. The following are the steps that I use in most cases.

First I ask it to change only "Positions (y,p,r)". Those are angular positions: yaw, pitch and roll. It changes the images positions around the sphere of the visual area.

After optimizing positions, usually the result is not the best. The average and maximum CPs distance are still too high, but I accept them in the confirmation dialog, like the one in Figure 25. It shows a very good result for a first

position optimization. After this first step the values can be much higher, like hundreds of points in the maximum distance.

As a second step I optimize "Positions and view (y,p,r,v)" and accept the result again.

The final optimization is "Everything without translation". If I try this last option first, usually the result is something really crazy.

After optimization it is a good idea to check the "Fast panorama preview window", what we do by clicking the little blue and green button shown in Figure 26 and the window shown in Figure 27 shows up. The "Fast preview" window is not only to look at the panorama, we can make some adjusts on the resulting image.

Notice the tabs, each one with a specific purpose. Specifically the "Move/Drag" tab, shown in Figure 28, is important to manually level the panorama as I have mentioned. Using it you just click and drag over one picture

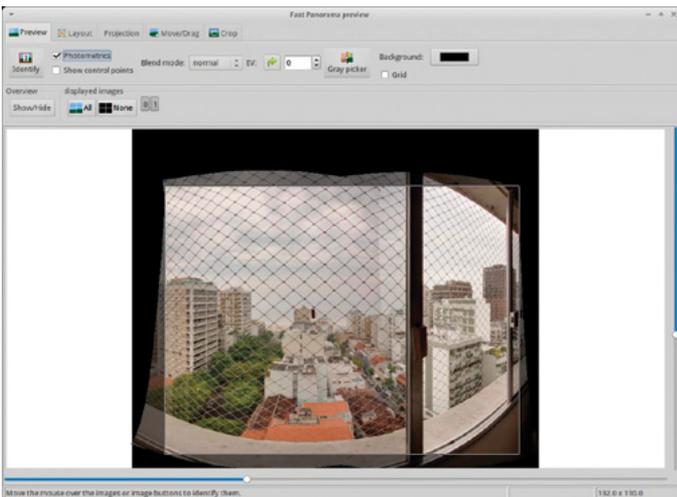


Figure 27. Fast preview panorama window

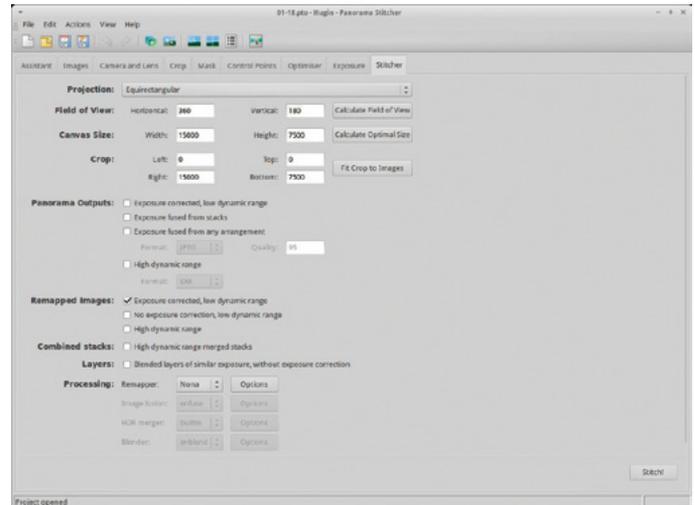


Figure 29. Hugin Stitcher tab

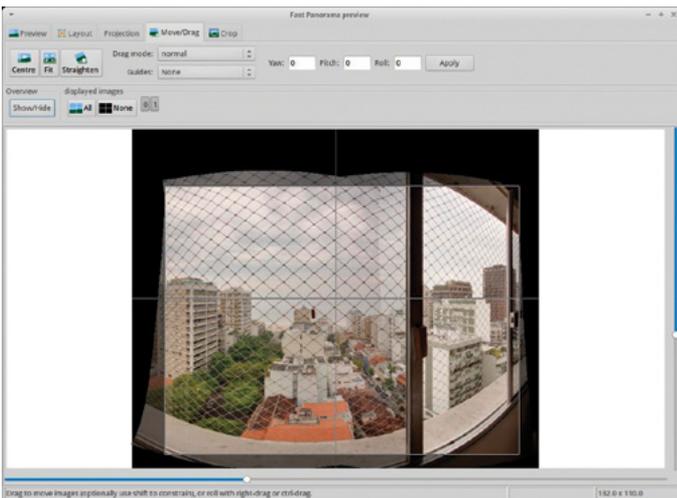


Figure 28. Fast preview panorama window "Move/Drag" tab

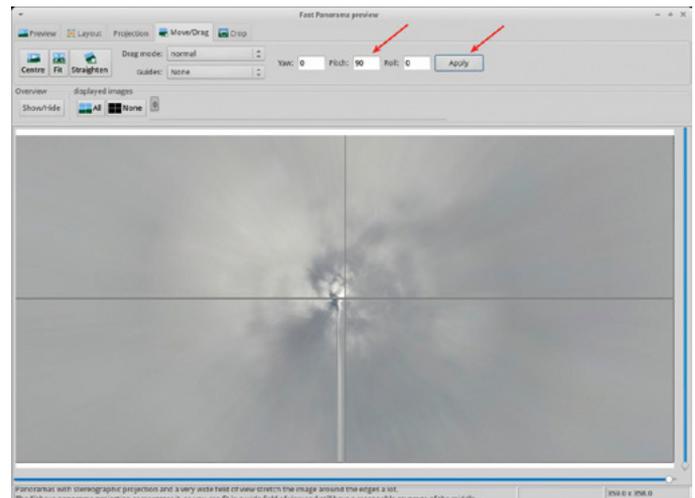


Figure 30. Move/Drag tab to pitch to a Little Planet

and the others, that are connected to it by the CPs, move according to your needs. Other buttons can be very useful for partial panoramas, like the "Center" and "Fit". The "Straighten" button will make a try to level the panorama. You can also test it and use Ctrl+Z if it fails.

After optimizing it is time to generate the final image. Sometimes it is still interesting to make some adjusts using masks, reviewing bad CPs, setting up lines and re-optimize. When the result is good enough we go to the Stitcher, shown in Figure 29. The hints for this tab are:

- the "Equirectangular" projection is the most common and we will use it,
- use the "Calculate Optimal Size" to have the better resolution of the final image,
- chose your preferred output format from TIFF, JPG or PNG,
- if you want each individual image distorted and replaced to make adjusts in an image editor, choose "Remapped Images" and in "Nona" options unset "Save cropped images"

Choose your options, click in the "Stitch!" button and voila! ... well, it is probably going to take some time to Hugin to finish your image, just be a little patient and watch the progress. In fact it is also a good idea to save the file before stitching.

That ready made image can be viewed interactively with viewers like Panini [27] before being published. It is necessary to compile it under BSDs, but it is not difficult. You just need to install some qt4 dependencies and follow the instructions in the panini-build.txt file.

You can try to find other options here: http://wiki.panotools.org/Panorama_Viewers#Stand_alone_Viewers or you can jump to the publishing section.

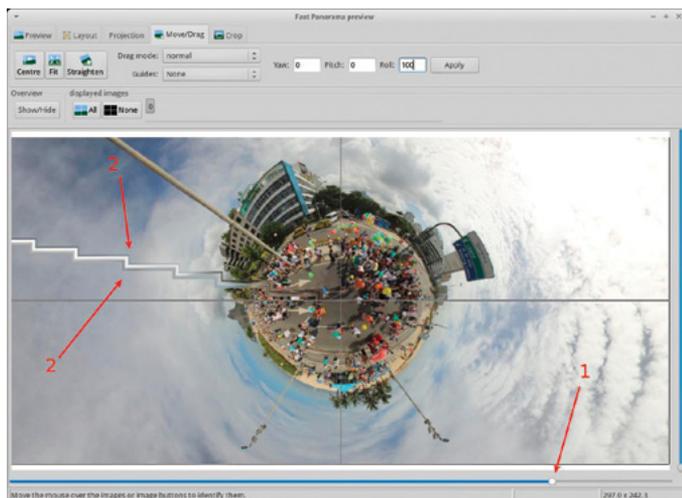


Figure 31. Move/Drag tab to adjust a Little Planet

The Little Planet Format

Well, there are many little planets in the article, so I think I must talk about them. They are a very interesting format of the full sphere panoramas. In fact they are a specific projection, as already said, that is one of the most interesting to get your panorama printed.

To do them I prefer to finish the equirectangular and even the final corrections, mentioned in the next section *Post editing*. Then you come back to Hugin, and that is why I am talking about it here.

- load the final equirectangular panorama as the only image in Hugin,
- configure the lens as an equirectangular with a 360 HFOV,



Figure 32. One more Little Planet

- go to the "Stitcher" tab and configure the projection to "Stereographic",
- go to the "Fast preview panorama" window and change to the "Move/Drag" tab, as shown in Figure 28,
- put 90 as the "Pitch" value, as shown in Figure 30, then click the "Apply" button,
- adjust the zoom with the horizontal scroll shown in Figure 31, arrow number 1,
- don't mung with lines like the one shown in Figure 31 arrow 2, cause they are only present in the fast preview, the rendered image won't have it,
- finish the final positioning by dragging and moving the picture, then you can stitch as you have stitched before: choose your image size and stitch it!

Et Voilà, one more little planet in the planet!

Mounting a Complete Panorama Step 3. Post Editing

Even after adjusts made in Hugin probably other corrections are still necessary to make the final image perfect. Some not well joined lines, half parts of objects and people and removing the tripod image are good examples of things that can need more work. It is possible to do this job in some different ways, but all of them implies the use of an image editor like GIMP. It is curious how the best tricks to make these corrections are the same used to do some virtual plastic surgery. I have learned them searching for tutorials on making people thicker, removing pimples and acne or things like that. "Extreme makeup" is also a good thing to search, just add "GIMP" or your favorite image editor to the search terms. I expect that you already know the basics of GIMP and there are plenty of material and tutorials out there, so I will only mention the main tools I use, which are:

- clone,
- healing,
- bend,
- copy and paste a feathered selection then move and/or rotate it,
- IWarp filter (inside "Distort" filters).

They will enable you to make a bent line straight or to cut a little part of the image and paste it in another place making the joints smooth. The equirectangular image allow us to correct many problems in the vertical central area of the picture. Even in some distorted places it is still possible to correct errors, like the one shown in Figures 33 and corrected in Figure 34.

Closer to the vertical extremes of the equirectangular the distortions make it impossible to do those corrections. One of the least obvious tricks is used to edit the nadir point of view. The equirectangular image does not allow we to edit it at all, nor the zenith.

One alternative using Hugin is:

- load the equirectangular into Hugin,
- pitch it 90 degrees in the fast preview window,
- generate a new image to edit into GIMP and then do the steps back.

I sometimes prefer to use another tool, in fact it is a tool kit called `Panotools::Script` [28] made by Bruno Postle. It is a Perl package available at CPAN: <http://search.cpan.org/dist/Panotools-Script/>. When installed it gives us two tools, among others: `erect2cubic` and `cubic2erect`. These two scripts, as their names suggest, make the conversion from equirectangular and cubic format in both ways. The installation can make you a little tired, as it will need many other CPAN Perl packages dependencies installation.



Figure 33. Bad joined line error



Figure 34. Bad joined line error corrected

In the cubic format we have 6 separated images, as shown in Figure 35. You can think of them as if they were the 6 faces from a cube with you inside. Each image is in a rectilinear format, with no distortion. Straight lines are straight on them. This format finally allows us to edit the zenith and nadir views easily and you can take the opportunity to put some logo or message at the nadir. After finished you can convert it back to equirectangular with the *cubic2erect* tool.

The *erect2cubic* can be used like this:

```
erect2cubic --ptofile=cubic.pto --erect=yourEquirectangular.jpg
```

You just replace the name *yourEquirectangular.jpg* with the name of your equirectangular file and give the PTO file the name you want. I suggested *cubic.pto*. After that you can run:

```
nona -o cubic cubic.pto
```

nona is a command that comes with Hugin and in this case will output the 6 remapped images, that correspond to the 6 cube faces. In this example it will give us 6 files named *cubic0000.tif*, *cubic0001.tif*, *cubic0002.tif*, *cubic0003.tif*, *cubic0004.tif* and *cubic0005.tif*. The nadir file is the last one and the zenith is the *cubic0004.tif*. You can edit any of them and the command to come back to equirectangular would be:

```
cubic2erect cubic0000.tif cubic0001.tif cubic0002.tif cubic0003.tif cubic0004.tif cubic0005.tif newEquirectangular.tif
```

Panini is capable of loading the cube faces to show you the interactive panorama and Panotools::Script comes with many other tools that can automate the panorama



Figure 35. The cubic format

generation outside Hugin, using the command line or a script. Cool, isn't it?

Mounting a Complete Panorama Step 4. Publishing

To finally show the world your precious art it is necessary to publish it at the Internet. Here I am going to show you two ways of doing this: one using a site where everybody can publish full sphere panoramas and a second way, publishing it in your own site.

Publishing on a Ready Made Site

The easiest way to publish is using the site 360 Cities [29], shown in Figure 36. You just need to do a free registration and upload the equirectangular image. It will take a short time for it to be approved, probably one day or so. To increase the chances of approval you should make an image with at least 6,000 x 3,000 pixels and fill the fields that describe the picture, including its local in google maps. If you are lucky your picture can be selected to Google Earth's 360 cities gallery.

It is possible to embed the picture published in 360 Cities in your own site, you just need to use the tags they give at the panorama visualization. You need to use the "EMBED & SHARE" button, seen in Figure 37 arrow 1. Then you must agree with the non commercial terms and a new window opens. After it you need to copy either the code to embed the panorama in your site, seen in arrow 2 in Figure 37, or the link to refer to it, seen in arrow three in Figure 37.

I know that the TourWrist [30] site is another alternative, where maybe you will be able to publish less quality panoramas. I know they allow you to publish partial panoramas and they have an application for Apple devices.

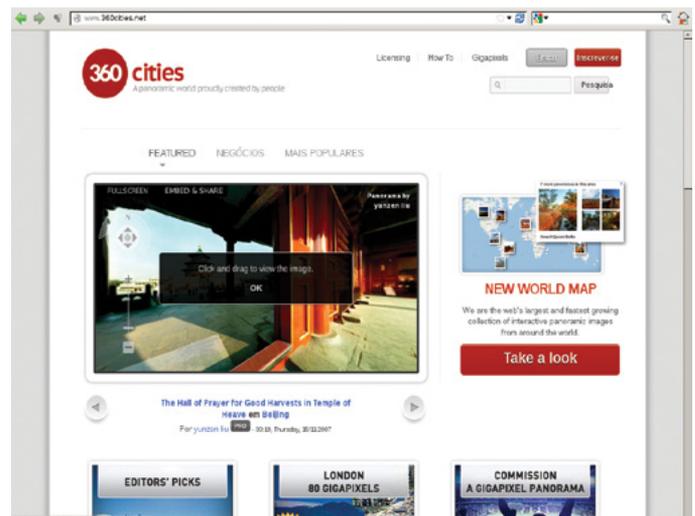


Figure 36. 360 Cities site, main page

Publishing on Your Own Site

To publish by yourself you can use, for example, flash, java and html5 formats and put the panorama in your own site. Among the free options, I prefer the Salado Player [31] for flash enabled devices and VR5 pano viewer [32] for the iThings and other HTML5 browsers. Pannelum [33] is also a new option for HTML5. The java plugin, PTViewer [34], made by the notorious Helmut Dersch [35], is heavier, has fewer features and won't be able to deal with very large images. It will accept the equirectangular format directly, and in fact only it. That makes publishing much easier, but is one of the reasons of the image size limit. At his site there are other newer alternatives, which you can take a look, but I haven't tested them yet.

Talking about the ones I prefer, to use the Salado Player we need to convert the image as a first step. This is done using Salado Converter [36], a small Java software that does the job. It transforms the equirectangular into a "DeepZoomCubic" format, which I recommend.

So, the first step is to download Salado Converter and uncompress it in some directory in your machine. After that you will have a file called *SaladoConverter.jar*. You might have to increase the java heap size to work with larger images, so I recommend using this command line to call it:

```
java -jar -Xmx3g SaladoConverter.jar
```

This will use 3 gigabytes of memory for the JVM heap size. To guarantee that it will work you can use as much as your computer has available. The program will then start with a window like the one shown in Figure 38.

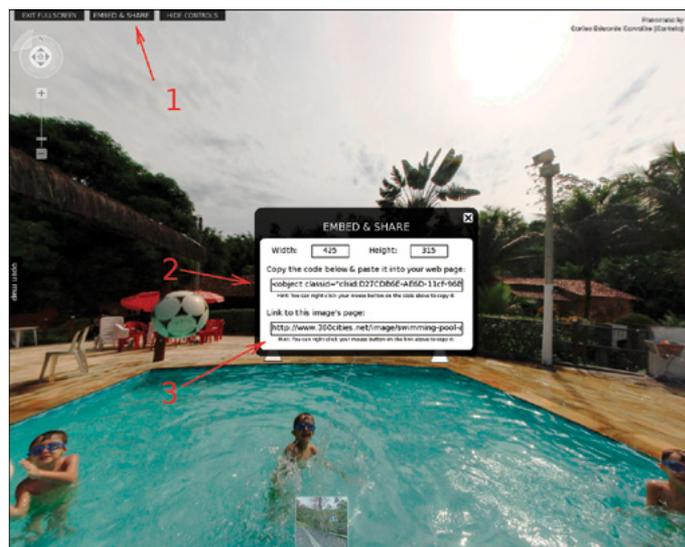


Figure 37. Embedding or linking to a 360 Cities panorama

Keep
FreeBSD
Free!



The
FreeBSD
FOUNDATION

Support FreeBSD
by donating to
The FreeBSD
Foundation



To find out more,
please visit
our Web site:

www.freebsd.foundation.org

Then you just need to add your image using the *Add* button and then the *Run* one. If everything is ok it will open another window, like the one from Figure 39, showing the conversion progress. The image is converted into 6 walls, corresponding to each cube face. It shows also the percent progression of one by one until the end. When it finishes, this second window gets closed. The output will be in the output directory that was set on the tool. Usually it creates a sub directory below "output", with the same name of the image a *dz_* in front of it. The *dz* means "deep zoom".

If you get an error you can try to decrease your equirectangular image size or increase the memory used. The image size limit which you will be able to work with will depend on the amount of memory available in your hardware.

Inside the *dz* output directory there will be many sub folders and files, but we don't need to deal with them separately. We just need to deal with the complete *dz* folder.

After converting the image it is necessary to configure the Salado Player XML file to point to the *dz* directory. The XML file has many other possible configurations. We can

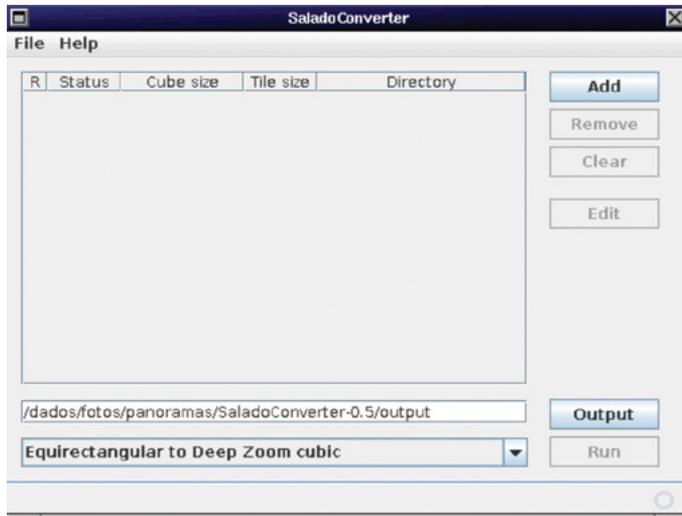


Figure 38. Salado Converter Initial Screen

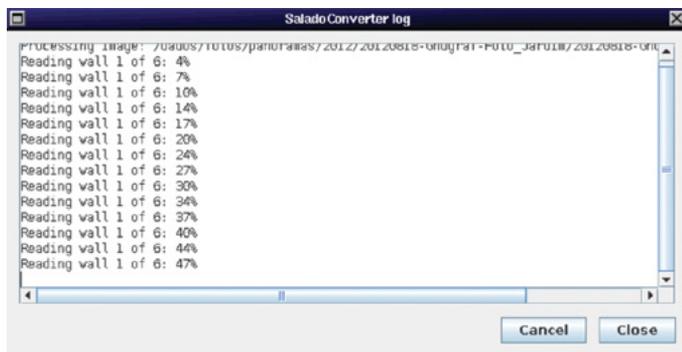


Figure 39. Salado Converter conversion progress

put a background sound, hotspots, video, links to other panoramas and other things. The Salado Player site gives complete information on configuring the XML. After creating the first XML you can then use it as a template for your next panoramas. You can also take a template from my blog, searching for a panorama where I have used Salado Player. Every panorama I publish has a "Technical datasheet" with all tools I have used, besides other information.

Let me try to show it in practice. Download a recent example can be found in this link: <http://cartola.org/panoramas/20120818-Gnugraf-Palestra/index.html>.

You will need this first HTML to point to the XML and other files. After downloading it you must edit it in your favorite text editor (what about vi?) changing:

- the title text between `<title>` and `</title>` tags,
- put the name of your XML file in place of `bn.xml` (or use it as your file name).

Then you will need the XML file. Download it from: <http://cartola.org/panoramas/20120818-Gnugraf-Palestra/bn.xml> and then:

Glossary

Exif

Exchangeable image file format is a standard that specifies the formats for images, sound, and ancillary tags used by digital cameras and other devices. Take a look at the Wikipedia for more detailed information: <http://en.wikipedia.org/wiki/Exif>.

Exposure

It can be understood by the total density of light allowed to fall on the sensor during the taking of a photograph. More informatin here: http://en.wikipedia.org/wiki/Exposure_%28photography%29.

No Parallax Point

Is the point where an imaginary rotation axis passes and one should rotate the camera+lens through this axis in order to obtain no parallax errors between different shoots.

Nadir

Is the direction pointing directly below a particular location. In the case of a full sphere panorama, is the point you see when you look down. In general there are tips and tricks concerning shooting the nadir point to make an full sphere panorama.

Zenith

Is the imaginary point directly above a particular location. In the case of a full sphere panorama, is the point you see when you look up in the vertical direction.

- change the file name to your desired file name,
- edit it by changing:
 - `dz_equi04/equi04_f.xml` to the name of the `*_f.xml` file in your `dz_` directory,
 - erasing the section between `<ImageButton` and `</ImageButton>` tags, which is my logo and link to my site home page.

Notice that this HTML and XML files will expect that you have the Salado Player in the same directory. More specifically, they expect a directory called `SaladoPlayer-1.0` and another called `embed`. Notice also that this sample files are still using version 1.0 of Salado Player. As long as I have tested, changing to a newer version implies on-

ly in changing the path names. I tested the 1.1 version, but at the time of this writing the latest one was 1.3 and has some new features, like dynamic moving lens flare effect.

Another way of doing less configuration is to use always the same image name. The `dz` directory gets the image name as it's suffix and this complete name is what you need to put in the XML file. If you use always the same template and the same image name you can do no configuration at all and put each of your panorama in a different folder to get organized.

I don't have one, but if you want your panoramas to be seen by as many people as possible another thing to do is to make your panorama compatible with apple devices. The free tool I use for this is the VR5 pano viewer [32].

References

1. GIMP: Image Editor – <http://www.gimp.org/>
2. UFRJ – Universidade Federal do Rio de Janeiro, Brazil – <http://www.ufrj.br/>
3. My 360 pictures blog – <http://cartola.org/360>
4. Panoforum – the first open forum in portuguese dedicated to panoramic photography – <http://www.panoforum.com.br/>
5. Google Maps, where you can find Google Street View – <http://maps.google.com/>
6. Hugin tutorial on stitching two pictures together – <http://hugin.sourceforge.net/tutorials/two-photos/en.shtml>
7. Panoramic Tripod Heads – <http://wiki.panotools.org/Heads>
8. Photography misconceptions: nodal point – <http://toothwalker.org/optics/misconceptions.html#m6>
9. Nodal Point – http://en.wikipedia.org/wiki/Nodal_point#Nodal_points
10. Paper about the pivot point – http://web.archive.org/web/20060513074042/http://doug.kerr.home.att.net/pumpkin/Pivot_Point.pdf
11. Panorama Tools Wiki – <http://wiki.panotools.org/>
12. Tripod Heads – <http://wiki.panotools.org/Heads>
13. A simplified self made tripod head model – <http://wp.me/P1AGa0-gS>
14. How to find the No Parallax Point – <http://www.johnhpanos.com/epcalib.htm>
15. NPP/Entrance Pupil Database – http://wiki.panotools.org/Entrance_Pupil_Database
16. Reference page at Cartola's blog – <http://cartola.org/360/en/reference-links/>
17. UFRaw: Raw treatment software – <http://ufraw.sourceforge.net/>
18. Raw Therapee: Raw treatment software – <http://www.rawtherapee.com/>
19. Darktable: Raw treatment software and photography workflow application – <http://www.darktable.org/>
20. Picture organizer – <http://yorba.org/shotwell/>
21. Neatimage: noise reduction tool – <http://www.neatimage.com/>
22. ImageMagick – Convert, Edit, and Compose Images. A very powerful and mature command line tool set to manipulate images – <http://www.imagemagick.org/>
23. Fred's ImageMagick Scripts: ISONOISE – <http://www.fmwconcepts.com/imagemagick/isonoise/index.php>
24. Enfuse: exposure fusion tool – <http://enblend.sourceforge.net/>
25. Hugin stitching software – <http://hugin.sf.net/>
26. Hugin tutorial on creating architectural projection – a good reference for creating vertical lines – <http://hugin.sourceforge.net/tutorials/architectural/>
27. Panini panorama viewer – <https://sourceforge.net/projects/pvqt/>
28. Bruno Postle's Panotools::Script toolkit – <http://search.cpan.org/dist/Panotools-Script/>
29. 360 Cities: immersive panorama gallery to watch and publish – <http://www.360cities.net>
30. TourWrist – yet another site to publish your panoramas – <http://www.tourwrist.com/>
31. Salado Player: a free flash plugin to play a 360x180° image – http://panozona.com/wiki/SaladoPlayer:Quick_start
32. VR5 pano viewer: a free HTML5 code to publish a 360x180° image – <http://www.vrhabitat.com/#vr5>
33. Pannellum: a free HTML5 code to publish a 360x180° image – <https://bitbucket.org/mpetroff/pannellum/>
34. PTViewer: a free java plugin to publish a 360x180° image – <http://webuser.hs-furtwangen.de/~dersch/>
35. Helmut Dersch – pioneer Panoramatoools developer – <http://webuser.hs-furtwangen.de/~dersch/>
36. Salado Converter: converts images to use with Salado Player – <http://panozona.com/wiki/SaladoConverter>
37. Hugin mailing list – a good place to ask for help – <http://groups.google.com/group/hugin-ptx>

To put your image in its format you need the cubic images I mentioned when I talked about post editing. Once you have them you need to convert each cube face to two resolutions, one with 1024x1024 pixels and other with 480x480 pixels. You need to give them the correct names and put them in the correct directory. Inside the file that we download with the software there is a small text file with all the instructions needed. You will have to do something like this:

- download and unpack the vr5 pano viewer,
- take your cube faces and convert them, for example, with the convert command from ImageMagick:
 - supposing you generated the six cube faces with names like: cubic000.tif, cubic0001.tif, cubic0002.tif, cubic0003.tif, cubic0004.tif, cubic0005.tif, generated by nona,

```
foreach i (0 1 2 3 4 5)
convert -resize 1024 cubic000$i.tif 1024_${i}.jpg
convert -resize 480 cubic000$i.tif 480_${i}.jpg
end
```

after these commands you will have the files: 1024_0.jpg, 1024_1.jpg, 1024_2.jpg, 1024_3.jpg, 1024_4.jpg, 1024_5.jpg and 480_0.jpg, 480_1.jpg, 480_2.jpg, 480_3.jpg, 480_4.jpg, 480_5.jpg

- put all those twelve converted files in the directory `vr_files/cube_faces` from the vr5 unpacked files
- put the whole vr5 thing in a subdirectory below your panorama, where you have put the flash version, for example, create a directory called "entrance",
- edit the file "entrance/vr_files/config.js
- put a title in the variable "vrTitle", like: `var vrTitle = 'Palestra V Gnugraf - 18/08/2012';`,
- put an alternative path to be shown in non iThing devices in the variable "flashPage", like: `var flashPage = '../index.html';`
- if you followed the example, creating the index.html for the flash version and putting the vr5 in a subdirectory, then the variable flashPage gets exactly like shown above.

That's all folks! You need now to point your panorama link to the "entrance" folder and the vr5 will detect iThings. If it is being accessed by one of those or not it will know what to do and the panorama will be shown correctly!

Summary

I hope I have been able to show you a brief of the many aspects of the full sphere panoramic photography and

how to make it under a BSD system. As I have mentioned there are plenty of ways to achieve the final result and this is just one approach. That said you can, obviously be free to try other ways, but the big steps will not change very much:

- you will take your pictures,
- you will maybe treat your images,
- you will mount the panorama (stitching it),
- you will make adjustments (or not),
- you will publish it (or not).

Each step has its secrets to success. The one I consider the most critical is the stitching step. It is where, in my opinion, you can get in different difficulties and not have a ready solution. The Hugin mailing list [37] is a very good place to ask for help and I am there also. For the beginner, the stitching of bad images can be hard, so the stitching step depends of well done pictures. I am not talking about beautiful pictures, I am talking about using the No Parallax Point to make them, which, remember, makes a huge difference in the easiness of stitching a full sphere panorama. Also remember the recommendations of a good scene selection.

Much more information and many other concepts can be studied and I hope this article peaks your interest in the subject, making you search for more and become one more panoramist photographer in the world! Particularly in the BSD world and also in all the free software community I hope we can maybe find more people to help the free softwares to evolve and continue growing..

CARLOS E. G., "CARTOLA" CARVALHO

Carlos E. G. Carvalho is electronic engineer, graduated in 1995 in UFRJ [2]. Have worked for around 15 years with Unix, mainly with FreeBSD but also Solaris, Tru64, Linux and others. Experienced in free and open source software, IT infra-structure, network management, system development and project management. Organized the BSDCon Brazil in 2003, created the "BSD em Revista", the first magazine dedicated to BSD in Portuguese and the news site <http://www.myfreebsd.com.br> (not online anymore). Organized and taught in FreeBSD courses. Studied photography at school in 1985. Built amplifiers, tripods, home made cameras and so on. Make panoramic pictures since 2002 and got specialized in 2010. Keep a blog of immersive pictures [3] and the first open forum in Portuguese, the Panoforum [4].

Great Specials

On FreeBSD & PC-BSD Merchandise

Give us a call & ask about our
SOFTWARE BUNDLES
1.925.240.6652

\$39.95

FreeBSD 9.0 Jewel Case CD Set
or FreeBSD 9.0 DVD

\$29.95

PC-BSD 9.0 DVD

\$49.95

The PC-BSD 9.0 Users Handbook
PC-BSD 9.0 DVD

\$99.95

The FreeBSD CD or DVD Bundle

Inside each CD/DVD Bundle, you'll find:
FreeBSD Handbook, 3rd Edition
Users Guide FreeBSD Handbook, 3rd Edition, Admin Guide
FreeBSD 9.0 CD or DVD set
FreeBSD Toolkit DVD



Stylish Dress Attire
Look Your Professional Best



Comfy Hoodies
Stay Warm in Pullovers & Zip Ups

T-Shirts
Lots of Styles to Choose From

FreeBSD 9.0 Jewel Case CD/DVD \$39.95

CD Set Contains:

- **Disc 1:** Installation Boot LiveCD (i386)
- **Disc 2:** Essential Packages Xorg, GNOME2 (i386)
- **Disc 3:** Installation Boot LiveCD (amd64)
- **Disc 4:** Essential Packages Xorg, GNOME2 (amd64)

FreeBSD 8.2 CD \$39.95

FreeBSD 8.2 DVD \$39.95

FreeBSD Subscriptions

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

FreeBSD Subscription, start with CD 9.0 \$29.95

FreeBSD Subscription, start with DVD 9.0 \$29.95

FreeBSD Subscription, start with CD 8.2 \$29.95

FreeBSD Subscription, start with DVD 8.2 \$29.95

PC-BSD 9.0 DVD (Isotope Edition)

PC-BSD 9.0 DVD \$29.95

PC-BSD Subscription \$19.95

The FreeBSD Handbook

The FreeBSD Handbook, Volume 1 (User Guide) \$39.95

The FreeBSD Handbook, Volume 2 (Admin Guide) \$39.95

The FreeBSD Handbook Specials

The FreeBSD Handbook, Volume 2 (Both Volumes) \$59.95

The FreeBSD Handbook, Both Volumes & FreeBSD 9.0 \$79.95

PC-BSD 9.0 Users Handbook \$24.95

BSD Magazine \$11.99

The FreeBSD Toolkit DVD \$39.95

FreeBSD Mousepad \$10.00

FreeBSD & PCBSD Caps \$20.00

BSD Daemon Horns \$2.00



Bundle Specials!
Save \$\$\$

Just Plain Fun
Mousepads & Novelty Horns



BSD Magazine
Available Monthly



For even MORE items
visit our website today!

www.FreeBSDMall.com

FreeBSD Programming Primer (Part 1)

In this new series we will look at the tools, processes and methods involved in writing software, including developing a Content Management System (CMS) which will run under an AMP stack on FreeBSD, OpenBSD, Linux, etc.

What you will learn...

- How to configure a development environment and write HTML, CSS, PHP and SQL code

What you should know...

- BSD and general PC administration skills

Within the I.T. environment there are many disciplines, and often these skill sets work in isolation. The sys-admin doesn't always understand the challenges faced by the programmer or developer, the support engineer doesn't understand the problems of the developer, and the project manager doesn't understand the problems of the technical staff. In this new series, we will examine from first principles how to develop a CMS that will run on any Apache / MySQL / PHP stack. This will involve writing HTML, CSS, PHP and SQL code.

Code is Everywhere

To the uninitiated, writing computer code from scratch may seem a challenge. Certainly, some programming languages are more complex than others, but the fact remains you have already programmed some device at some stage without realizing it even if you have not been near the command line (for example a VHS recorder, central heating timer etc.). As a result you have instructed the device to do something (Record the Simpsons at 10:00PM on Friday evenings). Software is effectively just a collection of instructions, logic and actions like this that allow the computer to interact with another computer, an end user or just itself. The skill is in writing good code that meets the following guiding principles:

- Does "what it says on the tin"
- Is user friendly
- Is secure and reliable under stress
- Is fast and efficient (Don't Repeat Yourself)
- Is easily modified and extended
- Can be easily understood
- Has documentation

While some of these points are essential to any piece of software, some may be more important than others depending on the operating environment and specification. For instance, a piece of code that pulls pages from a website on a daily basis into a new directory in the format day_month_year (like 01_01_2013, 02_01_2013 etc.) for later reading by a technician would not necessarily require anything other than a log file entry saying "404 Not Found" if no content was available. However, if this was a critical program designed for an end user, it would be better practice to raise a friendly error message e.g. "The page you requested was not found. Please try again later or contact the helpdesk on 123 456789".

Software writing should be creative and enjoyable, and part of the challenge is to have a reasonable idea of what you want to achieve beforehand, who your audience is, what limitations you must consider, and the environment the software will run under. A good functional

specification should cover these details, but it is important to realize that software is never really finished. More functionality may be required, the environment may change, or bugs and faults need to be rectified in the program. That is why code should be easily modified and understood as it is the programmers worst nightmare having to maintain a badly written, undocumented, broken program. Trying to get inside someone else's logic especially when under pressure to meet deadlines can be very stressful!

Computers Are Not Very Clever

The old adage “Garbage In = Garbage Out” is most applicable in the area of programming. As CANVC, they can only literally interpret any instructions that they receive. For instance, you might think you have asked the program to print the date, but due to an error in your logic, it might return 01-01-1970, NULL, or UNDEF. It might not even return anything at all. Sometimes when writing code you will be convinced the computer is your enemy. This is where defensive programming and debugging come to the fore, by re-thinking the obvious (and not so obvious) assumptions such as “All input data is valid”. The defensive programmer would respond by saying “All data is important and tainted unless proved otherwise”. Expect the unexpected. Sometimes it is best to walk away, take a break and return to the problem later. Late night coding sessions can be frustrating, especially if the result is not what is expected. Trying to debug an issue without a decent IDE (Integrated Development Environment) is possible, but time consuming.

Choosing the Language

Not all programming languages are equal, and some are less equal than others. Different languages are geared towards different tasks.

Shell programming languages (for example Bash, Sh etc.) are great for system administration tasks e.g. clearing out and archiving directories, running commands depending on the user response etc. However they are not fully fledged programming languages as such.

BASIC and Pascal are great for learning how to code, but they have some limitations. While it would be possible to write a CMS in either of them, as they are not primarily geared towards the web the program would be complex and convoluted.

The same argument applies to C. C is extremely powerful and flexible and PHP, Apache and MySQL are written using it. It would be complete overkill to write the CMS in scratch from C as we would effectively have to re-invent the wheel.

Java would make a great platform for a CMS due in part to its extensive library support and security, but as it is object orientated rather than procedural, the code and underlying principles would be more complex.

Script based languages (for example Ruby, Perl, Python, PHP) are geared towards the Internet, and most ISP's will support them. As PHP has good support, is very portable, the documentation is excellent, and integrates well with both Apache (Our web server software) and MySQL (our database) it is a strong choice. While the other script languages are just as suitable for our CMS, the author has more experience with PHP so that is the reason for the choice.

SQL, HTML and CSS are different types of language. While not considered “real” programming languages as such (on their own you could not write a software application) they are essential to our CMS.

SQL (*Sequential Query Language*) is the de facto standard language of databases. While most databases today use some form of SQL to extract, view and alter data, the “dialect” differs from database to database. We will use SQL to fetch our dynamic content from our database.

HTML (*Hyper Text Markup Language*) is the language of the web page. Each document has separate elements e.g. a body, header, images etc. and the HTML standard defines what these elements are. HTML pages are served by Apache and interpreted by the client browser e.g. Firefox.

CSS (*Cascading Style sheets*) are used in conjunction with HTML to change the style of the raw HTML pages. While it would be possible to write a CMS without it, it would probably not be very aesthetic.

JavaScript is a lightweight programming language used for dynamic tasks in conjunction with HTML e.g. changing content on the fly. It is run seamlessly from the client browser.

Generally, programming languages fall into 2 categories, compiled and interpreted. For instance C, Basic and Pascal are compiled whereas most script languages are interpreted. The major difference between compiled and interpreted languages is how the program itself is accessed and run. In the compiled scenario, the initial source code is passed through a compiler which generates a stand-alone binary if the source code is valid. The operating system then handles the corresponding output. A binary compiled for one particular Operating System will not run on another – in general the compiler has to match the O/S unless some form of emulation and library support is available. With interpreted languages each line of the source code is passed through the interpreter which handles the corresponding output. Both language types sup-

port additional libraries which extend the core functionality of the language (e.g. graph support) and these are used as required. See Figure 1 and Figure 2 – Compiled and Interpreted languages.

The bottom line is that you need to choose your language for the task you have in hand. Some all purpose languages are great but you need to remember the limitations. The author often uses PHP for add-hoc scripts, but Perl or Bash would be just as effective. Often it is a case of what you feel most comfortable with, but at the same time you don't want to fall into the trap "When the only tool you have is a hammer every problem is a nail".

To err is Human

Writing code is paradoxically both infinitely creative and flexible yet structured and pedantic. One missed semicolon, a full stop in the wrong place, even word case can be the difference between a working code segment and an esoteric error message. Sometimes by fixing one problem other problems are introduced, sometimes the real problem was never addressed at all. It is important that we are

able to snapshot and document our changes as well as quickly isolate any problems. As part of the series we will look at version control and debugging.

The Draft Specification

The initial specification of our CMS is per Table 1. Further additions may be made over the series to demonstrate specific principles. The inspiration for parts of the specification came from the excellent CMS, Drupal by Dries Buytaert.

Testing

It is critical that any application is properly tested before release. While automated testing methods are available, for the purpose of this series will limit testing to some crude load and security testing and ensuring that the program "just works as advertised".

The Development Environment

In a commercial environment, the bare minimum would probably consist of a test (development) server, a live

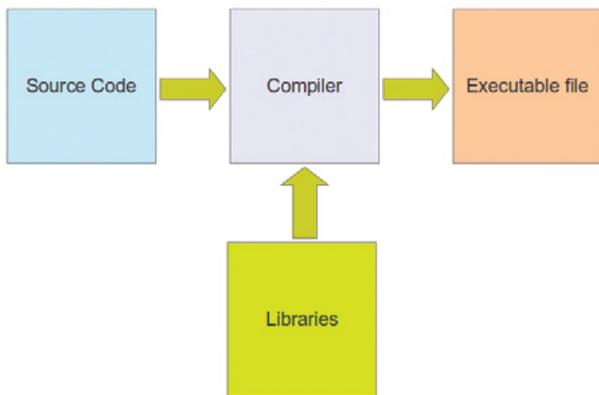


Figure 1. A compiled program

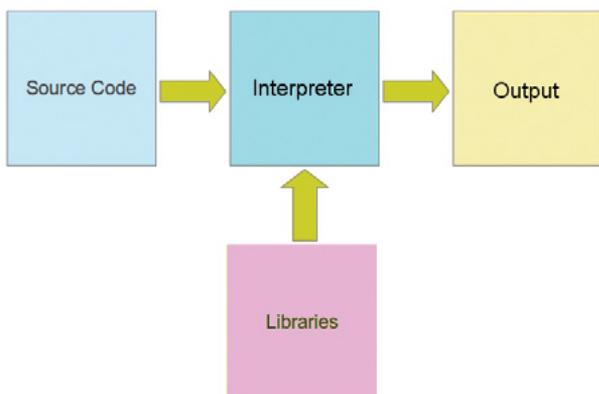


Figure 1. An interpreted script

Table 1. CMS draft specification

Initial CMS Specification
Allow an authorised user to create a W3C valid web page
Database transactions (MySQL InnoDB storage engine)
Efficient search
Image and attachment uploads
Menu module
Modular and extensible
Run under a standard AMP stack with little modification
Support XHTML 1.0 strict
Taxonomy
Template and region driven – separate the rendering logic from page content
Visitor statistics

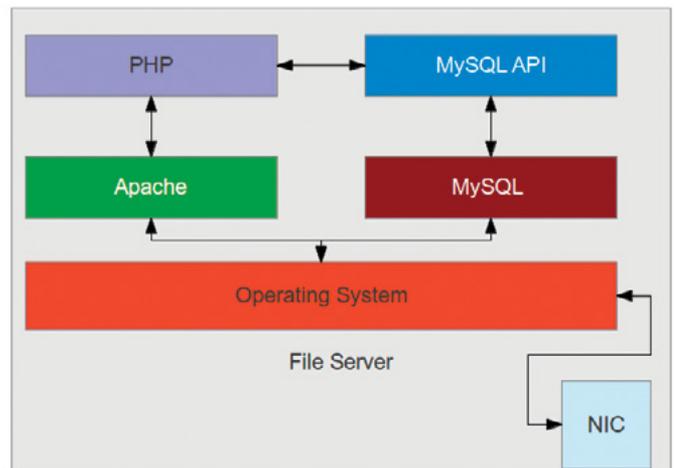


Figure 3. Our CMS architecture

(production) server, a Version Control Server (VCS), possibly a database server (MySQL) and the developers workstation with an Integrated Development Environment (IDE) for code development, syntax checking and debugging. Source code would be pulled from the VCS, edited and tested on either the workstation or the development server, committed to VCS and pushed to the production server for access by the users when stable and ready for release. This scenario is too complex for our series, but while it is possible to develop just from the command line, debugging (and certainly testing) will be close to impossible outside of a graphical environment. As a very bare minimum, you will need a headless FreeBSD box (without any GUI) and some sort of workstation with Firefox installed, but ideally your BSD development box should support Firefox, Netbeans, Apache, PHP, GIT and MySQL. Your favorite CLI editor can of course still be used for editing.

In the Next Article

We will start programming in earnest and start serving our first CMS page.

ROB SOMERVILLE

Rob Somerville has been passionate about technology since his early teens. A keen advocate of open systems since the mid eighties, he has worked in many corporate sectors including finance, automotive, airlines, government and media in a variety of roles from technical support, system administrator, developer, systems integrator and IT manager. He has moved on from CP/M and nixie tubes but keeps a soldering iron handy just in case.

The BSD Certification Group Inc. (BSDCG) is a non-profit organization committed to creating and maintaining a global certification standard for system administration on BSD based operating systems.

? WHAT CERTIFICATIONS ARE AVAILABLE?

BSDA: Entry-level certification suited for candidates with a general Unix background and at least six months of experience with BSD systems.

BDSP: Advanced certification for senior system administrators with at least three years of experience on BSD systems. Successful BDSP candidates are able to demonstrate strong to expert skills in BSD Unix system administration.

✓ WHERE CAN I GET CERTIFIED?

We're pleased to announce that after 7 months of negotiations and the work required to make the exam available in a computer based format, that the BSDA exam is now available at several hundred testing centers around the world. Paper based BSDA exams cost \$75 USD. Computer based BSDA exams cost \$150 USD. The price of the BDSP exams are yet to be determined.

Payments are made through our registration website:
<https://register.bsdcertification.org/register/payment>

i WHERE CAN I GET MORE INFORMATION?

More information and links to our mailing lists, LinkedIn groups, and Facebook group are available at our website:
<http://www.bsdcertification.org>

Registration for upcoming exam events is available at our registration website:
<https://register.bsdcertification.org/register/get-a-bsdcg-id>



Register Early and SAVE!



The Best SharePoint Training!

Choose from over
90 Classes & Workshops!

Check out these **NEW!** classes,
taught by the industry's best experts!



How to Install SharePoint 2013 Without
Screwing It Up
Todd Klindt  and Shane Young 

Creating Simple Dashboards Using
Out-of-the-Box Web Parts
Jennifer Mason 

What IS SharePoint Development?
Mark Rackley

Integrating SharePoint 2010 and Visual
Studio Lightswitch
Rob Windsor 

SharePoint Performance: Best Practices
from the Field
Jason Himmelstein

Solving Enterprise Search Challenges with
SharePoint 2010
Matthew McDermott 

Creating a Great User Experience in
SharePoint
Marc Anderson 

Getting Stuff Done! Managing Tasks with
SharePoint Designer Workflows
Chris Beckett 

Ten Best SharePoint Features You've
Never Used
Christian Buckley 

SharePoint 2013 Upgrade Planning for
the End User: What You Need to Know
Richard Harbridge

Understanding and Implementing
Governance for SharePoint 2010
Bill English

Ten Non-SharePoint Technical Issues
That Can Doom Your Implementation
Robert Bogue 

Building Apps for SharePoint 2013
Andrew Connell 

SharePoint MoneyBall: The Art of Winning
the SharePoint Metrics Game
Susan Hanley

SharePoint Solutions with SPServices
Marc Anderson 

Lists: Used, Abused and Underappreciated
Wes Preston 

Intro to Branding SharePoint 2010 in the
Farm and Online
Randy Drisgill  and John Ross 

Planning and Configuring Extranets in
SharePoint 2010
Geoff Varosky

How to Best Develop Requirements for
SharePoint Projects
Dux Raymond Sy 



Check out more than
55 exhibiting companies!

Big Data gets real at Big Data TechCon!

The **HOW-TO** conference for Big Data and IT Professionals



Discover how to master Big Data from real-world practitioners – instructors who work in the trenches and can teach you from real-world experience!

Come to Big Data TechCon to learn the best ways to:

- Collect, sort and store massive quantities of structured and unstructured data
- Process real-time data pouring into your organization
- Master Big Data tools and technologies like Hadoop, Map/Reduce, NoSQL databases, and more
- Learn HOW TO integrate data-collection technologies with analysis and business-analysis tools to produce the kind of workable information and reports your organization needs
- Understand HOW TO leverage Big Data to help your organization today

Over 50
how-to
practical classes
and workshops
to choose
from!

BigData
TECHCON

April 8-10, 2013

Boston, MA

www.BigDataTechCon.com

Register Early and SAVE!

A BZ Media Event

Big Data TechCon™ is a trademark of BZ Media LLC.



Headquarters:
San Jose, CA



855.GREP.4.IX | Contact Us

99% Compatibility

online now...

IXSYSTEMS AND YOU ARE
THE PERFECT MATCH



SHARED INTERESTS

- Enterprise Storage Solutions
- Personalized Customer Service
- Bold New Information Technology

I'm a

In

Looking for

- A Technology Partner
- More Technical Experience
- New Business Opportunities

Visit Today!



iXsystems

Technology Partner Seeking
Resellers/Integrators for
TrueNAS™ Storage Appliance



WWW.IXSYSTEMS.COM/PERFECTMATCH

