

BSD

FOR NOVICE AND ADVANCED USERS

BSD as Operating System

INSIDE

THE FREEBSD UBUNTU CHALLENGE
NETWORK MONITORING WITH NAGIOS AND OPENBSD (PART 1)
MAINTENANCE SYSTEMS OVER BSD
LOW RESOURCE PCS WITH FREEBSD
MAKING THE UNKNOWN GIANT VISIBLE AND KNOWN

EXCLUSIVELY

▶ INTRODUCTION TO MIDNIGHT BSD

VOL.3 NO.8
ISSUE 8/2010(14)
1898-9144



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



✓ Increased Performance ✓ Impressive Energy Savings

iX-Triton TwinBlade Servers: The Easy-to-Manage, Greener Way to Serve

› AFFORDABLE › ECONOMICAL › SAVINGS



Greenest, most energy-efficient
blade server in the industry!

The new Triton TwinBlade Server is the most technologically advanced blade server system in the industry, and the ideal solution for power-efficiency, density, and ease of management.

The Triton TwinBlade Server supports up to 120 DP servers with 240 Intel® Xeon® 5600/5500 series processors per 42U rack, achieving an unmatched 0.35U per DP node. Up to two 4x QDR (40 Gbps) Infiniband switches, 10GbE switches or pass-through modules give the TwinBlade the bandwidth to support the most demanding applications.

With N+1 redundant, high efficiency (94%) 2500W power supplies, the TwinBlade is the Greenest, most energy-efficient blade server in the industry. The

energy saved by the iX-Triton TwinBlade Server will keep the environment cleaner and greener, while leaving the green in your bank account.

Server management is also simple with the Triton Twin Blade Server.

Remote access is available through SOL (Serial Over Lan), KVM, and KVM over IP technologies. A separate controller processor allows all of the Triton's remote management and monitoring to function regardless of system failures, offering true Lights Out Management.

Using the Triton's management system, administrators can remotely control TwinBlades, power supplies, cooling fans, and networking switches. Users may control the power remotely to reboot and reset the Triton TwinBlade Center and individual Twin Blades, and may also monitor temperatures, power status, fan speeds, and voltage.

For more information on the **iX-Triton TwinBlade**, or to request a quote, visit:

<http://www.ixsystems.com/tritontwinblade>

20 Server Compute Nodes in 7U of Rack Space

The iX-TB4X2 chassis holds 10 TwinBlade servers and each TwinBlade supports two nodes. This gives the iX-TB4X2 chassis the ability to house 20 nodes in 7U of rack space. The powerful Triton TwinBlade achieves 0.35U per dual-processor node, and is twice as dense as the previous generation of dual-processor blades.

A fully-loaded iX-Triton TwinBlade supports 40 Intel® Xeon® 5600/5500 series processors and up to 2.5 TB DDR 1333/1066/800MHz ECC Registered DIMM memory. In a 42U rack this translates into 120 nodes with 240 Intel® Xeon® 5600/5500 series processors and 15 TB DDR 1333/1066/800MHz ECC Registered DIMM memory.



- ▶ By replacing 1U servers with TwinBlade servers, the power savings of the iX-TB4X2 can reach more than \$1000* per year, per server with reduced cooling costs added in.



- ▶ Replacing 1U rackmount servers with an iX-TB4X2 Twin Blade can reduce carbon dioxide emissions by over 5.5 metric tons.**



- ▶ The iX-Triton TwinBlade delivers the most energy-efficient blade server in the industry with four N+1 redundant, high efficiency (94%) 2500W power supplies.

* Electricity costs vary by location.

** According to Energy Information Agency (a statistical agency of the U.S. Department of Energy), saving one kilowatt hour of electricity reduces carbon dioxide emissions by 1.43 pounds.



Call iXsystems toll free or visit our website today!
+1-800-820-BSDi | www.iXsystems.com

Key features:

- Up to 10 dual-node TwinBlades in a 7U Chassis, 6 Chassis per 42U rack
- Remotely manage and monitor TwinBlades, power supplies, cooling fans, and networking switches
- Hardware Health Monitor
- Virtual Media Over Lan (Virtual USB, Floppy/CD, and Drive Redirection)
- Integrated IPMI 2.0 w/ remote KVM over LAN/IP
- Remote Power Control
- Supports one hot-plug management module providing remote KVM and IPMI 2.0 functionalities
- Up to four N+1 redundant, hot-swap 2500W power supplies
- Up to 16 cooling fans

Each of the TwinBlade's two nodes features:

- Intel® Xeon® processor 5600/5500 series, with QPI up to 6.4 GT/s
- Intel® 5500 Chipset
- Up to 128GB DDR3 1333/ 1066/ 800MHz ECC Registered DIMM / 32GB Unbuffered DIMM
- Intel® 82576 Dual-Port Gigabit Ethernet
- 2 x 2.5" Hot-Plug SATA Drive Trays
- Integrated Matrox G200eW Graphics
- Mellanox ConnectX QDR InfiniBand 40Gbps or 10GbE support (Optional)



Dear Readers!

I am happy to introduce you August issue.

This time we will be mentioning Windows, Ubuntu in our magazine, but surely it will be more than connected to BSD.

Read it and let us know if it was usefull and interesting. :)

We also have modified and have another survey for you, please find some time to fill it in.

At the moment we are planning to open russian version of BSD Magazine in September.

The magazine will be also free online publication.

And we are looking for authors, betatesters and proofreaders with russian as native language.

Please contact olga.kartseva@bsdmag.org in case you want to contibute or have an idea where we should announce this news.

Please spread the word about it on your blogs, forums, websites!

Thank you!

*Olga Kartseva
Editor in Chief
olga.kartseva@software.com.pl*



MAGAZINE BSD

Editor in Chief:

Olga Kartseva
olga.kartseva@software.com.pl

Contributing:

Rob Somerville, Daniele Mazzocchio, Rashid N. Achilov, Joseba Mendez, Laura Michaels
Lukas Holt, Caryn Holt, Laura Michaels

Special thanks to:

Marko Milenovic, Worth Bishop and Mike Bybee

Art Director:

Ireneusz Pogroszewski

DTP:

Ireneusz Pogroszewski

Senior Consultant/Publisher:

Paweł Marciniak pawel@software.com.pl

National Sales Manager:

Ewa Łozowicka
ewa.lozowicka@software.com.pl

Marketing Director:

Ewa Łozowicka
ewa.lozowicka@software.com.pl

Executive Ad Consultant:

Karolina Lesińska
karolina.lesińska@bsdmag.org

Advertising Sales:

Olga Kartseva
olga.kartseva@software.com.pl

Publisher :

Software Press Sp. z o.o. SK
ul. Bokserska 1, 02-682 Warszawa
Poland

worldwide publishing
tel: 1 917 338 36 31
www.bsdmag.org

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

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.

The editors use automatic DTP system **AOPUS**

Mathematical formulas created by Design Science MathType™.

GET STARTED

06 Introduction to MidnightBSD

Lukas Holt, Caryn Holt

MidnightBSD was founded in 2006 by Lucas Holt.

The project is a FreeBSD 6.0 fork with an emphasis on creating a desktop focused BSD.

While there are other BSD desktop projects (most notably PC-BSD and DesktopBSD),

we wanted to create an entire desktop centered BSD from the kernel all the way up to the standard applications. We want a BSD that a grandmother could install and use.

HOW TO'S

12 The FreeBSD Ubuntu challenge

Rob Somerville

FreeBSD makes a great server, but can it rise to the challenge of running Compiz as a workstation?

One of the many criticisms of Open Source software (indeed even FreeBSD) is that it is not ready for the desktop.

18 Network monitoring with Nagios and OpenBSD (PART 1)

Daniele Mazzocchio

So our OpenBSD-based network now includes redundant firewalls, domain name servers, a mail gateway and a web proxy cache. (Read previous issues of BSD Magazine) All the services provided by these machines are particularly critical and can't afford even minimal downtime.

Redundancy may give us the time to recover a failure before having angry users trying to knock down our door, but it doesn't free us from the responsibility to detect and solve ongoing problems.

36 Replacing Microsoft Exchange Server

Rashid N. Achilov

Installing set of open-source programs without lack of functionality Instead of Microsoft Exchange Server. This way Groupware-part will be replaced on Horde Groupware.

LET'S TALK

42 Maintenance Systems over BSD

Joseba Mendez

I was talking in previous articles about how to run applications widely used in the Industry that can be supported by BSD apart of classical IT services.

As clear example of this is SAP Suite. SAP covers all possible asset management to control the cost related to production and also maintenance but as per tighted cost in investments today, the Plants must run 24/7 with maximum reliability and productivity possible.

46 Low Resource PCs with FreeBSD

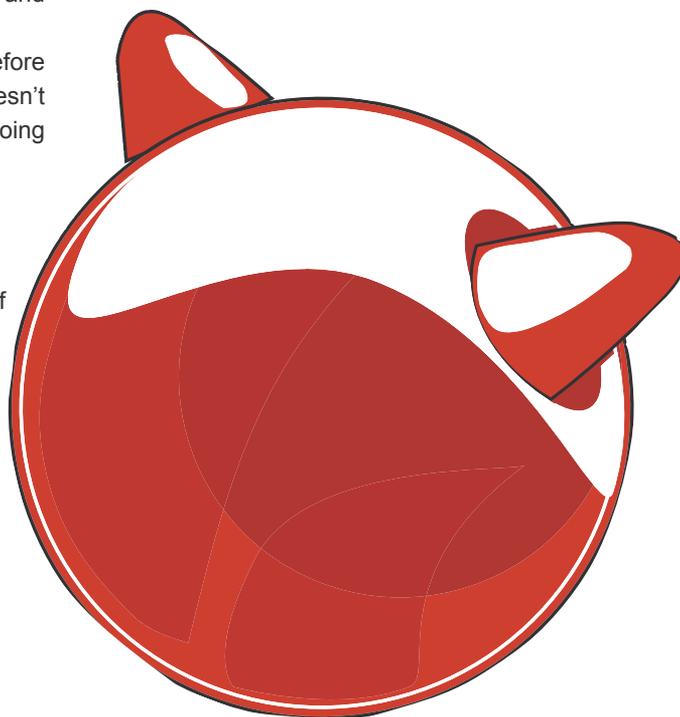
Laura Michaels

FreeBSD is my pick for best modern operating system to use on older PCs. I can't believe how many used PCs end up as landfill while students, educators, low income families and others go without a computer at all.

50 Making the Unknown Giant Visible and Known

Joshua Ebarvia

Making the Unknown Giant Visible and Known FreeBSD has the moniker Unknown Giant. I confirm that it is true in my place. I have asked system administrators, computer enthusiasts, and hobbyist about FreeBSD and they didn't even know what I'm talking about.



An Introduction

to MidnightBSD

While there are other BSD desktop projects (most notably PC-BSD and DesktopBSD), we wanted to create an entire desktop centered BSD from the kernel all the way up to the standard applications. We want a BSD that a grandmother could install and use.



What you will learn...

- what is MidnightBSD : installation, booting, mports

What you should know...

- how to use computer

MidnightBSD History

MidnightBSD was founded in 2006 by Lucas Holt. The project is a FreeBSD 6.0 fork with an emphasis on creating a desktop focused BSD.

While there are other BSD desktop projects (most notably PC-BSD and DesktopBSD), we wanted to create an entire desktop centered BSD from the kernel all the way up to the standard applications. We want a BSD that a grandmother could install and use.

The current development focus is on creating a solid foundation. We are working on further developing the mports system, creating a new package management system, improving the MidnightBSD build cluster and implementing an easy-to-use installer.

System Requirements

A computer with an Intel Pentium or equivalent CPU and 64MB RAM is required to install MidnightBSD. For optimal desktop use, we recommend at least 256MB of RAM and a 686 class CPU.

Getting MidnightBSD

MidnightBSD is available for i386 and amd64 architectures. The latest release is 0.2.1 with 0.3 under active development. Developer snapshots are available for version 0.3 on i386 and amd64; those familiar with BSD development or who like to experiment with newer technologies may wish to try the developer snapshots. Most users will have a better experience with 0.2.1-RELEASE.



Figure 1. A selection screen

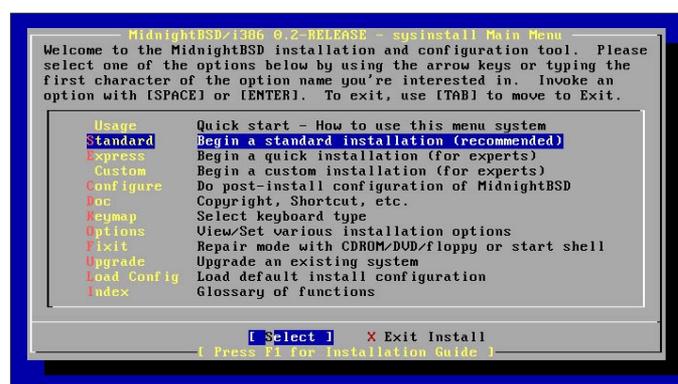


Figure 2. The sysinstall installation program

ISO images may be obtained from <http://www.midnightbsd.org/download/>. If one wishes to use a graphical desktop environment such as KDE or WindowMaker & GNUstep, download three files 0.2.1-RELEASE-i386-disc1.iso, 0.2.1-RELEASE-i386-disc2.iso, and 0.2.1-RELEASE-i386-disc3.iso. Disk two and three contain packages and are not needed for a basic installation. Burn these files to CDs using your favorite program.

Trying MidnightBSD

A live CD is available on the project wiki at <http://www.midnightbsd.org/wiki/livecd>. Using a live CD can help you determine if you wish to install MidnightBSD on your computer as well as test for hardware compatibility.

Installing MidnightBSD

Place the MidnightBSD disc1 CD into your computer and start the system. You will be presented with a selection screen similar to the following: see Figure 1.

You may wait or press enter. If you have an older system and experience difficulties booting MidnightBSD, try option 3 *Boot in Safe Mode*. After a few moments, you will be presented with the sysinstall installation program see Figure 2.



Figure 3. Choose OK to continue to fdisk see

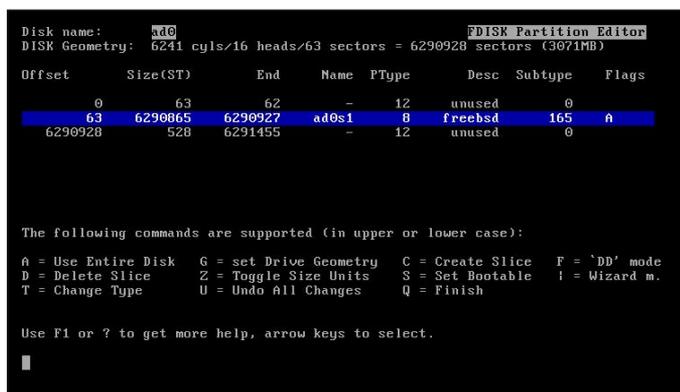


Figure 4. This will allocate your entire hard disk to MidnightBSD

You may wait or press enter. Select *Standard* from the menu by pressing down and then press Enter on your keyboard. You will then be presented with an informational box explaining the fdisk process. Choose OK to continue to fdisk see Figure 3. For this article, it is assumed that you wish to install MidnightBSD as the only operating system on your computer or virtual machine. Installing MidnightBSD along with another system requires free disk space for MidnightBSD to use on the hard drive and a boot manager. A simple boot manager is included with MidnightBSD, but we recommend using a third party manager called GAG as it is compatible with Windows 7.

To setup your hard disk in fdisk, select a for *Use Entire Disk*, arrow down to the newly created *freebsd* type, press S for *Set Bootable* and Q for *Finish*. This will allocate your entire hard disk to MidnightBSD see Figure 4.

Next, you will select the boot manager. If you will be using MidnightBSD on the entire drive or with GAG, arrow down and select *Standard*. When sharing the drive with Windows XP or another OS, use *BootMgr* see Figure 5.

Another informational message will pop-up. Select OK and continue to the DiskLabel Editor. Most users will be able to use the defaults. Select A for *Auto Defaults* and then Q to *Finish* see Figure 6.

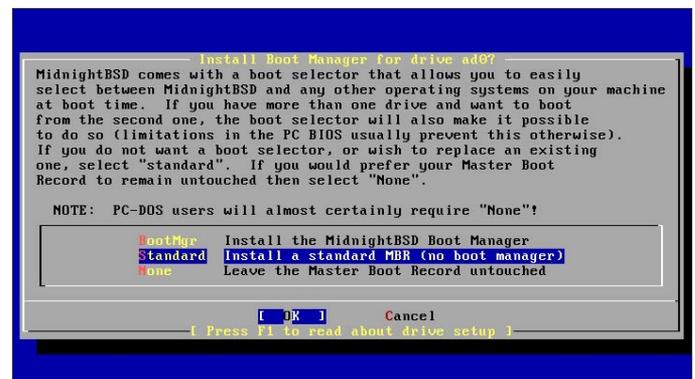


Figure 5. When sharing the drive with Windows XP or another OS, use BootMgr

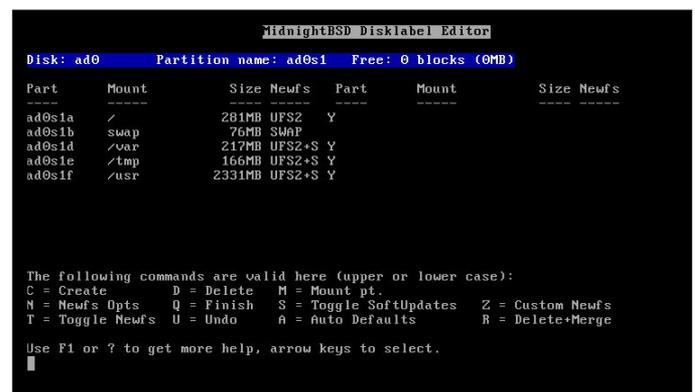


Figure 6. Select A for Auto Defaults and then Q to Finish see

You will be presented with the *Choose Distributions* screen. Select *All* which includes source code needed by some mports to compile kernel modules. With a developer snapshot, select *Exit* here see Figure 7.

You will see a new screen asking about the MidnightBSD Ports Collection, choose *YES* so that you may choose from over 2,000 additional ported applications see Figure 8.

Select *Exit* on the screen to continue the install. Next, you will select the CD-ROM method of installation. It is also possible to install over FTP which is useful when you have difficulty with CD-ROM detection; FTP install works around bugs with Microsoft Virtual PC 7 for Mac OS for example see Figure 9.

Finally, you will be presented with a screen asking you if you want to continue. Choose *YES* to begin installation. This process may take a few minutes to over an hour to copy the core system files over depending on the computer and selected packages. Your hard disk will be partitioned and formatted during this process. If you choose *NO*, you will abort changes to your system. You will be asked to swap CDs during this process several times see Figure 10.

You will be asked to swap CDs during this process several times. MidnightBSD core files have been installed.

Now you will be asked for several system configuration options and given the option to install additional software see Figure 11.

You will be given the option to setup ethernet. Select *YES* and continue into the ethernet configuration screen. You will see at least two options: `s10` and `ppp0`. Any additional options will be network interfaces such as `em0`, `fxp0`, `re0`, `rl0`, and so on. For those unfamiliar with BSD, drivers for network cards tend to be named differently rather than a convention like `eth0` used in Mac OS X and Linux see Figure 12.

Most users will want to skip IPV6 configuration as home networks tend to be IPV4. It is possible to setup an IPV6 tunnel in MidnightBSD using a service such as `sixxs.net`. It is recommended that you select DHCP configuration as home routers tend to provide this feature. Your network cable should be plugged in before trying DHCP configuration.

Type in a host name and make any necessary changes. Then select *OK* see Figure 13.

At this point, you will be asked a few more questions such as *Do you want this machine to function as a network gateway?* Select *no*. *Do you want to configure inetd and the network services it provides?* Select *no*. *Would you*

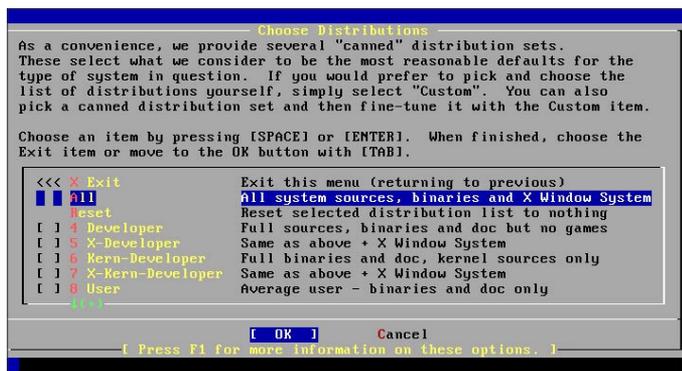


Figure 7. With a developer snapshot, select *Exit* here see Figure 7



Figure 8. A new screen asking about the MidnightBSD Ports Collection

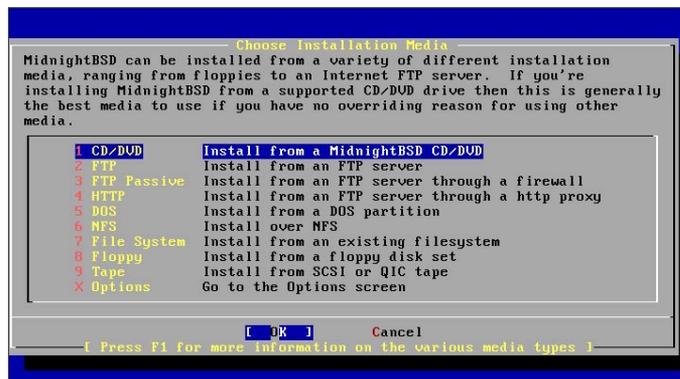


Figure 9. FTP install works around bugs with Microsoft Virtual PC 7 for Mac OS

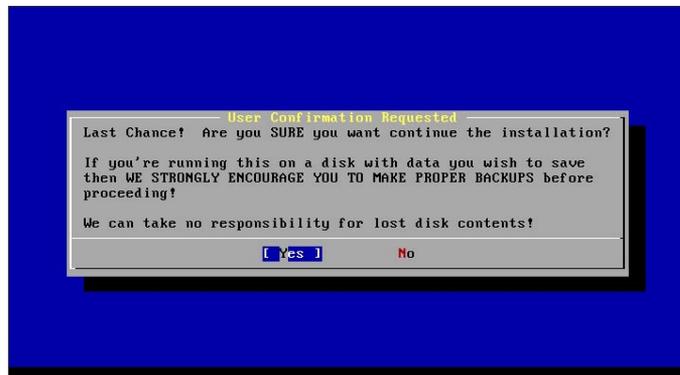


Figure 10. You will be asked to swap CDs during this process several times

like to enable SSH login? I often enable this feature, but it depends on your requirements. Enabling SSH will allow others to try to login to your system while on the Internet. It is not recommended that you run an anonymous FTP server or nfs related services. Select no to those questions. You may also select no on the system console settings unless you need to change your localization or keyboard setup.

Be sure to select the time zone. Most desktop PCs are not set to UTC time. In my case, I select North America, United States, and Eastern Time – Michigan.

It is recommended that you enable linux binary compatibility. It will allow you to run older linux applications such as Mozilla Firefox and Adobe Flash. You may also install games such as Enemy Territory. In MidnightBSD 0.2.1, we have support for Linux 2.4 kernel libraries and use Fedora Core 4. The development version of MidnightBSD supports Linux 2.6 emulation.

Most older systems have a PS/2 mouse, so select yes to this question if you have a mouse. You will be given an opportunity to test the moues. Once you feel comfortable with the settings, select exit to continue on to the package selection screen.

The package selection screen allows you to browse packages included on disk one, two and three. If you

You will find here:

- materials for articles-listings, additional documentation, tools
- the most interesting articles to download
- current information on the upcoming issue

www.bsdmag.org



Figure 11. For several system configuration options and the option to install additional software

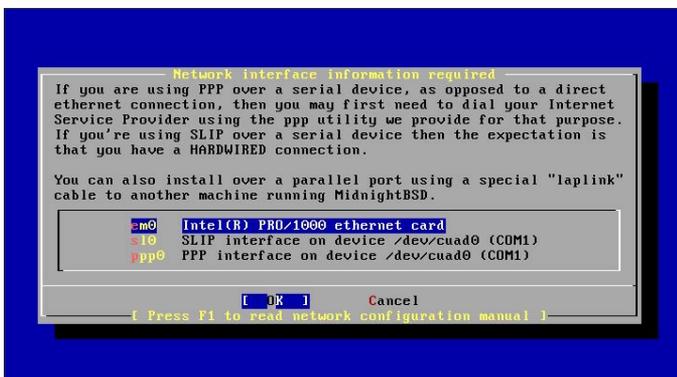


Figure 12. Network cards names

downloaded all three CDs, you can select various software packages such as KDE. You may also skip this step if you did not download the CDs or prefer to fetch the packages over the Internet. The first time the system boots up, a shell script is run that allows you to configure your graphical login settings. This will also fetch software packages from the MidnightBSD FTP server as necessary. Prepare to switch between CDs several times if you select a large number of software packages.

Next you will be presented with an initial user account screen. Create user accounts for each person who will use the system. This version of MidnightBSD relies on the root account at first, but you may setup sudo later. Remember to add the user to the wheel group if you want them to be able to use su or sudo. I recommend a different shell such as `/bin/tcsh` or `/bin/mksh` for most users. After the user creation step, you will need to type in a root password.

Finally, you will be asked if there are any remaining configuration changes. Select no and you will end up at the original sysinstall screen. Select *Exit Install* and be sure to remove the CD from the optical drive.

First Boot

The system will reboot into MidnightBSD and proceed to run the firstboot script. It will ask a few questions. You can choose to report your install using `bsdstats`. This will let us know how many people are using MidnightBSD.

Next, you will be asked if you wish to install a graphical environment. Type yes to install GNUstep and WindowMaker. KDE 3.5 is available on the CD as packages.

If you make any mistakes during the first boot script, you may delete the file `/etc/fbreceipt` and run `/etc/rc.d/firstboot start`.

You may temporarily disable the firewall to help with FTP issues behind NAT by using the command `ipfw disable firewall`.

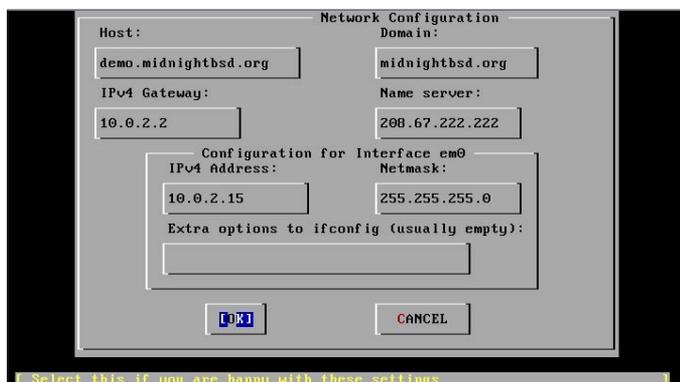


Figure 13. Type in a host name and make any necessary changes, then select OK

Other Considerations

MidnightBSD includes a script to automatically detect and load sound drivers on system startup. Occasionally, it does not work for a particular sound card. Try running `kldload sound` as root to test your sound card. You can make adjustments to the firewall rules in `/etc/rc.firewall`.

Documentation for the system is available on <http://www.midnightbsd.org/> and <http://www.midnightbsd.org/wiki/>. Help is available on IRC (`irc.freenode.net` `#midnightbsd`) and our mailing lists.

Install software by using the `pkg_add` tool as follows: `pkg_add -r program_name` to fetch it from the FTP server.

Mports

MidnightBSD includes a ports system called mports. An older snapshot of mports is included in MidnightBSD 0.2.1 and directions for getting a newer copy are available on the wiki using cvs or cvsup. mports differ from FreeBSD ports in several ways including the fake system which allows us to install into a temporary directory and then create a package. Every time you install a port, it is from a package. It allows us to find bugs in package generation and makes it easier for users to distribute packages. In the future, the `pkg_add` tool will be replaced by a sophisticated package management system called mport which is similar in functionality to tools found in the Linux community like yum and apt-get. Mport tools are available for testing on MidnightBSD 0.3 and rely on sqlite 3 databases for meta data.

LUCAS HOLT

Lucas Holt is a software engineer at PRIME Research; working on large scale data collection and processing applications. He's worked with BSD since 2000, starting on NetBSD with a Sun SparcStation IPC.

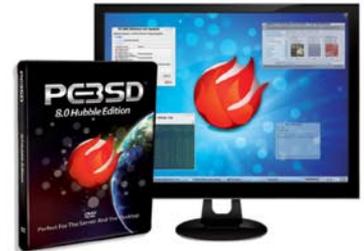
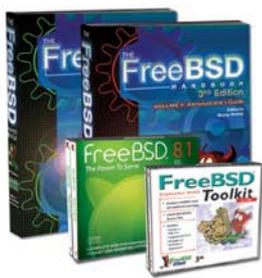
CARYN HOLT

Caryn Holt is a software engineer at Rovi Corporation. She is currently working on graphical versions of the mport tools for the MidnightBSD project.



FreeBSD
Mall

Your FreeBSD &
PC-BSD Resource
www.FreeBSDMall.com



FreeBSD 8.1 Jewel Case CD/DVD

Set contains:

- Disc 1: Installation & Live File System (for system recovery)
- Disc 2: Packages and Documentation
- Disc 3: Additional Packages
- Disc 4: More Packages

FreeBSD 8.1 CD	\$39.95
FreeBSD 8.1 DVD	\$39.95
FreeBSD 7.3 CDROM	\$39.95
FreeBSD 7.3 DVD	\$39.95

FreeBSD Subscriptions

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

FreeBSD Subscription , start with CD 8.1	\$29.95
FreeBSD Subscription, start with DVD 8.1	\$29.95
FreeBSD Subscription, CD 7.3	\$29.95
FreeBSD Subscription, DVD 7.3	\$29.95

PC-BSD 8 DVD (Hubble Edition)

PC-BSD 8 DVD	\$29.95
PC-BSD Subscription	\$19.95

BSD Magazine

BSD Magazine	\$11.99
BSD Magazine Subscription	\$11.99

The FreeBSD Handbook

The FreeBSD Handbook, Volume 1 (User Guide)	\$39.95
The FreeBSD Handbook, Volume 2 (Admin Guide)	\$39.95
★ Special: The FreeBSD Handbook, Volume 2 (Both Volumes)	\$59.95
★ Special: The FreeBSD Handbook, Both Volumes, & FreeBSD 8.1	\$79.95

The FreeBSD Bundle

Inside the Bundle, you'll find:

- FreeBSD Handbook, 3rd Edition, Users Guide
- FreeBSD Handbook, 3rd Edition, Admin Guide
- FreeBSD 8.1 4-disc set
- FreeBSD Toolkit DVD

★ Special: The FreeBSD CD Bundle	\$89.95
★ Special: The FreeBSD DVD Bundle	\$89.95

The FreeBSD Toolkit DVD

.....	\$39.95
-------	---------

FreeBSD Mousepad

.....	\$10.00
-------	---------

FreeBSD Caps

.....	\$20.00
-------	---------

PC-BSD Caps

.....	\$20.00
-------	---------

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

CALL 925.240.6652 Ask about our software bundles!

t-shirts
\$18-\$21.99



The FreeBSD

Ubuntu challenge

FreeBSD makes a great server, but can it rise to the challenge of running Compiz as a workstation?

What you will learn...

- installation of Compiz
- installation of OpenOffice
- Installation the 3D accelerated video driver

What you should know...

- have an idea of Ubuntu and FreeBSD OS

One of the many criticisms of Open Source software (indeed even FreeBSD) is that it is not *ready for the desktop*. While this is a reasonable argument when it comes down to cross-platform compatibility with Microsoft applications using Wine, one of the misconceptions with the BSD platform (with the exception of PC-BSD) is that it is more geared towards the server farm out of the box than a fully functional desktop with GUI. This how-to aims to dispel this myth, and it will demonstrate how to get a fully functioning basic desktop with OpenOffice (*Office Suite*), Firefox (*Browser*),

Gimp (*Graphics editor tool*) as well as the eye-catching 3D windowing effects of Compiz.

System requirements

An essential requirement for Compiz is a graphics card that will support 3D-accelerated graphics. Unfortunately, this limits the choice of graphics card used as some vendors will not release the source code or provide a driver for the *BSD or other Open Source platforms. I will be using the Nvidia driver, and as Compiz integrates very well with the Gnome desktop (The default Ubuntu desktop) and I will use FreeBSD 8.0 as the basis for the install. Depending on the time available to you, you may prefer to compile

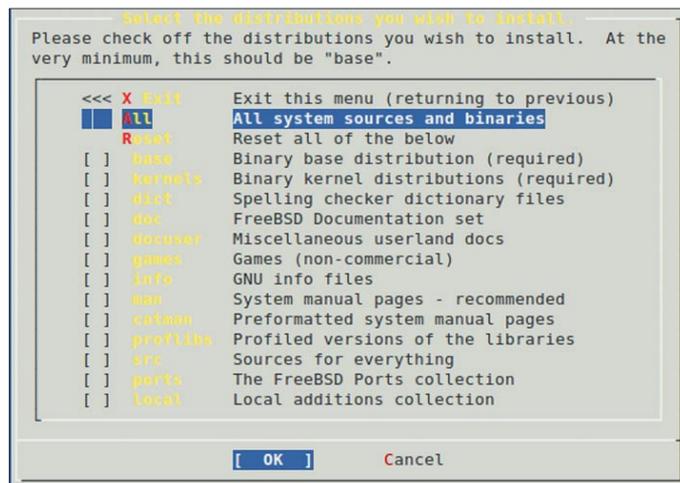


Figure 1. Standard FreeBSD installation

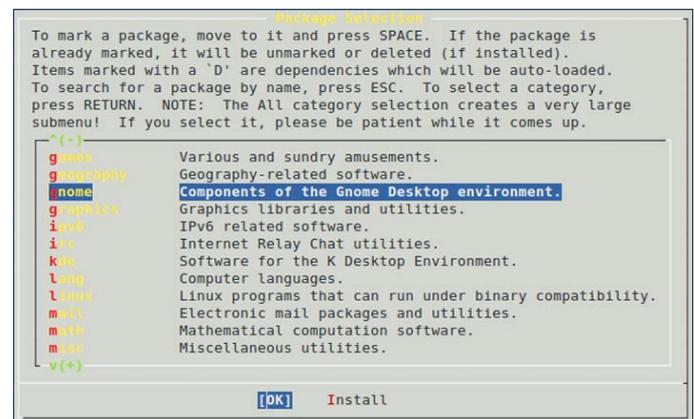


Figure 2. Installation packages: Gnome and all its derivatives

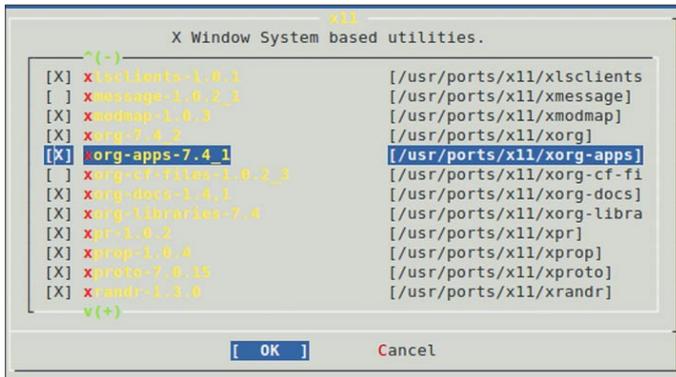


Figure 3. Installation of Xorg 7.4.2

the very latest source code from scratch by using the ports collection. I have tried to avoid this where possible by using packages as the installation method to speed up the install.

Installation

Part 1 – O/S install

Proceed as normal with a standard FreeBSD installation, and install all system sources and binaries, as well as ports (see Figure 1). Additionally, you will need to install the following packages: Gnome and all its derivatives (e.g. GDM, see Figure 2), Xorg 7.4.2 (Figure 3), and xorg-server (Figure 4). Network configuration can proceed to suit your environment, I have used DHCP but access to the internet from the target machine will be required later. When prompted to test and configure the mouse daemon, say yet to this.

Part 2 – Preparing for Xorg and the Nvidia driver

Add the following lines to `/etc/rc.conf` to provide Linux support and allow Xorg to pick up the mouse:

```
linux_enable="YES"
dbus_enable="YES"
hald_enable="YES"
```

Generate a generic `xorg.conf` file so that we can add the driver and various customisations later:

```
Xorg -configure
mv /root/xorg.conf.new /etc/X11/xorg.conf
```

Reboot to ensure Linux compatibility etc. is running prior to compiling the Nvidia driver.

Part 3 – Installing the 3D accelerated video driver

Login to a terminal as root, and compile and install the Nvidia driver – ensure FreeBSD AGP support is disabled (see Figure 5):

If you wish to contribute to BSD magazine, share your knowledge and skills with other BSD users - do not hesitate - read the guidelines on our website and email us your idea for an article.

Join our team!

Become BSD magazine

Author or Betatester

As a betatester you can decide on the contents and the form of our quarterly. It can be you who read the articles before everybody else and suggest the changes to the author.

Contact us:

editors@bsdmag.org

www.bsdmag.org

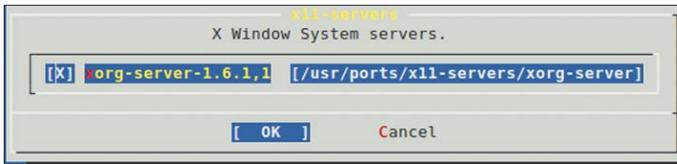


Figure 4. Installation of Xorg-server

```
cd /usr/ports/x11/nvidia-driver
make install clean
```

We can now add the following line to `/boot/loader.conf` to load the Nvidia driver at start-up:

```
nvidia_load="YES"
```

As the Nvidia driver already has AGP support built in, we will need to remind the kernel not to load AGP support. Add this line to `/boot/device.hints`:

```
hint.agp.0.disabled="1"
```

Edit the `/etc/X11/xorg.conf` you generated earlier, and add the following entry under Modules:

```
Load      "extmod"
Load      "glx"
```

Add the following to the Screen section under Monitor:

```
DefaultDepth 24
Option      "AddARGBGLXVisuals" "True"
```

In the Device section, amend the Driver line to read:

```
Driver      "nvidia"
```

At the end of the file add:

```
Section "Extensions"
  Option "Composite" "Enable"
EndSection
```

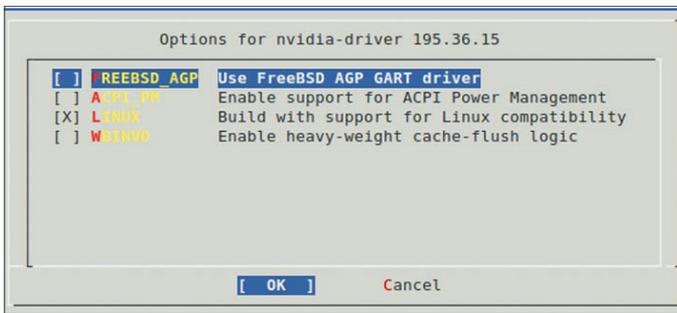


Figure 5. Installing the 3D accelerated video driver

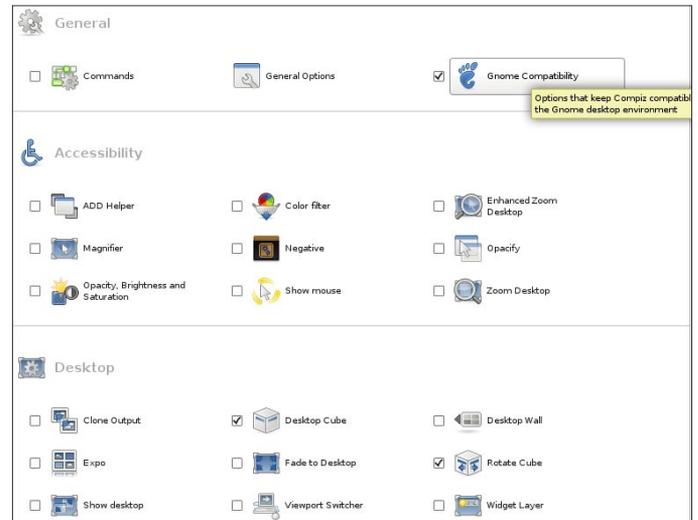


Figure 6. Installing Compiz 1

Part 4 – Installing Compiz

Install Compiz from the packages:

```
pkg_add -r compiz-fusion
```

As root run the following:

```
gdm
```

You should be greeted with the GDM login screen. Login as a standard user, and configure Compiz by running CCSM in a terminal window:

```
ccsm
```

Ensure that the effects are enabled as shown in Figure 6 and Figure 7. Create a shell script in your home directory called `compiz-startup.sh` with the following content:

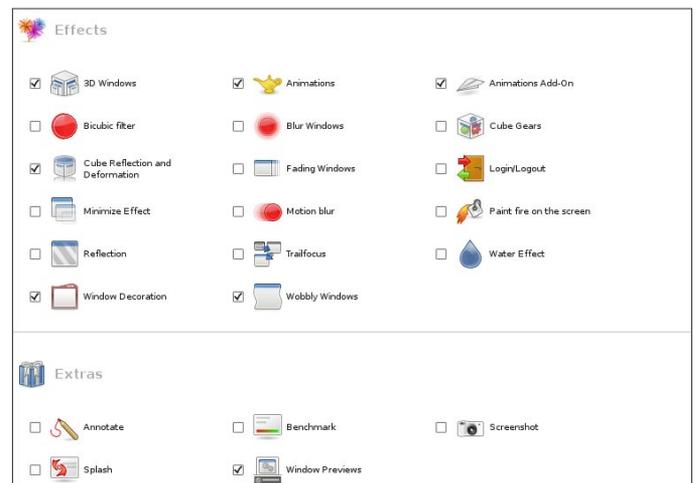


Figure 7. Installing Compiz 2

Listing 1. Sample xorg.conf

```

Section "ServerLayout"                                #Option      "UseFBDev"          # [<bool>]
    Identifier    "X.org Configured"                  #Option      "Rotate"            # [<str>]
    Screen      0  "Screen0" 0 0                      #Option      "VideoKey"          # <i>
    InputDevice "Mouse0" "CorePointer"               #Option      "FlatPanel"         # [<bool>]
    InputDevice "Keyboard0" "CoreKeyboard"           #Option      "FPDither"         # [<bool>]
EndSection                                           #Option      "CrtcNumber"       # <i>
Section "Files"                                       #Option      "FPScale"           # [<bool>]
    ModulePath   "/usr/local/lib/xorg/modules"        #Option      "FPTweak"          # <i>
    FontPath     "/usr/local/lib/X11/fonts/misc/"     #Option      "DualHead"         # [<bool>]
    FontPath     "/usr/local/lib/X11/fonts/TTF/"      Identifier    "Card0"
    FontPath     "/usr/local/lib/X11/fonts/OTF/"      Driver        "nvidia"
    FontPath     "/usr/local/lib/X11/fonts/Type1/"    VendorName    "nVidia Corporation"
    FontPath     "/usr/local/lib/X11/fonts/100dpi/"   BoardName     "G73 [GeForce 7600 GS]"
    FontPath     "/usr/local/lib/X11/fonts/75dpi/"    EndSection
EndSection                                           Section "Screen"
Section "Module"                                       Identifier    "Screen0"
    Load         "dbe"                                Device        "Card0"
    Load         "dri2"                                Monitor       "Monitor0"
    Load         "extmod"                              DefaultDepth  24
    Load         "glx"                                  Option        "AddARGBGLXVisuals" "True"
    Load         "record"                              SubSection    "Display"
EndSection                                           Viewport      0 0
Section "InputDevice"                                 EndSubSection
    Identifier    "Keyboard0"                          SubSection    "Display"
    Driver        "kbd"                                  Viewport      0 0
EndSection                                           Depth         4
Section "InputDevice"                                 EndSubSection
    Identifier    "Mouse0"                              SubSection    "Display"
    Driver        "mouse"                               Viewport      0 0
    Option        "Protocol" "auto"                     Depth         8
    Option        "Device"  "/dev/sysmouse"             EndSubSection
    Option        "ZAxisMapping" "4 5 6 7"             SubSection    "Display"
EndSection                                           Viewport      0 0
Section "Monitor"                                     Depth         15
    Identifier    "Monitor0"                            EndSubSection
    VendorName    "Monitor Vendor"                     SubSection    "Display"
    ModelName     "Monitor Model"                       Viewport      0 0
EndSection                                           Depth         16
Section "Device"                                       EndSubSection
    ### Available Driver options are:-                 SubSection    "Display"
    ### Values: <i>: integer, <f>: float, <bool>:      Viewport      0 0
    ###           "True"/"False",                      Depth         24
    ### <string>: "String", <freq>: "<f> Hz/kHz/MHz"   EndSubSection
    ### [arg]: arg optional                             EndSection
    #Option      "SWcursor"          # [<bool>]
    #Option      "HWcursor"          # [<bool>]
    #Option      "NoAccel"           # [<bool>]
    #Option      "ShadowFB"          # [<bool>]
    Section "Extensions"
        Option        "Composite" "Enable"
    EndSection

```

References

- Nvidia page at *FreeBSD.org*: <http://www.freebsd.org/doc/en/articles/compiz-fusion/nvidia-setup.html>
- FreeBSD website – Configuring sound: http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/sound-setup.html
- OpenOffice pre-load files: Please open <http://java.sun.com/javase/downloads/index.jsp> in a web browser and follow the Download link for *JDK US DST Timezone Update Tool – 1_3_18* to obtain the time zone update file, *tzupdater-1_3_18-2009k.zip*. Please download the patchset, *bsd-jdk16-patches-4.tar.bz2*, from <http://www.eyesbeyond.com/freebsd/java/jdk16.html>.
- Emerald Themes: <http://compiz-themes.org/index.php?xcontentmode=103>
- FreeBSD forums – Install Flashplayer: <http://forums.freebsd.org/showthread.php?t=5786>

```
#!/bin/sh
compiz --replace --sm-disable --ignore-desktop-hints ccp &
emerald --replace &
```

When you created it, flag it as executable:

```
chmod +x compiz-startup.sh
```

Now try running Compiz from within a terminal:

```
./compiz-startup.sh
```

Compiz should load the Emerald window decorator and the desktop effects should be enabled. If you do not have a title bar, check that your *xorg.conf* is setup similar to the sample *xorg.conf* (Figure 8): see Listing 1.

Further Compiz configuration

- Compiz setting are changed via *ccsm*
- Emerald themes are changed via *emerald-theme-manager*.

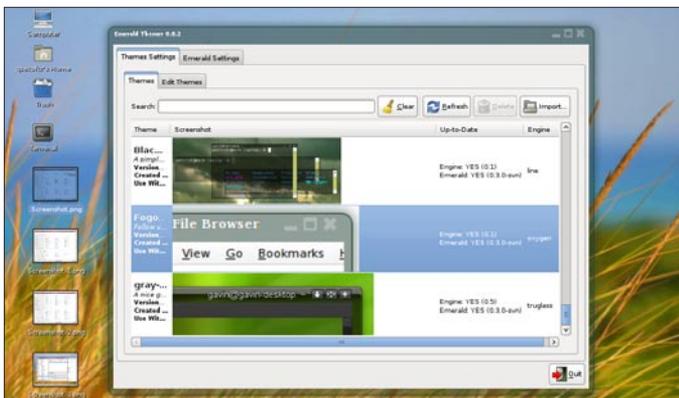


Figure 8. Check that your *xorg.conf* is setup similar to the sample *xorg.conf*

- You can auto-load Compiz by adding it to Startup Applications in the System menu.
- To force GDM to start on boot, add `gdm_enable="YES"` to *rc.conf*.
- Gnome-terminal refused to work on my test box – I got round this by copying the shortcut to the desktop and using *xterm* instead.

Installing the Browser, OpenOffice and Flash etc.

Installing the Gimp, Firefox

```
pkg_add -r gimp
pkg_add -r firefox35
```

Installing Open Office

At the time of writing, OpenOffice was not available as a binary – you may choose to skip installing this as the compile time is extensive. If you want to install it, you will need to download the files listed in the appendix and copy these to `/usr/ports/distfiles` before you commence. To compile OOO from source:

```
cd /usr/ports/editors/openoffice.org-2
make install clean BATCH=YES
```

Further improvements

Unfortunately, the kit I was working with didn't have a suitable sound card (3D support with a virtual machine is very experimental). I was also experiencing a fatal error installing flashplayer: Attempting to fetch from <http://fpdownload.macromedia.com/get/flashplayer/current/>. fetch:http://fpdownload.macromedia.com/get/flashplayer/current/install_flash_player_10_linux.tar.gz: size mismatch: expected 4044751, actual 4760657

Due to publication deadlines there was not time to investigate this, but hopefully by the time this how-to is released the problem will have been solved. I have run Flashplayer and Firefox together on other desktops, and it works well.

Please see the appendix for details of how to configure these with FreeBSD.

ROB SOMERVILLE

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

genioDATA

Data Security

Replicate your databases in high class data centers.

Have an email archive run by **genioDATA** that leaves nothing more to wish for.

Copy your files to several sites to plan for disaster recovery.

Get Your Own World

Operating Systems are the worlds in the IT universe.

Get a world of your own. Thoroughly engineered. Run on first class hardware by highly experienced admins using perfectly developed processes.

Book your own BSD, MacOS X, Linux, Solaris, HPUX, Windows servers. Virtualized or on real hardware.

Got an idea? Make it live. In a **genioDATA** Server.

Individual Appliances

Need an ERP environment

(enterprise resource planning)?

Have to operate a web(services)

cluster with 99,999 % availability?

Need an email environment where

not one email gets lost?

genioDATA engineers it. **genioDATA** runs it. You use it.

To you it is an appliance. Just get and switch it on.



Network monitoring

with Nagios and OpenBSD, Part 1

So our OpenBSD-based network now includes redundant firewalls (<http://www.kernel-panic.it/openbsd/carp/index.html>), domain name servers (<http://www.kernel-panic.it/openbsd/dns/index.html>), a mail gateway (<http://www.kernel-panic.it/openbsd/mail/index.html>) and a web proxy cache (<http://www.kernel-panic.it/openbsd/proxy/index.html>).

What you will learn...

- Installing Nagios
- How to monitor network with Nagios and Open BSD

What you should know...

- A good knowledge of OpenBSD administration
- Basic MySQL database administration

All the services provided by these machines are particularly critical and can't afford even minimal downtime. Redundancy may give us the time to recover a failure before having angry users trying to knock down our door, but it doesn't free us from the responsibility to detect and solve ongoing problems.

To put it short, it's time to think about monitoring our network! And the following are the perfect ingredients for implementing a full-featured, secure and reliable network monitoring system:

OpenBSD (<http://www.openbsd.org/>)

the operating system for the security paranoid, with *only two remote holes in the default install, in a heck of a long time!*;

Nagios (<http://www.nagios.org/>)

the leader and industry standard in enterprise system, network, and application monitoring;

Apache (<http://httpd.apache.org/>)

the secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards.

My pick goes to Nagios for its ease of use, flexibility and extensibility. It also features a very clean and straightforward design, as it is structured into three basic building blocks:

- a daemon process, running periodic checks on specific hosts and services and managing notifications when problems arise;
- an optional web interface, to access current status information, historical logs and reports via a simple web browser;
- a set of external plugins, i.e. the (possibly custom) scripts executed by the daemon process to actually perform the checks and send out notifications.

Furthermore, these basic components can be easily extended with external modules, making it easy for Nagios to meet even your most demanding needs! Therefore, after the installation and configuration of the Nagios' core components, we will take a brief look at some of its most popular and useful addons (http://www.nagiosexchange.org/AddOn_Projects.22.0.html):

- NRPE (<http://www.kernel-panic.it/openbsd/nagios/nagios5.html#nagios-5.1>), the Nagios Remote Plugin Executor, which allows you to execute local plugins on remote hosts;
- NSCA (<http://www.kernel-panic.it/openbsd/nagios/nagios5.html#nagios-5.2>), the Nagios Service Check Acceptor, which processes passive service check results submitted by clients to the Nagios server;

- NagVis (<http://www.kernel-panic.it/openbsd/nagios/nagios5.html#nagios-5.3>), the Nagios Visualization Addon, which allows you to deeply customize how Nagios data is displayed;

A good knowledge of OpenBSD is assumed, since we won't delve into system management topics such as base configuration or packages/ports installation.

Installation and base configuration

Before delving straight into the details of Nagios installation and configuration, let's take a brief look at the layout of the network that we're going to monitor (Figure 1).

It's a very simple and small network, made up of:

- a LAN (172.16.0.0/24), containing clients and servers not accessible from the public Internet (e.g. file server, DHCP server);
- a DMZ (172.16.240.0/24), containing the servers that must access the Internet (e.g. mail, web and proxy servers);
- a router, in a small subnet (172.16.250.0/24), connecting the DMZ to the Internet.

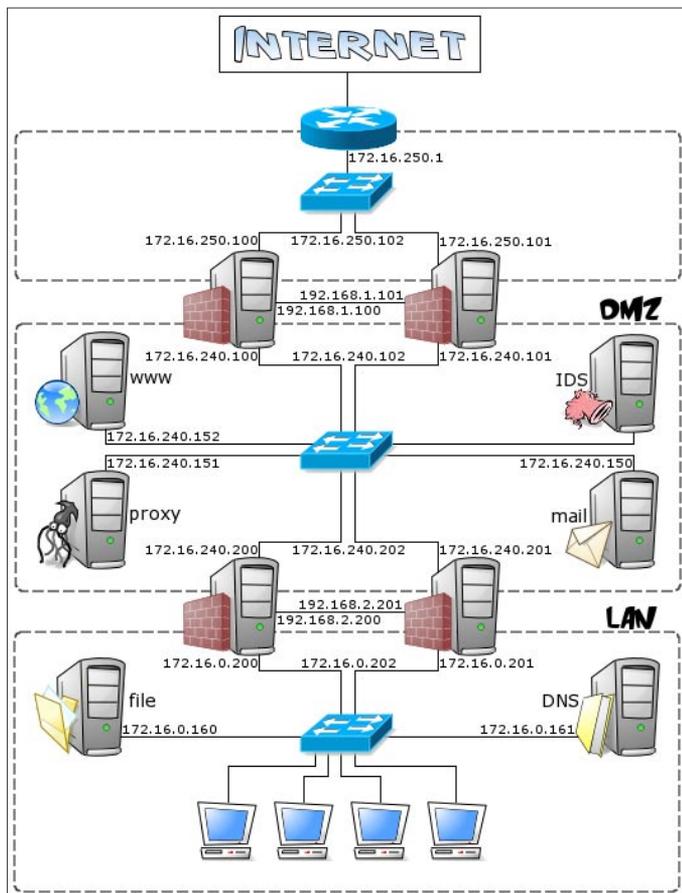


Figure 1. Monitored Network

Our network monitoring system is a security-critical host and won't need to directly access the Internet, so it will perfectly fit in the internal LAN.

The OpenBSD installation procedure is documented in full detail in the official FAQ (<http://www.openbsd.org/faq/faq4.html>), so we won't linger on it here. Nagios doesn't have particular requirements and a standard OpenBSD installation will do just fine: according to the documentation (http://nagios.sourceforge.net/docs/3_0/about.html#requirements), Nagios makes do with just a machine running Linux (or UNIX variant). That doesn't sound so fussy, does it?

Packages installation

Nagios installation only requires adding a few packages (<http://www.openbsd.org/faq/faq15.html#PkgInstall>):

- libiconv-x.x.x.tgz
- gettext-x.x.x.tgz
- pcre-x.x.tgz
- glib2-x.x.x.tgz
- libltdl-x.x.tgz
- nagios-plugins-x.x.tgz
- nagios-x.x-chroot.tgz
- nagios-web-x.x-chroot.tgz

The installation procedure will automatically create the user and group that the monitoring daemon will drop its privileges to (`_nagios`). The `chroot` flavor will install Nagios in a way suited for `chrooted httpd(8)` (<http://www.openbsd.org/cgi-bin/man.cgi?query=httpd&sektion=8>), i.e. with the CGIs (http://nagios.sourceforge.net/docs/3_0/cgis.html) statically linked and all the configuration and log files stored inside the `/var/www` directory. By the way, Nagios has a particular directory structure that you will have to become familiar with:

`/var/www/nagios/`

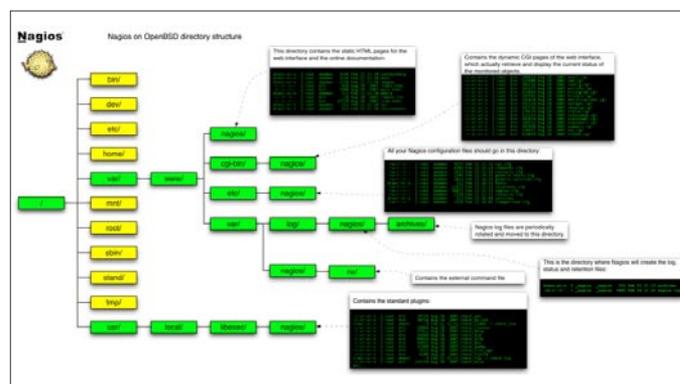


Figure 2. Directory structure of Nagios

this directory contains the static HTML pages for the web interface and the online documentation;

```
/var/www/cgi-bin/nagios/
```

contains the dynamic CGI pages of the web interface, which actually retrieve and display the current status of the monitored objects;

```
/var/www/etc/nagios/
```

you should put all your Nagios configuration files in this directory: we will examine them one by one in a moment;

```
/var/www/var/log/nagios/
```

this is the directory where Nagios will create the log (http://nagios.sourceforge.net/docs/3_0/configmain.html#log_file), status (http://nagios.sourceforge.net/docs/3_0/configmain.html#status_file) and retention (http://nagios.sourceforge.net/docs/3_0/configmain.html#state_retention_file) files;

```
/var/www/var/log/nagios/archives/
```

Nagios log files are periodically rotated and moved to this directory;

```
/var/www/var/nagios/rw/
```

contains the external command file (http://nagios.sourceforge.net/docs/3_0/configmain.html#command_file);

```
/usr/local/libexec/nagios/
```

contains the standard plugins (http://nagios.sourceforge.net/docs/3_0/plugins.html).

As a reference, below is a visual representation of the directory structure of Nagios, kindly submitted by Bren Smith (click [here http://www.kernel-panic.it/openbsd/nagios/nagiosdirstruct.png](http://www.kernel-panic.it/openbsd/nagios/nagiosdirstruct.png) for a larger view; see Figure 2).

Configuration overview

Nagios configuration may look overly complicated at first glance; even the documentation (http://nagios.sourceforge.net/docs/3_0/beginners.html) warns that *Nagios is quite powerful and flexible, but it can take a lot of work to get it configured just the way you'd like*. Anyway, don't despair! Once you've figured out the underlying logic of its *object-oriented* configuration, you

will appreciate Nagios' flexibility and clean design. For the first tests, you can start by tweaking the sample configuration files contained in the `/usr/local/share/examples/nagios/` directory, customizing them to your needs.

The syntax of Nagios configuration files follows a few basic rules:

- comments start with a # character and span to the end of the line;
- variable names must begin at the start of the line (i.e. no indentation allowed);
- variable names are case sensitive;
- no spaces are allowed around the = sign.

Configuration involves setting several parameters concerning the monitoring daemon, the CGIs and, of course, the hosts and services you want to monitor. All this information is spread across multiple files: we will now examine them one by one.

The main configuration file

The overall behaviour of the Nagios daemon is determined by the directives included in the main configuration file, `/var/www/etc/nagios/nagios.cfg`. Though this file contains several dozens of parameters, for most of them the default value is the most reasonable option and you will probably want to care about only very few of them (usually `cfg_file` http://nagios.sourceforge.net/docs/3_0/configmain.html#cfg_file, `cfg_dir` (http://nagios.sourceforge.net/docs/3_0/configmain.html#cfg_dir) and `admin_email` http://nagios.sourceforge.net/docs/3_0/configmain.html#admin_email). In any case, you can find a detailed description of each and every parameter in the official documentation (http://nagios.sourceforge.net/docs/3_0/configmain.html see Listing 1).

The resource file

The resource file allows you to assign values to the user-definable macros `$USERn$` (where `n` is a number between 1 and 32 inclusive). Basically, in Nagios, macros are variables (starting and ending with a dollar sign, `$`) that you can insert into command definitions and that will get expanded to the appropriate value immediately prior to the execution of the command. User-defined macros (and the several other macros http://nagios.sourceforge.net/docs/3_0/macros.html Nagios makes available) allow you to keep command definitions generic and simple (see the next chapter for some examples).

Listing 1a. Main configuration file

```

/var/www/etc/nagios/nagios.cfg
# Path to main log file and log archive directory. All
    pathnames are relative
# to the chroot directory '/var/www/'
log_file=/var/log/nagios/nagios.log
log_archive_path=/var/log/nagios/archives

# Paths to files managed internally by the application
object_cache_file=/var/nagios/objects.cache
precached_object_file=/var/nagios/objects.precache
status_file=/var/nagios/status.dat
state_retention_file=/var/nagios/retention.dat
command_file=/var/nagios/rw/nagios.cmd
lock_file=/var/run/nagios/nagios.pid
temp_file=/var/nagios/nagios.tmp
temp_path=/tmp
check_result_path=/var/spool/nagios

# Object definitions (see next chapter) can be split
    across multiple files.
# You may either list files individually (using the
    'cfg_file' parameter) or
# group them into directories (using the 'cfg_dir'
    parameter). In the latter
# case, Nagios will process all files with a '.cfg'
    extension found in the
# specified directories and their subdirectories
cfg_file=/etc/nagios/timeperiods.cfg
cfg_file=/etc/nagios/contacts.cfg
cfg_file=/etc/nagios/commands.cfg
cfg_file=/etc/nagios/generic-hosts.cfg
cfg_file=/etc/nagios/generic-services.cfg
cfg_dir=/etc/nagios/hosts
cfg_dir=/etc/nagios/services

# Path to the resource file, containing user-defined
    macros (see below). You can
# specify more than one resource file using multiple
    'resource_file' statements
resource_file=/etc/nagios/resource.cfg

# User and group the Nagios process will run as
nagios_user=_nagios
nagios_group=_nagios

# Email address and pager number for the administrator
    of the local machine
admin_email=nagios@kernel-panic.it
admin_pager=xxx-xxx-xxxx

# Date format (available options: us, euro, iso8601 or
    strict-iso8601)
date_format=euro

# Enable checks, notifications and event handlers.
    Passive checks allow external
# applications to submit check results to Nagios. Event
    handlers are optional
# commands that are executed whenever a host or service
    state change occurs
execute_service_checks=1
accept_passive_service_checks=1
execute_host_checks=1
accept_passive_host_checks=1
enable_notifications=1
enable_event_handlers=1

# Checks freshness options. Enabling these options will
    ensure that passive
# checks are always up-to-date
check_service_freshness=1
service_freshness_check_interval=60
check_host_freshness=0
host_freshness_check_interval=60
additional_freshness_latency=15

# External commands allow the web interface and
    external applications (such as
# NSCA) to issue commands to Nagios. With a check
    interval of '-1', Nagios will
# check for external commands as often as possible
check_external_commands=1
command_check_interval=-1
external_command_buffer_slots=4096

# Various logging options
log_rotation_method=d
use_syslog=1
log_notifications=1
log_service_retries=1
log_host_retries=1
log_event_handlers=1
log_initial_states=0
log_external_commands=1
log_passive_checks=1

```

Listing 1b. Main configuration file

```

# Enable retention of state information between program
    restarts (refer to
# documentation for details)
retain_state_information=1
retention_update_interval=60
use_retained_program_state=1
use_retained_scheduling_info=0
retained_host_attribute_mask=0
retained_service_attribute_mask=0
retained_process_host_attribute_mask=0
retained_process_service_attribute_mask=0
retained_contact_host_attribute_mask=0
retained_contact_service_attribute_mask=0

# State flapping detection options (refer to
    documentation for details)
enable_flap_detection=0
low_service_flap_threshold=5.0
high_service_flap_threshold=20.0
low_host_flap_threshold=5.0
high_host_flap_threshold=20.0

# Miscellaneous tuning, performance and security
    options (refer to
# documentation for details)
interval_length=60
service_inter_check_delay_method=s
max_service_check_spread=30
service_interleave_factor=s
host_inter_check_delay_method=s
max_host_check_spread=30
max_concurrent_checks=0
check_result_reaper_frequency=10
max_check_result_reaper_time=30
max_check_result_file_age=3600
cached_host_check_horizon=15
cached_service_check_horizon=15
enable_predictive_host_dependency_checks=1
enable_predictive_service_dependency_checks=1
soft_state_dependencies=0
auto_reschedule_checks=0
auto_rescheduling_interval=30
auto_rescheduling_window=180
status_update_interval=15
event_broker_options=-1

sleep_time=0.25
service_check_timeout=60
host_check_timeout=30
event_handler_timeout=30
notification_timeout=30
ocsp_timeout=5
perfdata_timeout=5

use_aggressive_host_checking=0
process_performance_data=0
obsess_over_services=0
obsess_over_hosts=0
translate_passive_host_checks=0
passive_host_checks_are_soft=0

check_for_orphaned_services=0
check_for_orphaned_hosts=1

p1_file=/usr/local/bin/p1.pl
enable_embedded_perl=1
use_embedded_perl_implicitly=1

illegal_object_name_chars='~!$%^&*|'"<>?,()=
illegal_macro_output_chars='~$&|'"<>
use_regexp_matching=0
use_true_regexp_matching=0
daemon_dumps_core=0
use_large_installation_tweaks=0
enable_environment_macros=1

# Debug options
debug_level=0
debug_verbosity=1
debug_file=/var/nagios/nagios.debug
max_debug_file_size=1000000

```

User-defined macros are normally used to store recurring items in command definitions (like directory paths) and sensitive information (like usernames and passwords). It is recommended that you set restrictive permissions (600) on the resource file(s) in order to keep sensitive information protected.

```
/var/www/etc/nagios/resource.cfg
# Set $USER1$ to be the path to the plugins
$USER1$=/usr/local/libexec/nagios
# MySQL username and password
$USER2$=root
$USER3$=password
```

The next step is configuring object data, which is probably the trickiest part of the configuration. We will therefore devote the next chapter entirely to this topic.

Object data configuration

So now it's time to tell Nagios what to keep tabs on. Therefore, we must supply it with information about:

- *when* and *how* to perform checks and send out notifications;
- *whom* to notify;
- *which* hosts and services to monitor.

All this information is represented by means of objects, which are defined by a set of `define` statements, enclosed in curly braces and containing a variable number of newline-separated directives, in keyword/value form. Keywords are separated from values by whitespace and multiple values can be separated by commas; indentation within statements is allowed.

To recap, the basic syntax of an object declaration can be represented as follows:

```
define object {
    keyword-1    value-1
    keyword-2    value-2,value-3,...
    [...]
    keyword-n    value-n
}
```

Object definitions can be split into any number of files: just remember to list them all in the main configuration file by using the `cfg_file` and/or `cfg_dir` directives.

Timeperiod definition

The `timeperiod` statement allows you to specify, for each day of the week, one or more time slots in which to run

certain checks and/or notify certain people. Time intervals can't span across midnight and excluded days are simply omitted.

In the following example, all the `timeperiod` definitions are grouped together in a file named `timeperiods.cfg` stored in the `/var/www/etc/nagios/` directory (see Listing 2).

Command definition

The next step is to tell Nagios *how* to perform the various checks and send out notifications; this is accomplished by defining multiple `command` objects specifying the actual commands for Nagios to run.

Command definitions are pairs of short names and command lines (both mandatory) and can contain macros. As we mentioned before, macros are variables, enclosed in `§` signs, that will get expanded to the appropriate value immediately prior to the execution of a command; macros allow you to keep command definitions generic and straightforward. A simple example will make this clear.

Suppose you want to monitor a web server with IP address `1.2.3.4`; you could then define a command such as the following:

```
define command {
    command_name    check-http
    command_line    /usr/local/libexec/nagios/check_http
                    -I 1.2.3.4
}
```

This definition is correct and will certainly do the job. But what if you later decide to add a new web server? Would you find it convenient to define a new (almost identical) command, with only the IP address changed? It is way more efficient to take advantage of macros by writing a single generic command such as:

```
define command {
    command_name    check-http
    command_line    $USER1$/check_http -I $HOSTADDRESS$
}
```

and leave Nagios the responsibility to expand the built-in `$HOSTADDRESS$` macro to the appropriate IP address, obtained from the host definition (see below). As you'll remember from the previous chapter, the `$USER1$` macro holds the path to the plugins directory.

Now let's complicate things a bit! What if you want Nagios to check the availability of a particular URL on each web server? This URL may differ from server to server, so what we need now is a command definition

HOW TO'S

that is still generic and yet server-specific! Though this may sound contradictory, once again Nagios solves this problem with macros: in fact, the `$ARGn$` macros (where `n` is a number between 1 and 32 inclusive) act as placeholders for service-specific arguments that will be specified later within service definitions (see below for further details). Therefore, the above command definition would turn into:

```
define command {
    command_name    check-http
```

```
    command_line    $USER1$/check_http -I $HOSTADDRESS$ -u
                    $ARG1$
}
```

In addition to the ones we have just seen, Nagios provides several other useful macros. Please refer to the documentation (http://nagios.sourceforge.net/docs/3_0/macros.html) for a detailed list of all available macros and their validity context. Below is a sample set of command definitions (see Listing 3).

Listing 2. The time period definitions are grouped together and stored

```
/var/www/etc/nagios/timeperiods.cfg
# The following timeperiod definition includes normal
    work hours. The
# 'timeperiod_name' and 'alias' directives are
    mandatory. Note that weekend days
# are simply omitted
define timeperiod {
    timeperiod_name    workhours
    alias              Work Hours
    monday             09:00-18:00
    tuesday            09:00-18:00
    wednesday          09:00-18:00
    thursday           09:00-18:00
    friday             09:00-18:00
}

# The following timeperiod includes all time outside
    normal work hours. The
# time slot between 6 p.m. and 9 a.m. must be split
    into two intervals, to avoid
# crossing midnight
define timeperiod {
    timeperiod_name    nonworkhours
    alias              Non-Work Hours
    sunday             00:00-24:00
    monday             00:00-09:00,18:00-24:00
    tuesday            00:00-09:00,18:00-24:00
    wednesday          00:00-09:00,18:00-24:00
    thursday           00:00-09:00,18:00-24:00
    friday             00:00-09:00,18:00-24:00
    saturday           00:00-24:00
}

# Most checks will probably run on a continuous basis
define timeperiod {
    timeperiod_name    always
    alias              Every Hour Every Day
    sunday             00:00-24:00
    monday             00:00-24:00
    tuesday            00:00-24:00
    wednesday          00:00-24:00
    thursday           00:00-24:00
    friday             00:00-24:00
    saturday           00:00-24:00
}

# The right timeperiod when you don't want to bother
    with notifications (e.g.
# during testing)
define timeperiod {
    timeperiod_name    never
    alias              No Time is a Good Time
}

# Some exceptions to the normal weekly time (see
    documentation for more examples)
define timeperiod {
    timeperiod_name    exceptions
    alias              Some random dates
    2008-12-15         00:00-24:00    ; December 15th, 2008
    friday 3           00:00-24:00    ; 3rd Friday of every
                                month
    february -1       00:00-24:00    ; Last day in
                                February of every year
    march 20 - june 21 00:00-24:00    ; Spring
    day 1 - 15        00:00-24:00    ; First half of every
                                month
    2008-01-01 / 7    00:00-24:00    ; Every 7 days from
                                Jan 1st, 2008
}
```

Contact definition

contact objects allow you to specify people who should be notified automatically when the alert conditions are met. Contacts are first defined individually and then

grouped together in contactgroup objects, for easier management.

For the first time, in the following definitions, we will refer to previously defined objects. In fact, the

Listing 3. A sample set of command definitions

```

/var/www/etc/nagios/commands.cfg
#####
# Notification commands                                #
# There are no standard notification plugins; hence    #
# notification commands are #                          #
# usually custom scripts or mere command lines.      #
#####
define command {
    command_name    host-notify-by-email
    command_line    $USER1$/host_notify_by_email.sh
                  $CONTACTEMAIL$
}

define command {
    command_name    notify-by-email
    command_line    $USER1$/notify_by_email.sh
                  $CONTACTEMAIL$
}

define command {
    command_name    host-notify-by-SMS
    command_line    /usr/local/bin/sendsms $ADDRESS1$
                  "Nagios: Host $HOSTNAME$
                  ($HOSTADDRESS$) is in state:
                  $HOSTSTATE$"
}

define command {
    command_name    notify-by-SMS
    command_line    /usr/local/bin/sendsms $ADDRESS1$
                  "Nagios: Service $SERVICEDESC$
                  on $HOSTALIAS$ is in state:
                  $SERVICESTATE$"
}

#####
# Check commands                                      #
# The official Nagios plugins should handle most of your
# needs for host and #
# service checks. Anyway, should they not, we will
# discuss in a moment how to #
#####
# write custom plugins.                                #
#####
define command {
    command_name    check-host-alive
    command_line    $USER1$/check_ping -H $HOSTADDRESS$
                  -w 3000.0,80% -c 5000.0,100% -p 1
}

define command {
    command_name    check-ssh
    command_line    $USER1$/check_ssh $HOSTADDRESS$
}

define command {
    command_name    check-http
    command_line    $USER1$/check_http -I $HOSTADDRESS$
                  -u $ARG1$
}

define command {
    command_name    check-smtp
    command_line    $USER1$/check_smtp -H $HOSTADDRESS$
}

define command {
    command_name    check-imap
    command_line    $USER1$/check_imap -H $HOSTADDRESS$
}

define command {
    command_name    check-dns
    command_line    $USER1$/check_dns -s $HOSTADDRESS$
                  -H $ARG1$ -a $ARG2$
}

define command {
    command_name    check-mysql
    command_line    $USER1$/check_mysql -H $HOSTADDRESS$
                  -u $USER2$ -p $USER3$
}
[...]
```

Listing 4. Command objects

```

/var/www/etc/nagios/contacts.cfg
define contact {
# Short name to identify the contact
    contact_name          john
# Longer name or description
    alias                  John Doe
# Enable notifications for this contact
    host_notifications_enabled 1
    service_notifications_enabled 1
# Timeperiods during which the contact can be notified
    about host and service
# problems or recoveries
    host_notification_period    always
    service_notification_period always
# Host states for which notifications can be sent out
    to this contact
# (d=down, u=unreachable, r=recovery, f=flapping,
    n=none)
    host_notification_options    d,u,r
# Service states for which notifications can be sent out
    to this contact
# (w=warning, c=critical, u=unknown, r=recovery,
    f=flapping, n=none)
    service_notification_options w,u,c,r

# Command(s) used to notify the contact about host and
    service problems
# or recoveries
    host_notification_commands    host-notify-by-
        email,host-notify-by-SMS
    service_notification_commands notify-by-
        email,notify-by-SMS
# Email address for the contact
    email                    jdoe@kernel-panic.it
# Nagios provides 6 address directives (named address1
    through address6) to
# specify additional "addresses" for the contact (e.g.
    a mobile phone number
# for SMS notifications)
    address1                    xxx-xxx-xxxx

# Allow this contact to submit external commands to
    Nagios from the CGIs
    can_submit_commands        1
}

# The following contact is split in two, to allow for
    different notification

# options depending on the timeperiod
define contact {
    contact_name                danix@work
    alias                        Daniele Mazzocchio
    host_notifications_enabled    1
    service_notifications_enabled 1
    host_notification_period      workhours
    service_notification_period    workhours
    host_notification_options      d,u,r
    service_notification_options    w,u,c,r
    host_notification_commands      host-notify-by-email
    service_notification_commands    notify-by-email
    email                          danix@kernel-
        panic.it
    can_submit_commands            1
}

define contact {
    contact_name                danix@home
    alias                        Daniele Mazzocchio
    host_notifications_enabled    1
    service_notifications_enabled 1
    host_notification_period      nonworkhours
    service_notification_period    nonworkhours
    host_notification_options      d,u
    service_notification_options    c
    host_notification_commands      host-notify-by-
        email,host-notify-by-SMS
    service_notification_commands    notify-by-
        email,notify-by-SMS
    email                          danix@kernel-panic.it
    address1                        xxx-xxx-xxxx
    can_submit_commands            1
}
[...]
# All administrator contacts are grouped together in
    the "Admins"
# contactgroup
define contactgroup {
    contactgroup_name            Admins
    alias                          Nagios
        Administrators
    members                        danix@work,danix@home,john
}
[...]

```

Listing 5. *The creation of a template*

```

define host {
    name                generic-host-                across program restarts
                    template # Template name                retain_nonstatus_information 1

    check_command       check-host-alive                # Enable notifications
                    notifications_enabled                1
    check_period        always                          # Time interval (in minutes) between consecutive
                    # server being _still_ down or unreachable
    max_check_attempts  5                               notification_interval        120
    notification_options d,u,r                          # Time period during which notifications about this host
                    # Don't register it!                notification_period          always
}
                    # Host states for which notifications should be sent
                    # u=unreachable, r=recovery, f=flapping, n=none)
                    out (d=down,
                    notification_options                d,u,r
                    # Don't register this definition: it's only a template,
                    not an actual host
                    register                                0
}

```

Listing 6. *Nagios allows multiple levels of template objects*

```

/var/www/etc/nagios/generic-hosts.cfg
# The following is a template for all hosts in the LAN
define host {
# Template name
    name                generic-lan-host                # DMZ hosts inherit all attributes from the generic-
                    # Command to use to check the state of the host
                    check_command                check-host-alive
                    # Contact groups to notify about problems (or
                    recoveries) with this host
                    contact_groups                Admins
                    # Enable active checks
                    active_checks_enabled        1
                    # Time period during which active checks of this host
                    can be made
                    check_period                always
                    # Number of times that Nagios will repeat a check
                    returning a non-OK state
                    max_check_attempts          3
                    # Enable the event handler
                    event_handler_enabled        1
                    # Enable the processing of performance data
                    process_perf_data            1
                    # Enable retention of host status information across
                    program restarts
                    retain_status_information     1
                    # Enable retention of host non-status information
                    # Don't register this definition: it's only a template,
                    not an actual host
                    register                                0
}
}

```

values of the `host_notification_period` and `service_notification_period` directives must be timeperiod objects; (<http://www.kernel-panic.it/opensource/nagios/nagios3.html#nagios-3.1>) and the values of the `host_notification_command` and `service_notification_command` directives must be command objects (<http://www.kernel-panic.it/opensource/nagios/nagios3.html#nagios-3.2>; see Listing 4).

Host definition

Now we have finally come to one of the most important facets of Nagios configuration: the definition of the hosts (servers, workstations, devices, etc.) that we want to monitor. This will lead us to introduce one of the most powerful features of Nagios configuration: object inheritance (http://nagios.sourceforge.net/docs/3_0/objectinheritance.html). Note that, though we are discussing it now first, object inheritance applies to all Nagios objects; however, it's in the definition of hosts and services that you can get the most out of it.

In fact, configuring a host requires setting up quite a few parameters; and the value of these parameters will normally be the same for most hosts. Without object inheritance, this would mean wasting a lot of time typing the same parameters over and over again and eventually ending up with cluttered, overweight and almost unmanageable configuration files.

But luckily, Nagios is smart enough to save you a lot of typing by allowing you to define special template objects, whose properties can be *inherited* by other objects without having to rewrite them. Below is a brief example of how a template is created: see Listing 5.

As you can see, a template definition looks almost identical to a normal object definition. The only differences are:

- every template must be assigned a name with the name directive;
- since this is not an actual host, you must tell Nagios not to register it by setting the value of the `register` directive to 0; this property doesn't get inherited and defaults to 1, so you won't need to explicitly override it in all *children* objects;
- a template object can be left incomplete, i.e. it may not supply all mandatory parameters.

To create an actual host object from a template, you simply have to specify the template name as the value of the `use` directive and make sure that all mandatory fields are either inherited or explicitly set:

```
define host {
    host_name          hostname
    use                generic-host-template
    alias              alias
    address             x.x.x.x
}
```

Well, now let's move from theory to practice and define two host templates for our servers. Note that the second one inherits from the first; this is possible because Nagios allows multiple levels of template objects (see Listing 6).

Now we can take advantage of our templates to define the actual hosts in a few lines (see Listing 7).

Hosts can optionally be grouped together with the `hostgroup` statement, which has no effect on monitoring, but simply allows you to display the hosts in groups in the CGIs (see Listing 8).

Service definition

Configuring the services to monitor is much like configuring hosts: object inheritance can save you a lot of typing and you can group services together with the optional `servicegroup` statement. The definition of our service template (see Listing 9).

Now, before moving to services definitions, we should complete our discussion on passing service-specific arguments to commands by means of the `$ARGn$` macros. As you'll remember, these macros act as placeholders: they expand to the *n*th argument passed to the command in the service definition; for instance, a command definition such as the following expects to be passed two arguments:

```
define command {
    command_name      some-command
    command_line      $USER1$/check_
                      something $ARG1$ $ARG2$
}
```

Therefore, to configure a service check to use the above command, we will need to assign the `check_command` variable a string containing the command's short name followed by the arguments, separated by `!` characters. E.g.:

```
define service {
    service_description  some-service
    check_command        some-command!arg-
                        1!arg-2
    [...]
}
```

Listing 7. Defining the actual hosts in a few lines

```

/var/www/etc/nagios/hosts/servers.cfg
# Configuration for host dns1.lan.kernel-panic.it
define host {
    use                generic-lan-host
    host_name          dns1
    alias              LAN primary master
                    name server
    address            172.16.0.161
    notes              This is the Squid
                    proxy server
    notes_url          http://www.kernel-
                    panic.it/openbsd/proxy/
    icon_image         proxy.png
    icon_image_alt     [Proxy]
    statusmap_image    proxy.gd2
}

# Extended information (completely optional)
notes                This is the
                    internal primary master name server
# URL with more information about this host
notes_url            http://www.kernel-
                    panic.it/openbsd/dns/
# Image associated with this host in the status CGI;
                    images must be placed in
# /var/www/nagios/images/logos/
icon_image           dns.png
# String used in the 'alt' tag of the icon_image
icon_image_alt       [dns]
# Image associated with this host in the statusmap CGI
statusmap_image      dns.gd2
}

# Configuration for host mail.kernel-panic.it
define host {
    use                generic-dmz-host
    host_name          mail
    alias              Mail server
    address            172.16.240.150
    notes              This is the Postfix
                    mail server (with IMAP(S) and web
                    access)
    notes_url          http://www.kernel-
                    panic.it/openbsd/mail/
    icon_image         mail.png
    icon_image_alt     [Mail]
    statusmap_image    mail.gd2
}

# Configuration for host proxy.kernel-panic.it
define host {
    use                generic-dmz-host
    host_name          proxy
    alias              Proxy server
    address            172.16.240.151
    notes              This is the Squid
                    proxy server
    notes_url          http://www.kernel-
                    panic.it/openbsd/proxy/
    icon_image         proxy.png
    icon_image_alt     [Proxy]
    statusmap_image    proxy.gd2
}

[...]

/var/www/etc/nagios/hosts/firewalls.cfg
# Configuration for host fw-int.kernel-panic.it
define host {
    use                generic-lan-host
    host_name          fw-int
    alias              Internal firewalls'
                    CARP address
    address            172.16.0.202
    notes              Virtual CARP
                    address of the internal firewalls
    notes_url          http://www.kernel-
                    panic.it/openbsd/carp/
    icon_image         fw.png
    icon_image_alt     [FW]
    statusmap_image    fw.gd2
}

# Configuration for host mickey.kernel-panic.it
define host {
    use                generic-lan-host
    host_name          mickey
    alias              Internal Firewall
                    #1
    address            172.16.0.200
    notes              Internal firewall
                    (first node of a two-nodes CARP
                    cluster)
    notes_url          http://www.kernel-
                    panic.it/openbsd/carp/
    icon_image         fw.png
    icon_image_alt     [FW]
    statusmap_image    fw.gd2
}

[...]

```

Listing 8. *Displaying the hosts in groups in the CGIs*

```

/var/www/etc/nagios/hosts/hostgroups.cfg
# Domain Name Servers
define hostgroup {
    hostgroup_name      DNS
    alias                Domain Name Servers
    members             dns1,dns2,dns3,dns4
    notes               Our internal Domain
                       Name Servers, running Bind 9.4.2-P2
}
# Firewalls
define hostgroup {
    hostgroup_name      firewalls
    alias               CARP Firewalls
    members             mickey,minnie,dona
                       ld,daisy,fw-int,fw-ext
    notes               Our CARP-enabled
                       firewalls (both virtual and physical
                       addresses)
}
# Web servers
define hostgroup {
    hostgroup_name      WWW
    alias               Web Servers
    members             www1,www2
    notes               Our corporate web
                       servers, running Apache 1.3
}
# Time interval (in minutes) between "regular" checks,
# i.e. checks that
# occur when the service is in an OK state or when the
# service is in a non-OK
# state, but has already been re-checked max_check_
# attempts number of times
normal_check_interval 5
# Time interval (in minutes) between non-regular checks
retry_check_interval 1
# Number of times that Nagios will repeat a check
# returning a non-OK state
max_check_attempts 3
# Enable service check parallelization for better
# performance
parallelize_check 1
# Enable passive checks
passive_checks_enabled 1
# Enable the event handler
event_handler_enabled 1
# Enable the processing of performance data
process_perf_data 1
# Enable retention of service status information across
# program restarts
retain_status_information 1
# Enable retention of service non-status information
# across program restarts
retain_nonstatus_information 1
# Enable notifications
notifications_enabled 1
# Time interval (in minutes) between consecutive
# notifications about the
# service being _still_ in non-OK state
notification_interval 120
# Time period during which notifications about this
# service can be sent out
notification_period always
# Service states for which notifications should be sent
# out (c=critical,
# w=warning, u=unknown, r=recovery, f=flapping, n=none)
notification_options w,u,c,r
register 0

```

Listing 9. *The definition of our service template*

```

/var/www/etc/nagios/generic-services.cfg
define service {
# Template name
name generic-service
# Services are normally not volatile
is_volatile 0
# Contact groups to notify about problems (or
# recoveries) with this service
contact_groups Admins
# Enable active checks
active_checks_enabled 1
# Time period during which active checks of this
# service can be made
check_period always
# Enable notifications
notifications_enabled 1
# Time interval (in minutes) between consecutive
# notifications about the
# service being _still_ in non-OK state
notification_interval 120
# Time period during which notifications about this
# service can be sent out
notification_period always
# Service states for which notifications should be sent
# out (c=critical,
# w=warning, u=unknown, r=recovery, f=flapping, n=none)
notification_options w,u,c,r
register 0
}

```

Listing 10. Proceeding to the definition of the actual services

```

/var/www/etc/nagios/services/services.cfg
# Secure Shell service
define service {
    use                generic-service
    service_description SSH
# Short name(s) of the host(s) that run this service.
    If a service runs on all
# hosts, you may use the '*' wildcard character
    host_name         *
    check_command      check-ssh
# This directive is a possible alternative to using the
    members directive in
# service groups definitions
    servicegroups     ssh-services
# Extended information
    notes              Availability of
                        the SSH daemon
    notes_url          http://
                        www.openssh.org/
    icon_image         ssh.png
    icon_image_alt     [SSH]
}

# Web service
define service {
    use                generic-service
    service_description WWW
    host_name          www1,www2
    check_command      check-http!/
                        index.html
    notes              Availability of the
                        corporate web sites
    notes_url          http://
                        www.apache.org/
    icon_image         www.png
    icon_image_alt     [WWW]
}

define service {
    use                generic-service
    service_description WWW
    host_name          mail
    check_command      check-http!/
                        webmail/index.html
    notes              Availability of the
                        web access to the mail server
    notes_url          http://

```

```

www.squirrelmail.org/
    icon_image         www.png
    icon_image_alt     [WWW]
}
[...]
```

Listing 11. Services can be grouped together with the „servicegroup“ directive

```

/var/www/etc/nagios/services/servicegroups.cfg
define servicegroup {
    servicegroup_name www-services
    alias              Web Services
# The 'members' directive requires a comma-separated
    list of host and
# service pairs, e.g. 'host1,service1,host2,service2,
    ...'
    members            www1,WWW,www2,WWW,
                        mail,WWW
}

define servicegroup {
    servicegroup_name dns-services
    alias              Domain Name Service
    members            dns1,DNS,dns2,DNS,
                        dns3,DNS,dns4,DNS
}

# The members of the following servicegroup are
# specified with the
# 'servicegroups' directive in the 'SSH' service
# definition
define servicegroup {
    servicegroup_name ssh-services
    alias              Secure Shell
                        Service
}
[...]
```

Now we can proceed to the definition of the actual services: see Listing 10.

Just like hosts, services can be grouped together with the `servicegroup` directive: see Listing 11.

Well, the bulk of the work is over now: the last step is configuring the web interface and then we will finally be able to set our Nagios server to work!

Listing 12. Reviewing how to create users in Apache

```

/var/www/etc/nagios/cgi.cfg
# Path to the main configuration file (relative to the
    chroot)
main_config_file=/etc/nagios/nagios.cfg
# Path to the directory where the HTML files reside
    (relative to the chroot)
physical_html_path=/nagios
# Path portion of the URL used to access the web
    interface
url_html_path=/nagios

# Disable context-sensitive help
show_context_help=0

# Enable authentication for the CGIs
use_authentication=1
# Uncomment the following directive to set a default
    user for unauthenticated
# sessions (strongly discouraged)
#default_user_name=guest

# The 'authorized_for_*' directives define a comma-
    separated list of
# authenticated web users who can:
# - view system/process information in the extended
    information CGI:
authorized_for_system_information=nagiosadmin,operator
# - view configuration information in the configuration
    CGI:
authorized_for_configuration_information=nagiosadmin,o
    perator
# - issue system/process commands via the command CGI:
authorized_for_system_commands=nagiosadmin
# - view status and configuration information for all
    services
authorized_for_all_services=nagiosadmin,operator
# - view status and configuration information for all
    hosts
authorized_for_all_hosts=nagiosadmin,operator
# - issue commands for all services via the command
    CGI:
authorized_for_all_service_commands=nagiosadmin

# - issue commands for all hosts via the command CGI:
authorized_for_all_host_commands=nagiosadmin

# Options for the Status Map and Status World CGIs
statusmap_background_image=smbbackground.gd2
default_statusmap_layout=5
default_statuswrl_layout=4
statuswrl_include=myworld.wrl

# Command to use when attempting to ping a host from
    the WAP interface
ping_syntax=/sbin/ping -n -c 5 $HOSTADDRESS$

# Time interval (in seconds) between page refreshes
refresh_rate=90

# List of audio files to play in the browser in case of
    problems. These files
# are assumed to be in the /var/www/nagios/media/
    directory
host_unreachable_sound=hostdown.wav
host_down_sound=hostdown.wav
service_critical_sound=critical.wav
service_warning_sound=warning.wav
service_unknown_sound=warning.wav
#normal_sound=noproblem.wav

# HTML and URL target options
action_url_target=_blank
notes_url_target=_blank
escape_html_tags=1

# Restrict users from changing the author name when
    submitting comments,
# acknowledgements and scheduled downtime from the web
    interface
lock_author_names=1

# Splunk integration options
enable_splunk_integration=0
#splunk_url=http://127.0.0.1:8000/

```

Listing 13. Apache configuration – 1

```
# openssl genrsa -des3 -out server.3des-key 1024
Generating RSA private key, 1024 bit long modulus
.....+++++
.+++++
e is 65537 (0x10001)
Enter pass phrase for server.3des-key: passphrase
Verifying - Enter pass phrase for server.3des-key:
    passphrase
# openssl rsa -in server.3des-key -out server.key
Enter pass phrase for server.3des-key: passphrase
writing RSA key
# openssl req -new -key server.key -x509 -out
    server.crt -days 365
You are about to be asked to enter information that
    will be incorporated
into your certificate request.
What you are about to enter is what is called a
    Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) []: IT
State or Province Name (full name) []: State
Locality Name (eg, city) []: Locality
Organization Name (eg, company) []: kernel-panic.it
Organizational Unit Name (eg, section) []: Information
    Technology
Common Name (eg, fully qualified host name) []:
    nagios.kernel-panic.it
Email Address []: nagios@kernel-panic.it
# chmod 600 server.key
# rm server.3des-key
# mv server.crt /etc/ssl/
# mv server.key /etc/ssl/private/
```

Listing 14. Apache configuration – 2

```
/var/www/conf/httpd.conf
ScriptAlias /cgi-bin/nagios "/var/www/cgi-bin/nagios"
<Directory "/var/www/cgi-bin/nagios">
    SSLRequireSSL
    Options ExecCGI
    AuthName "Nagios Access"
    AuthType Basic
    AuthUserFile /users/nagios.passwd
    Require valid-user
```

```
Order deny,allow
Deny from all
# Authorized clients
Allow from 127.0.0.1 172.16.0.13
</Directory>
Alias /nagios "/var/www/nagios"
<Directory "/var/www/nagios">
    SSLRequireSSL
Options None
AllowOverride None
AuthName "Nagios Access"
AuthType Basic
AuthUserFile /users/nagios.passwd
Require valid-user
Order deny,allow
Deny from all
# Authorized clients
Allow from 127.0.0.1 172.16.0.13
</Directory>
```

Listing 15. The -v option

```
# /usr/local/sbin/nagios -v /var/www/etc/nagios/
    nagios.cfg
Nagios 3.0.6
Copyright (c) 1999-2008 Ethan Galstad (http://
    www.nagios.org)
Last Modified: 12-01-2008
License: GPL
Reading configuration data...
```

```
Running pre-flight check on configuration data...
[...]
Total Warnings: 0
Total Errors: 0
Things look okay - No serious problems were detected
    during the pre-flight check
#
```

Setting up the web interface

Nagios doesn't have a specific client application to access the monitoring information; instead, it relies on the Apache (<http://httpd.apache.org/>) web server to provide a very simple yet powerful web interface, accessible via any browser and allowing users to access current status information, browse historical logs, create reports and, if so configured, issue commands to the monitoring daemon.

CGIs configuration

Nagios' web interface relies on a series of CGI programs written in C. The CGIs read their configuration information from two files: the main configuration file and `cgi.cfg`, located, by default, in the `/var/www/etc/nagios/` directory.

Below is a sample configuration file; pay particular attention when setting the `authorized_for_*` directives, because they allow you to assign special privileges to authenticated users and are, therefore, highly security critical. In the next section, we will review how to create users in Apache (see Listing 12).

Apache configuration

The web interface holds particularly sensitive information about network and services and may even allow the execution of commands that directly affect the monitoring daemon. As a consequence, it is strongly recommended that you configure authentication for accessing the CGIs.

User authentication files are managed with the `htpasswd(1)` (<http://www.openbsd.org/cgi-bin/man.cgi?query=htpasswd&sektion=1>) utility. *Note* that the first time you run this command, you must supply the `-c` option to create the password file:

```
# htpasswd -c /var/www/users/nagios.passwd nagiosadmin
New password: password
Re-type new password: password
Adding password for user nagiosadmin
# htpasswd /var/www/users/nagios.passwd danix@work
New password: password
Re-type new password: password
Adding password for user danix@work
#
```

An authenticated user whose username matches the short name of a contact definition is called an authenticated contact and is automatically granted access to information and commands for those hosts and services for which he is contact (please refer to the documentation (http://nagios.sourceforge.net/docs/3_0/cgiauth.html) for further details about authentication in the CGIs).

Well, now that we have Apache requiring users to authenticate, we should also configure SSL to avoid

sending passwords in clear text. Below are the `openssl(1)` (<http://www.openbsd.org/cgi-bin/man.cgi?query=openssl&sektion=1>) commands to create a self-signed certificate (a more detailed discussion about certificate management can be found here http://www.modssl.org/docs/2.8/ssl_faq.html; see Listing 13).

The last step is configuring Apache to actually require authentication and encryption to access the Nagios interface by adding the following lines to the `/var/www/conf/httpd.conf` configuration file: see Listing 14.

Running Nagios

Well, it looks like we're done with the configuration for now! Then we can make Nagios evaluate our hard work by invoking it with the `-v` option: see Listing 15.

If no errors were detected, then the long-awaited moment has arrived: we are ready to start Nagios! Though not before having created the directory for the lock file (*Note*: if you haven't rebooted since installing the Nagios packages, the `/var/run/nagios/` directory should already exist).

```
# apachectl startssl
/usr/sbin/apachectl startssl: httpd started
# install -d -o _nagios /var/run/nagios
# /usr/local/sbin/nagios -d /var/www/etc/nagios/nagios.cfg
```

You can check if everything is working fine by connecting to the web interface (<https://your.server.here/nagios/>) or taking a look at the logs (`/var/www/var/log/nagios/nagios.log`). To finish up, we have to configure the system to start both Apache and Nagios at boot time, by setting the `httpd_flags` variable in the `/etc/rc.conf.local` file:

```
/etc/rc.conf.local
httpd_flags="-DSSL"
```

and by adding the following lines to the `/etc/rc.local` file:

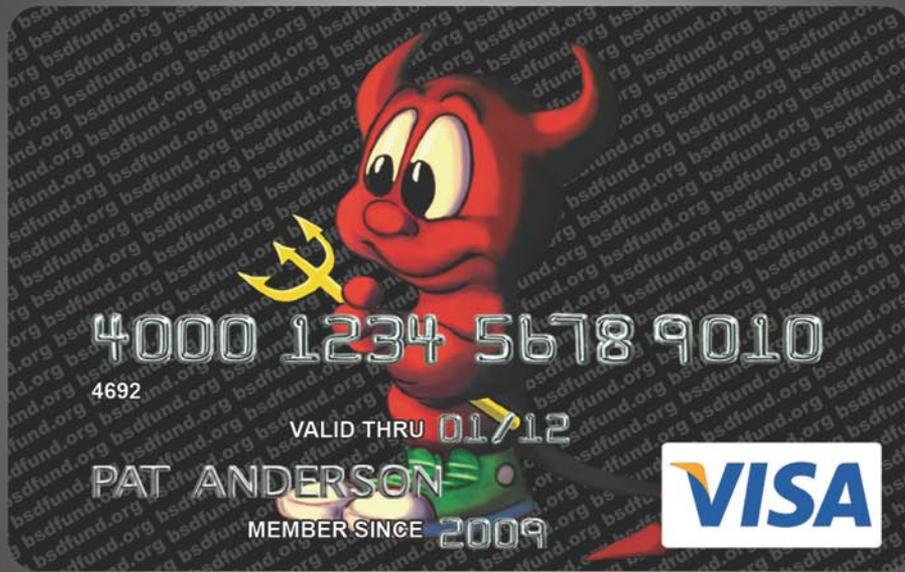
```
/etc/rc.local
if [ -x /usr/local/sbin/nagios ]; then
    install -d -o _nagios /var/run/nagios
    echo -n ' nagios'
    /usr/local/sbin/nagios -d /var/www/etc/nagios/
        nagios.cfg
fi
```

In the next chapter we will take a look at how to extend Nagios with some of its most popular addons.

DANIELE MAZZOCCHIO

Latest version: <http://www.kernel-panic.it/openbsd/nagios/>

Carry the card that supports BSD events around the world



BSD Fund is proud to sponsor of BSDCan 2010 and meetBSD California 2010 thanks to revenue from the BSD Fund Visa. A donation is made every time you use the card and simply charging your travel to an event can help sponsor that event.

BSD Fund also raises money through direct donations on behalf of BSD projects such as the pcc compiler.

Find our more at www.bsdfund.org

Replacing Microsoft Exchange Server

Step one: Installing Horde Groupware

Installing set of open-source programs without lack of functionality Instead of Microsoft Exchange Server. This way Groupware-part will be replaced on Horde Groupware.

What you will learn...

- how to install, configurate Horde
- changing tabs settings

What you should know...

- have an idea of Microsoft Exchange

Program without any analogue

So, exactly this name carried Microsoft Exchange for ages as for their supporters, as for their enemies on various Internet-forums. Any question for any search engine (*exchange replacing*) (exact phrase) usually generated at least 500 links. For my pity, most of these links carried one sentence – either replacing it with Communicate Pro [1] (good program, but is not opensource), or with Zimbra [2] (which is so tight binded with Linux, that guide for building Zimbra on FreeBSD reading as some terrible wizardry) [3]. Also, you can find some other programs with various stages or usability, self-made utilities... One of some variety of these utilites is a eGroupware [4], moreGroupware [5] and Horde Groupware Webmail Edition [6]. But, at bottom of any of these utilities guide you can find a remark – (*We are sorry, but when you will use our tool, you cannot do this task, and when use this tool – this task...*) Here I will give a short list of Exchange components and tell why both – their supporters and their enemies are calling it (*a program without any analogue*)

- SMTP-server, which task is to exchange mail with external servers
- POP3/IMAP/MAPI-server, which task is to exchange mail with internal users
- Groupware-server, which carried collaboration task (calendar, tasks, notes) also as Microsoft Outlook synchronization.

- Web-server, which allows access to Exchange mailbox from standard browser, called OWA – Outlook Web Access.
- *Active Directory* (AD) integration, allows taking data about mail users and contacts directly from AD.

It is not surprising, that when contrafact software was widely spread, Microsoft Exchange had reached this popularity – one program takes all (or quite all) tasks about communications. Replacing Exchange required at least 5 separate components to fully implement all functions, so we will replace Exchange consecutively – step one OWA and groupware will replaced with some opensource software, and step two – mail server will replaced with opensource projects – sendmail (for outgoing mailing) and dovecot (for incoming mailing, POP3, IMAP, shared folders accessing)

As a replacement for OWA and groupware server I have tried mentioned above eGroupware, moreGroupware and Horde Application Framework and selected Horde. Even though eGoupware seems to be a perspective development, due to the more simple installation for ordinary user without any programmer's skills, Horde looks more logical and better implemented, this makes it more understandable. That allows skilled user update it for their requirements pretty short time. Also, on selection Horde affected configuration management – any of

configs for Horde sets as typical UNIX-way – editing as text file. All of the parameters are transparent, but has only small quantity of comments.

Horde Groupware Webmail Edition (groupware-part from Horde Application Framework) takes three Exchange tasks – groupware-server (managing calendars, tasks and notes), accessing at mailboxes through browser and contacts with Active Directory integration.

But, as usual, main reason will be a price. And some time not only price...

Nokia with colorful display

At the beginning to be honest – first task wasn't about replacing Exchange. First task was simply to give an access to corporate mail from mobile box (as Nokia proudly called it), model Nokia N97. Also we had a Nokia N95 8G smartphone and cheap communicator Mitac MIO DigiWalker with Windows Mobile 5.x

A variant (*mobile VPN + Mail for Exchange*), which was offered by Nokia, was declined because it had no PPTP support, it had very complex setup, and required to spent many time to test some variants of using this program. Also this variant required hardware gateway, which not all user were in position to buy.

The only next variant – throw outside OWA, started on internal Exchange server through non-standard port and hardened this variant by SSL. Nothing was expected, that this small task would break Exchange.

I have skipped NAT setting details. To say only, that OWA runs on port 11222 as expected and we could see on communicator screen content of mailbox after required question about username and password.

Lucky and complacent, we type similar address on Nokia N97. And for long-time looked on line (*Internet: cannot connect to protected channel*) with a complete misunderstand – which (*protected connection*), https did not specified and SSL was not setup on server side!

After repeating this error on Nokia N95 with similar message, we have installed trial version of Opera Mini. Opera was more communicable and this moment we have stroked – authorization dialogue! Browser cannot display authorization dialogue, because OWA authorization does not dialogue with form, but does it with system procedures!

And latest kick on task to make an access through OWA was a phone call at Nokia technical support. Nokia support told about Mail for Exchange, about OWA and about this situation, that we cannot and did not access OWA – for Nokia best choice is Mail for Exchange :-)

A circle was finished. I do not like to discuss Nokia and

Microsoft relations, but we with our task was at one circle side, but OWA – at another side.

Welcome to Portal!

So, this way Nokia didn't show itself from the best side. So, we sigh for some time and start to working with Horde, when task is modified up to (*allow access to corporate mail with some way*). And, when we said (*corporate mail*), we should said (*calendar, tasks and notes*). So, using opensource groupware projects this way was very natural.

I will not describe in details a testing for groupware-software projects, which I have tried to use. I can tell only, that eGroupware has some potential, and it will be implemented, when eGroupware distributors understand, that contact list in groupware (especially, when authorization does through Active Directory) usually keeps in Active Directory. But now I have not detected at eGroupware any tools to pointing where we must take mail addresses and how to use it – only their own addressbook. MoreGroupware has similar flaw, and also has mostly primitive interface. So, that why I chose Horde Application Project.

At all, Horde contains many and many modules, which we do not need. Here is a photoalbums manager, bugtracker, file manager, bookmarks manager... I didn't need all of these functions and so I have searched Horde in ports. And, of course, I have found it. Quite unpleasant surprise was this – port was called `horde-meta`, not simply `horde` as similar ports – `kde`, `xorg` ...

```
# cd /usr/ports/www/horde-meta
# make
```

As any another good port, it had a configuration screen, where we can switch on or off some modules (see Figure 1).

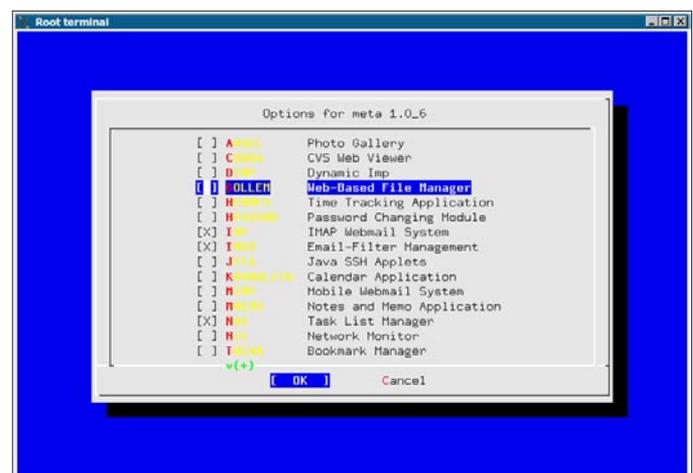


Figure 1. Configuration screen of horde-meta port

We need only some modules. Some another, of course, will be useful, but we need only:

- IMP – mail managing module. Managed an access to mailboxes on IMAP/POP3 protocol.
- MIMP – mobile version of IMP. Optimized for phones and communicators. Gives an access only for mailbox (so, when your browser recognized as mobile – we cannot access to contact list, calendar and tasks. But, usually mobile devices has their own)
- Ingo – mail filtering and spam protection module
- Kronolith – calendar/scheduler module
- Turba – address book module
- Nag – tasks managing module
- Mnemo – notes managing module

I should note, that Horde have a synchronizing server for mobile devices (smartphones, communicators) and Microsoft Outlook program with SyncML 1.1 or 1.2 version.

I was little bit disappointed, that documentation with good (or at least moderate) quality had missed. Neither

for Russian, nor for English. Only wiki on [7], where was done quite unsuccessful try to coordinate all things about setting up Horde. Of course, `conf.php` files has a big bunches of comments, some variables is self-named, some things are evident, about some another things I can read at mailing lists... But, at all as usual – to successful installation of portal you should be a programmer, preferable PHP.

Well, we are started.

We have only couple global requests for Horde – web server and database. We will use MySQL, but is possible using PostgreSQL also as 5 another DB, even Microsoft SQL. For more data about supported DB see `scripts/README`. Required also PHP with all required submodules. This (*all required*) is little surprise – here will OpenLDAP, mbstring and many and many another programs. More detailed list we can read in `ports` directory in `docs/INSTALL` file.

Before installing Horde itself, port will install a extremely huge pile of additional PEAR modules. It installs these modules, installs, installs and cannot finish... Even this install, you have not any guarantee, that installation checker did not find any missed module.

Setting our portal on port 18511 (Why? `18511 ->0x484F->"HO"` in literal). Testing address of our portal will be `http://horde.shelton.net`

Installation will finished without any error – we didn't do anything, which can throw an error. All was installed in directory `/usr/local/www/horde`. I insistently recommend do not touch content of this directory, but copy it at another place, when need. You can delete it, when you will finish your setup. I have made a virtual host this way: see Listing 1.

All files pathways will be shown from DocumentRoot, specified above.

Because portal at all and any their module as a part sets up with a simple edition of configuration files `conf.php`, there is no protection from incorrect setup (when at first place you set up some thing, but have forgotten set it up at second place). In setting up flow I had to reset my `conf.php` to intermediate copies at least ten times.

Starting installation. It drives for their own script (see Figure 2).

All menu items described wholly enough. Warning – do not use driver `mysql` (`mysqli`) – I do not know why, but their using leads to hand up portal immediately after starting – portal tries to load a login window and cannot finish this load. Use ordinary `mysql` driver.

Specifying database settings and create tables and user to connect to database to. I can explain only item 3 – Configure administrator settings. Old versions of Horde use auto-register as Administrator way to first login in

Listing 1. Making virtual host

```
Listen 18511
<VirtualHost 192.168.1.1:18511>
    ServerName horde.shelton.net
    ServerAdmin webmaster@shelton.net
    DocumentRoot "/usr/local/www/vhosts/horde/"
    ErrorLog "/usr/local/www/log/horde/httpd"
    CustomLog "/usr/local/www/log/horde/access" common
    <IfModule php5_module>
        AddType application/x-httpd-php .php
        AddType application/x-httpd-php-source .phps
    </IfModule>
    Include etc/apache22/extra/httpd-languages.conf
    <IfModule mime_module>
        AddType application/x-tar .tgz
        AddEncoding x-compress .Z
        AddEncoding x-gzip .gz .tgz
        AddHandler cgi-script .cgi
    </IfModule>
    <Directory "/usr/local/www/vhosts/horde">
        AllowOverride None
        Order allow,deny
        Allow from all
    </Directory>
</VirtualHost>
```

Portal. After 3.3.4 this way was changed on authorization by mail server – docs/INSTALL recommend edit file `imp/config/servers.php` to specify here address of your mail server, when does not installed on another box (where portal). This menu item allows setup logins, which will have Administrator rights at start.

I do not know, why they done this change – after clean install is more comfortable to auto-login as Administrator and switch to real authorization method, when all setup tasks will finished. For reverting first-time authorization method, you should edit `config/conf.php` (main Horde configuration file) similar this:

```
$conf['auth']['admins'] = array('Administrator');
$conf['auth']['driver'] = 'auto';
$conf['auth']['params'] = array('username' =>
    'Administrator');
```

Console actions finished. Probably, you should use it only for installing missed module.

Before starting setup, you should check on all required modules. Doing this by visit (in browser) on `http://horde.shelton.net:18511/test.php` and carefully read any text. Any message, colored with yellow – warning, you should read it, and correct any, pertained to our operating system, another ignore (i.e. when we will use MySQL, you can ignore warning about missed PostgreSQL). Any message, colored with red – errors. You should correct it fully, when you have any of errors, portal cannot run correctly. Usually, errors is missing some required components or important error in PHP settings.

After correcting all errors and related warnings, visit `http://horde.shelton.net:18511` (see Figure 3).

We will be auto-logged in as (*Administrator*). Mailbox now is closed – for running mailbox user must exist

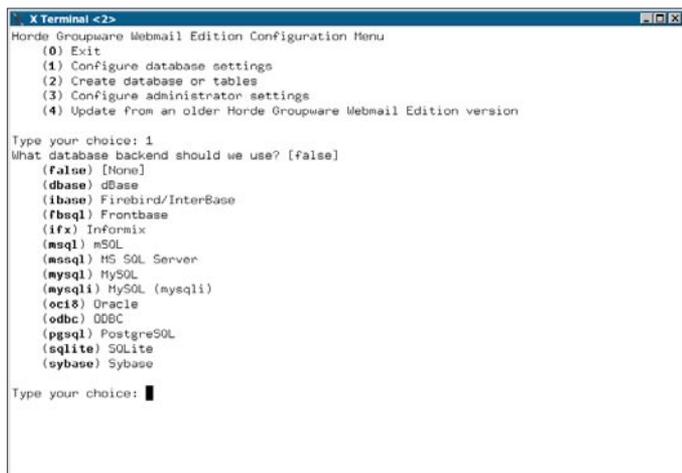


Figure 2. Horde Groupware Webmail Edition configuration screen

in mail server. Mail server still is an Exchange. But now we do not interest a mail, we need only system preferences.

Unroll item (*Administration*) and select (*Setup*). This is main place to setting up Horde parameters. Now, when only some required parameters are set, near all another items we can see line (*missing configuration*). It means that `conf.php` files for these modules still did not created. But at first time we will setup of portal itself.

Starting with *horde* module. We will see screen with huge setting tabs and you can ask – I really need to visit each tab? Of course, not, only 2/3 (see Figure 4) :-)

Tabs allocated quite in order to importance to visit it. Only (*Authentication*) tab skip and visit latest – when you will change authorization method on another, different from (*auto*), you immediately will see login screen. But set up a Horde – is a time-spent thing, and when you finished your configuration only partly, at next login you can see a frame with 4 error messages only. And you must be clear this setup, restore start configs and set up again... So, after each successful tab I recommend save a config in separate file. Below was described some parameters. When in description missed some tab – it does not need a change.

Tab General

Here will setup different common parameters

- `[tmpdir]` – here you can setup path for temporary files, when default is invalid for you
- `[server][port]` – here you mandatory must specified port number, when Horde installed on custom port
- `[cookie][path]` – this setting will correspond with path, where Horde will installed. When Horde was installed at root of VirtualHost, it must be `/`.

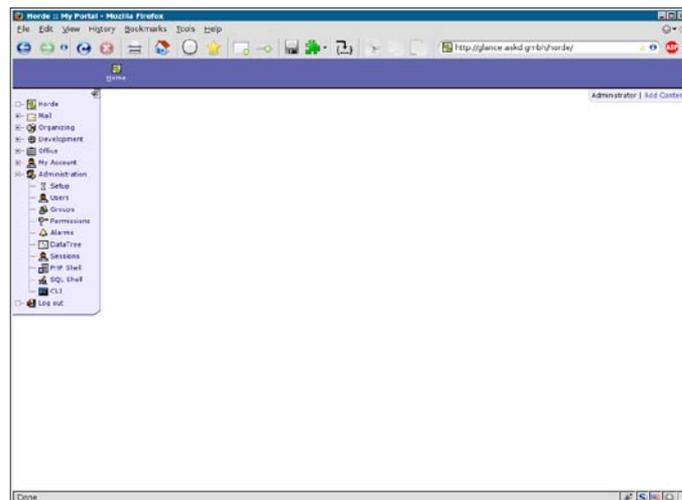


Figure 3. Main portal screen

Tab Database

Parameters on this tab is typical for selected database. For MySQL there are typical parameters – database name, user name, user password, address and port of database server. Beware! When you will use MySQL, select `mysql`, not `mysql (mysqli)`!

Tab Logging

Helps you when you will debug.

- `[log][enabled]` – switch on/off debug logging
- `[log][priority]` – specified detail level from panic errors up to debug messages
- `[log][type]` – specified type of debug logging. Have a huge pile of variants, from database up to window screen. Usually I prefer ordinary file on disk.

Another settings are understandable and commented, probably not so full, but sufficient

Tab Preference System

Here placed parameters of system, which keep personal settings for each Horde user. When you do not touch anything, settings wont be saved – it will used only for one session. But, this is one parameter which you need to change – `[prefs][driver]`. At default it set to (*PHP Session*), you must change it to (*SQL Database*) (or some another from list)

Tab Alarm System

`[alarms][driver]` – you must setup, how to keep alarm messages. There are only 2 variants – do not keep and keep in database.

Tab DataTree System

DataTree – is some tree-like structure, used by portal itself to keep data instead of database, i.e. I do not know, where I can use it, but it exists – so, probably it is needed for some tasks. Parameter `[database][driver]` specified, where will be this data kept. Exactly, list has only variants (*yes*) or (*no*), so you can only keep this data or not.

Tab Groups

`[group][driver]` – specified, where we will keep groups, created inside of portal. This is really internal groups, created in *Administration>Groups*.

Tab Permissions

`[perms][driver]` – specified, where will keep data about permissions, created inside of portal through *Administration>Permissions*. Minimal required rights for run portal will be described in the next part of article.

Tab Shares

Here will setup parameters for user objects. Portal users can grant access for some their objects (calendars, tasks), and these parameters setup how it will keep.

- `[share][any_group]` – when is set, user can grant access for any another user, when not set – only for users from their group
- `[share][driver]` – specified driver to keep share resources data

Tab Lock System

`[lock][driver]` – specified using object locking

Tab Mailer

Specified parameters for sending mail. Parameters set are different for using sendmail and for using another server. For using another server you should specify `[mailer][params][host]` and `[mailer][params][localhost]` – domain name for generating address, for using sendmail – `[mailer][params][sendmail_path]`

Tab Virtual File Storage

Customize the virtual file store. Used only if a further set Golem – a file manager.

- `[vfs][type]` – sets the driver VFS. In the simplest case – to keep the file system
- `[Vsf][params][vfsroot]` – sets the path to the root of the VFS, in case of storage in the file system

Tab HTTP Proxy

Specifies a set of fairly obvious settings to access the Internet if you are using a proxy server.

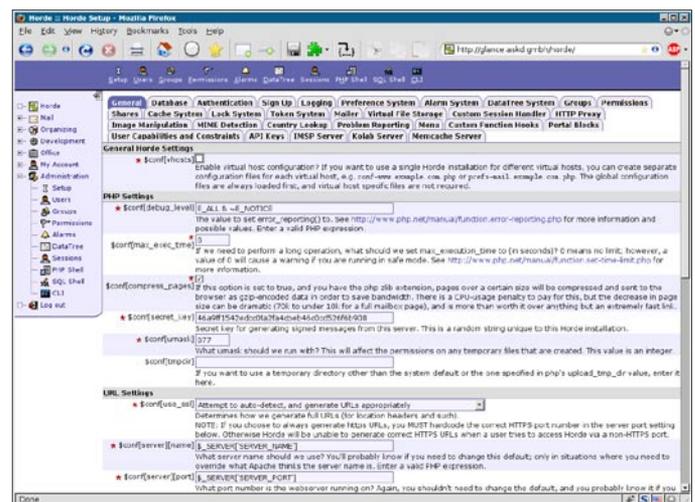


Figure 4. Portal settings

Tab MIME Detection

The only setting specifies the path to the database MIME magic. For FreeBSD, this is usually `/usr/share/misc/magic`.

We save the configuration of the portal. Yes, we were not on the tab *Authentication*. Not so fast there. If you enable authentication, not creating a configuration of modules, there is a risk immediately after the restart to see two windows with error messages about PHP instead of portal, since horde absolutely does not check the created configuration.

Go to each of the modules in turn and simply press *Save*. There is no need to change anything, and there are settings there – each module on a tab, except for the *Imp*, which on the first tab, you can optionally configure the path to *aspell* program and *gpg*.

Save the latest changes in the configuration, make a backup copy of `config/conf.php` and set up the authorization. Authorisation in horde can be configured in many ways, but we are only interested in one – authorization by the server Active Directory, so I will described only this set of parameters.

Tab Authentication

- `[Auth] [admins]` – specifies a list of accounts that are separated by a comma, which would have administrative rights. These accounts must exist in Active Directory
- `[Auth] [driver]` – sets the authentication mechanism. To use Active Directory, choose "LDAP authentication"
- `[Auth] [params] [hostspec]` – specifies the address of the server authentication
- `[Auth] [params] [basedn]` – sets the server's root DN, which is usually equal to its domain name.

For example, the domain `shelton.net` it will be equal `dc = shelton, dc = net`

`[Auth] [params] [binddn]` – specifies the user name from which you will connect to the server LDAP. Since Windows 2003 does not support anonymous connections, you must have some users who can not use any resources, and connect on his behalf. As a parameter to specify the complete name of the `CN`. For example, the user `ldapread`, located in the `OU Other Users`, the value will be equal to `CN = ldapread, OU = Other Users, DC = shelton, DC = net`

- `[Auth] [params] [password]` – sets the user password to connect to LDAP
- `[Auth] [params] [version]` – check that is LDAP v3, otherwise will not work
- `[Auth] [params] [scope]` – specifies the search area must be given *Subtree search*
- `[Auth] [params] [ad]` – specifies that the server that performs authentication is the Active Directory server
- `[Auth] [params] [uid]` – specifies the name of the attribute containing the user ID. In this field you must specify *samaccountname*
- `[Auth] [params] [encryption]` – specifies the type of password encryption in its verification. Specify the *plain* or *msad*
- `[Auth] [params] [filter_type]` – specifies the type of filter used for the selection of accounts for verification of authorization. Select *A complete LDAP filter expression*

Conclusion

`[Auth] [params] [filter]` – sets the proper filter for selection of accounts for verification of authorization. The filter needs to accelerate the authorization to not use the accounts groups, system objects, etc. A simple filter looks like this – `(& (sAMAccountName =*) (mail =*))`, which means to select all objects that have defined the field `sAMAccountName` and `mail`.

Well, the portal is installed and is already able to check your account and password on the server Active Directory. But still have only a small part of the work, the main thing – access to email and global address book, sync with Outlook – is waiting for us ahead

RASHID N. ACHILOV

FreeBSD integration specialist, at the moment working as telecommunication systems administrator.

<http://openoffice.mirahost.ru/>

References and Further Reading

- <http://www.communicate.com> – The home of Communicate Pro
- <http://www.zimbra.com> – The home of Zimbra Collaboration Suite
- <http://pcbsd.org/~dwhite/zimbra/> – An article about how to build Zimbra on FreeBSD
- <http://www.egroupware.ru/> – The home of eGroupware
- <http://www.moregroupware.de/> – The home of moreGroupware
- <http://www.horde.org/> – Site Horde Applications Framework
- <http://wiki.horde.org/> – Wiki on Horde Applications Framework

Maintenance Systems over BSD

I was talking in previous articles about how to run applications widely used in the Industry that can be supported by BSD apart of classical IT services.

As clear example of this is SAP Suite. SAP covers all possible asset management to control the cost related to production and also maintenance but as per tighted cost in investments today, the Plants must run 24/7 with maximum reliability and productivity possible. To achieve this goal, raise several technologies that support the maintenance activities as can be viewed in Figure 1.

All of these technologies were supported by software applications but most of cases run under windows because is (up to moment) the standard or at least the most widely used. In order to don't extend the article and because is not intended to do it, I will detail only one example of these programs and his features.

My example will be an *On-line Machine Condition Monitoring*. This system was develop by SPM Instrument AB and his preferred choice for this kind of systems, because many aspects but most important is the flexibility.

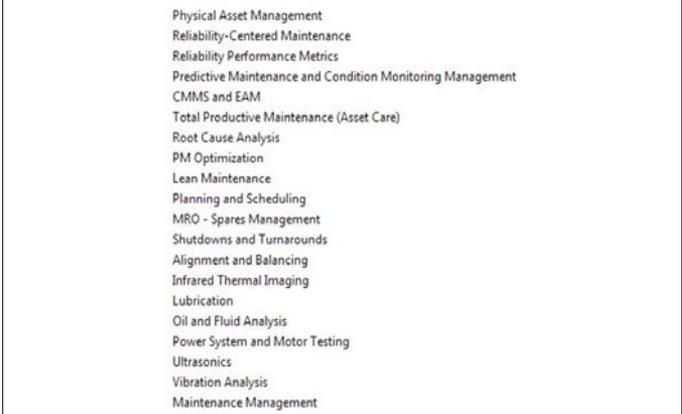
The CMS System is a modular, software controlled on-line machine condition monitoring system with automatic data evaluation. It is applied by leading industries all over the world for early fault detection, to avoid production losses through unplanned downtime, and to reduce the overall costs for maintenance.

The CMS System contains four types of measuring units, each with a specific task. This allows you to select the most cost-effective installation that meets your technical requirements:

Shock pulse measurement on rolling bearings supplies data on bearing damage, lubrication condition and the effects of alignment and load. In many applications, the bearings are the only machine elements which need monitoring.

Vibration severity measurement is the ISO recommended method for general condition monitoring. It detects the most common mechanical faults, such as unbalance, structural weakness and loose parts.

Analog signal monitoring is used to correlate data on flow, effect, pressure, temperature, etc., input as analog voltage or current signals, with the shock pulse and vibration measurements.



- Physical Asset Management
- Reliability-Centered Maintenance
- Reliability Performance Metrics
- Predictive Maintenance and Condition Monitoring Management
- CMMS and EAM
- Total Productive Maintenance (Asset Care)
- Root Cause Analysis
- PM Optimization
- Lean Maintenance
- Planning and Scheduling
- MRO - Spares Management
- Shutdowns and Turnarounds
- Alignment and Balancing
- Infrared Thermal Imaging
- Lubrication
- Oil and Fluid Analysis
- Power System and Motor Testing
- Ultrasonics
- Vibration Analysis
- Maintenance Management

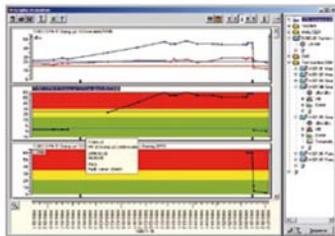
Figure 1.

Vibration monitoring with spectrum analysis allows you to target specific fault symptoms and get a machine specific condition evaluation.

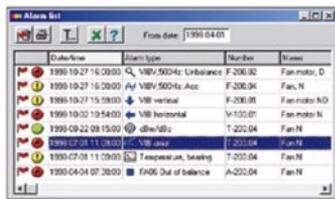
The core of the CMS System is the SPM software, *Condmaster@Nova*. This fifth generation program receives the measuring results from all SPM condition monitoring devices for evaluation and presentation.

Based on extensive empirical data, international standards and machine statistics, the evaluation result is an easy to understand colour code, highlighting potential trouble spots. By calibrating and adjusting limit values, you can tune the automatic evaluation process with great precision and get an immediate, reliable diagnosis.

A CMS System can contain up to 240 measuring units for bearing condition and vibration severity. They are



Your main tool for fast fault detection is the alarm location panel which shows the trouble spots. Import your own picture for perfect overview. Click on a point to get the details: development curves, trends, result lists, comments and more.



Date/Time	Alarm type	Hour/Day	Name
1999-10-27 18:20:00	Vib/500Hz: Unbalance	F-200:02	Fan motor, D
1999-10-27 18:20:00	Vib/500Hz: Acc	F-200:04	Fan, N
1999-10-27 15:58:00	Vib-vertical	F-200:02	Fan motor KD
1999-10-22 19:54:00	Vib horizontal	V-100:01	Fan motor KI
1999-09-22 09:15:00	dBm/40z	T-200:04	Fan, N
1999-07-08 11:08:00	Vib axial	T-200:04	Fan, N
1999-07-08 11:09:00	Temperature, bearing	T-200:04	Fan, N
1999-04-04 07:38:00	FAOS Out of balance	A-200:04	Fan, N

The alarm list is generated on the basis of automatically applied evaluation rules which can be modified by customer defined limit values. You can also program alarm delay conditions, log all alarm messages on a printer, and connected any or all measuring channels to the main alarm relay.



An Evam[®] spectrum highlights the selected fault symptoms and states their velocity value in relation to overall machine vibration.

Figure 2.

If you wish to contribute to BSD magazine, share your knowledge and skills with other BSD users – do not hesitate – read the guidelines on our website and email us your idea for an article.

Join our team!



Become BSD magazine Author or Betatester

As a betatester you can decide on the contents and the form of our quarterly. It can be you who read the articles before everybody else and suggest the changes to the author.

Contact us:
editors@bsdmag.org
www.bsdmag.org

supplied with 230 or 115 Vac and connected in series via the data cables in one or two LAN networks. A system unit with alarm relay links the measuring units with a PC. Up to 9 such systems can be handled by the software Condmaster®Nova.

For remote monitoring, use a connection via modem. Up to one week measuring results are stored in the units, to be recalled at suitable intervals.

All settings are made from the PC. Measuring interval and sequence are set individually for each unit, alarm limits and relay connection for each channel.

BMS units for bearing monitoring measure shock pulses on 16 channels. The signal is pick up by shock pulse transducers, installed on the bearing housings and connected with coaxial cables.

VMS units for vibration severity monitoring the RMS value of vibration velocity on 8 channels. They have four 4 relays to steer external alarm devices. Four control channels enable vibration measurement, e. g. to avoid measuring at critical speeds.

An AMS board with 16 channels can be installed in each BMS or VMS unit, to monitor voltage or current lines carrying analog data on any customer defined quantity. Each unit can also be equipped with an RPM boards with four channels.

Vibration analysis with EVAM® VCM-20 units are measuring computers for vibration analysis in frequency ranges up to 20 000 Hz. A unit has 8 or 24 vibration channels and 8 rpm channels, allowing synchronous and asynchronous measurement. VCM-20 units are

connected via PC network. Measuring results are analysed and stored locally in the VCM unit. The channel configuration and the measuring assignments are set up in Condmaster®Nova.

Measuring units have stainless steel housings and sealed cable connections. The transducer lines are monitored for transmission quality and electric faults. Transducers, cables, connectors and other installation equipment are high quality products, designed for harsh industrial environments.

Condmaster®Nova is SPM's universal condition monitoring program, used for hand-held data loggers as well as on-line systems. It operates under several Windows versions and uses SQL Server as a database handler.

Purely administrative data is kept at a minimum – you can set up one measuring point for as many as 9 different monitoring tasks, including two free values, user defined measuring functions. You work with your familiar administrative machine data and simply instruct Condmaster®Nova to accept your name and number formats.

The expert knowledge needed to evaluate machine condition is integrated in the program: a complete bearing catalogue, lubricant data, bearing life calculation, the SPM evaluation rules, the ISO limit values, mathematical models for spectrum analysis and fault symptom detection, and much more.

You set measuring time and measuring sequence, select the values to be stored and define the alarm condition. Via relay connections, you can control external alarm devices and automatic shut down.

You only activate the measuring functions you need, and automatically blank out all others. Thus, you can work exclusively with the CMS System, but you can also activate the functions for SPM data loggers and other hand-held devices.

Is a perfect tool for efficient maintenance, for required input data, you get instructive menus, default values and on-line help texts. You have copy and edit functions to save time when you register machines and measuring points.

The most powerful part is EVAM® – Evaluated Vibration Analysis Method. It is much more than the normal spectrum analysing product. In addition to 9 general condition parameters, you can select fault symptoms for special analysis and work with machine specific evaluation criteria.



Figure 3.

JOSEBA MENDEZ

HAKING

PRACTICAL PROTECTION

APC
by Schneider Electric

PROTECT YOUR COMPUTER,
THE ENVIRONMENT, AND YOUR WALLET

HAKING

PRACTICAL PROTECTION HARD CORE IT SECURITY MAGAZINE

MOBILE EXPLOITATION

PRIVACY KEEPING AND EXPLOITATION METHODS

EXPLOITING NULL POINTER DEREFERENCES
MOVEMENT ON THE MOBILE EXPLOIT FRONT
METHODS OF SECRECY
BRUTE FORCING USER NAMES
DATA MINING AS A TOOL FOR SECURITY

**MOBILE WEB:
PRIVACY KEEPING AND
EXPLOITATION METHODS**

**INTELLIGENCE REPORT:
ANALYSIS OF A SPEAR
PHISHING ATTACK**

**VIDEOJACKING:
HIJACKING IP VIDEO CALLS**

APPLICATIONS ON THE CD 

CERTIFIED WIRELESS NETWORK
ADMINISTRATOR TRAINING BY SEQRIT.ORG
DOUBLE ANTI-SPY PRO TRIAL

Vol 5, No 2 Price USD 14.99
Issue 2/2010(27) ISSN: 1733-7186



PLUS

A LOOK AT THE MALWARE TRENDS
EXPECTED IN 2010 BY JULIAN EVANS



IT SECURITY MAGAZINE

Low Resource PCs with FreeBSD

FreeBSD is my pick for best modern operating system to use on older PCs. I can't believe how many used PCs end up as landfill while students, educators, low income families and others go without a computer at all.

Just think of how many people could have their very own PC if some of those old machines that someone believes are no longer useful could be updated with a brand new operating system like FreeBSD. Also, if it runs well on older, low resources PCs, imagine how well it can do on a more high-powered machine. Why is FreeBSD my top choice for older machines? It ran the programs I needed more efficiently than Linux and other Open Source operating systems. It's an interesting path that brought me to FreeBSD. If you'd like to read more you can check my web site at <http://www.distasis.com/cpp/slin.htm> for further information.

I had tried FreeBSD many years ago and the one reason I hadn't continued with it was that I could never get X Windows working. After trying several other Open Source operating systems, I didn't care if X Windows worked, I just wanted a fast, stable system. I checked the specifications and it looked like FreeBSD would load in 64 MB RAM which is what I have. Why not give it a try? It was fairly easy to get the basics up and running, but there are only so many command line, curses and slang based programs out there. I still wanted to run a few favorite GUI applications. I had heard about running programs like mplayer with Framebuffer support in Linux. However, FreeBSD doesn't offer that ability. The closest equivalent I could find was kg4BSD.

I also found out SDL, ssvglib and WxWidgets libraries all offer compilation options that avoid running on top of X-Windows.

SDL is supposed to run on top of VGL, ssvglib or directfb as alternatives to X. GGI and Nano-X are also supposed to be alternatives to X Windows. I did my best to try to build alternate GUI libraries such as SDL with something other than X Windows, but I just couldn't get it to work. In the end, the only graphical program I was able to run via the command line was zgv which uses ssvglib. I decided, once again, to have a go at getting X Windows to work. Armed with two copies of my xorg.conf files from previous Linux installations, I tried to generate a file that would get X Windows running on FreeBSD. It took me a day and a lot of looking up details in the forums as well as referring to those files to get X Windows finally working. I highly recommend the April 2010 BSD Magazine article *X11 without dbus/hald and with three kings* which illustrates some of the tips I needed to make my system work.

Once I had X Windows, it was time to choose what to run on it. If you're using a low resource system with little memory and/or hard drive space, the programs you run can make or break your experience on the computer. Many people believe finding a lightweight window manager is one of the keys to making an older machine useable. If the window manager and

desktop programs take up too much memory or other resources, you won't be able to get other programs to run efficiently. There are several window managers out there to choose from. I happen to like three, Fluxbox, Openbox and dwm.

If you look at the source code for dwm, you'll see how compact it is. This is a great window manager for people who use the keyboard over a mouse or touchpad. If you have a background with C/C++ programming, you'll like the way you customize it. The settings are actually added as part of the code. You do need to recompile and link the program each time you modify settings. There are several examples of customizations for dwm available if you search the Internet. While I wouldn't want to use dwm with whatever settings it comes with, once customized, it can make an effective and efficient window manager. I tried to compare it to Fluxbox and Openbox based on how much memory they use. I ran conky for my measurements, but found that dwm doesn't seem to like conky and CPU usage goes up very high when both are running at once. If I run top to check memory usage, things look more normal. I also tried lxtask to check memory, but the FreeBSD package didn't appear to be working properly. Am hoping to try rebuilding it from source when I have more time. From my measurements, dwm uses the least memory of the three with Fluxbox coming next and then Openbox.

Fluxbox and Openbox have a lot in common. They were both based on Blackbox. I saw an interesting thread on one of the forums that listed differences between the two. For me, there are 4 differences I notice when using them. Openbox developers try in general to use as many standards (such as XML) as possible. Openbox has even been completely rewritten from its original Blackbox fork. Both window managers store customizations in text files rather than needing you to compile the information with code like dwm requires. Openbox uses the XML format for its text files. XML is designed to make it easy for a program to parse, but to do so, the files must follow a strict format. My first impression was that its easier to mess up the format with the Openbox XML settings files than it is to mess up the Fluxbox settings files. You can set up both Fluxbox and Openbox to switch between applications when you press a key combination such as alt-tab. I often use that feature on Windows. However, Openbox seems to do it more smoothly. It even brings up a dialog showing the applications you can switch between. I also noticed keyboard navigation of menus appears smoother to me in Openbox. The final difference is that Fluxbox appears to use 1 Meg of memory less than Openbox.

That's despite the fact that it includes a taskbar and Openbox does not. If you're still not sure which window manager you want to run, a nice feature of both Fluxbox and Openbox is that you can call and switch to other window managers right from their menus.

If you're going to use Fluxbox or Openbox, you'll want a helper program to draw the screen background or you'll start noticing some weird things going on when the screen is supposed to redraw. Both hsetroot and xsetroot are in ports and can be used with these window managers. I've also used feh with Fluxbox. As well as letting you set the background, it's a helpful, lightweight image viewer. Fluxbox's fbsetbg wrapper program can usually find feh if it's available. I've seen some posts on setting up Openbox to use feh, but I don't believe the support is as automatic as it can be with Fluxbox. You can make use of these types of programs to set and redraw backgrounds or background images with dwm as well as Fluxbox and Openbox.

If you like a desktop environment, with extra programs to make things easier and themes to coordinate your application colors, there are tricks to accomplish it even on low memory systems. I found Fluxbox relatively easy to customize by editing the configuration files in a text editor. However, Openbox was a bit harder. You can download and use obconf to configure colors, themes and other basic settings.

Dwm is strictly no frills and doesn't even include a menu. However, there's a lightweight program, dmenu, from the developers of dwm that does the trick and integrates well with dwm. You can also use it with other window managers. I didn't find a lot of documentation on dmenu, so if you want to try it out, be sure to check out the sample scripts that come with it. Once I experimented with it, I was able to create a cascading menu where I brought up a list of application categories and called dmenu again to show the applications in each category.

If you're switching from Fluxbox to Openbox and miss the taskbar, there are several standalone replacements. One I found useful and highly customizable was Tint 2. Personally, I don't miss the Fluxbox taskbar, but I do miss the clock feature that's part of the taskbar. Both Fluxbox and Openbox have several programs called dockable apps that run well with them. I sometimes use the wmfishtime dockable app as a clock replacement in my window manager. The FreeBSD port of wmfishtime uses *GTK+ 1*. With limited hard drive space, I do my best to cut down on the number of screen libraries I need to install on my system. If you check the Debian ports on the Internet, you'll find patches for wmfishtime

that upgrade it to *GTK+ 2* and add a digital clock feature I really like.

I also wanted to mention LXDE, the Lightweight X11 Desktop Environment. It uses Openbox, but provides several tools to help create the effect of an integrated desktop environment. LXDE is the lightest desktop I've been able to find. I haven't installed all of it, but I do use parts with Openbox and am able to run it all in 64 MB RAM. LXDE adds interesting programs like *lxtask* and *gpicview*.

For those who mainly run *GTK+ 2* based programs, you can color coordinate your applications and give them a similar look and feel. Programs like *GTK+ 2.0 Change Theme (gtk-chtheme)* make that task easy. I've used it to pick a color theme I like, such as *Crux* and to set the font to a more readable size.

It seems like I mainly use my window manager to navigate easily between all the terminal windows I have open. So naturally, choosing a lightweight terminal emulator was important to me. Most lightweight Linux distributions default to a program like *rxvt* over the more resource intensive *xterm* that's usually a default. I saw a couple of benchmarks documented on the Internet that seemed to back up that information and gave other alternatives as well.

When I read that some terminal programs let you run multiple instances and share the same process to reduce memory and save resources, I decided that was the way I wanted to go. The drawback, of course, was that if one instance crashes, they all could. I had read that *Sakura* and *lxterminal* (another LXDE offering) which are VTE based terminal emulators had the type of features I wanted. However, I didn't have all the dependent libraries I needed on my system to get them going. So, I was pleasantly surprised when I found out *urxvt* also had this feature and, of course, it required less library dependencies.

I run `urxvtd -q -f -o &` once from my `~/.xinitrc` file when I start up X Windows. Then, any time I need a new terminal, I run `urxvtc` with various settings to customize colors, fonts and scrolling. I added the following entry to my window manager's menus:

```
urxvtc -sl 2048 -bg grey65 -fg black -sr -fn 10x20
```

You can find *urxvt* in FreeBSD ports if you look up `rxvt-unicode`.

Since I use the console so often, I like to customize the colors. When I log in, I have a blue background and white font that's easier on the eyes than the default black and white. I've added the following commands to my `.profile` file:

```
export COLOR="\[033[0;37m\]\[033[44m\"
export PS1=$COLOR"|\d \T@| \w | >"
```

I usually have a blue background for my window manager as well, which makes it harder to see my terminal windows. Since *bash* is my default shell, I set the following in my `~/.bashrc` file which is checked once every time *bash* is started:

```
export PS1="|\d \T@| \w | >"
```

That makes sure my *urxvtc* terminal session keeps the colors I asked for via the command line when I call it. If you're using a different shell program, set the environment variables in the appropriate settings files. If you want to change your shell to *bash* like I did, make sure your *EDITOR* environment variable is set to an easy to work with console editor like *pico* and run the *chsh* program to change the shell default.

As I mentioned, I like to have several terminal windows open at once in my window manager. I also like to cut and paste between them and my programming editor. I thought cutting and pasting between Command prompts was a bit of a nuisance in Windows, but it is doable. It's even harder to cut and paste between terminals and other programs in X Windows. Applications may use the clipboard or a cut buffer. I still haven't figured out how to emulate the cut and paste keys I'm used to from Windows, but I can get my terminal emulator to work with my programming editor by using a program called *autocutsel*. I place *autocutsel* & in my window manager initialization file so that the program gets run once before startup and stays in the background.

These are some of the more basic programs I use everyday with FreeBSD. However, I've only scratched the surface on some of the great lightweight programs out there.

LAURA MICHAELS

Laura Michaels is a software engineer. She has written articles for magazines such as C/C++ Users Journal and Dr. Dobbs. Her website is at <http://www.distasis.com>

Looking for help, tip or advice?
Want to share your knowledge with others?

Visit BSD magazine forum



Give us your opinion about the magazine's content
and help us create the most useful source for you!

www.bsdmag.org

Making the

Unknown Giant Visible and Known

FreeBSD has the moniker Unknown Giant. I confirm that it is true in my place. I have asked system administrators, computer enthusiasts, and hobbyist about FreeBSD and they didn't even know what I'm talking about.

The only thing they knew and work with were Windows and Linux. I am not a fan of flame wars. I don't want to bash other operating systems, because at their own, they have their specialties, features, and weaknesses.

I understand that it is hard to introduce the *Unknown Giant* to system administrators because they are accustomed to using the systems which they are familiar with.

I know the slogan, *FreeBSD: Quality vs. Quantity*. With this, I helped the *Unknown Giant* become visible and known to some of my friends. Here is my story.

I once visited an old friend of mine. He is a system administrator at a small organization near my area. His network services were provided by Windows servers. He has DHCP server, ISA server, Exchange server, and SQL Server running in his network. Curious, I asked him what his common problems were in administering his systems. He said, *Well, I'm happy the way my network runs and I don't get that much problems except for that Conficker thing*. Conficker, also known as, Win32/Conficker.B is defined by the Microsoft Corporation as being a worm that infects other computers across a network by exploiting a vulnerability in the Windows Server service (SVCHOST.EXE). It adds that if the vulnerability is successfully exploited, it could allow remote code execution when file sharing is enabled. It also states that Conficker may also spread via removable drives and weak administrator passwords. And it explains that Conficker disables several important system services and security products.¹ He said that in an event of a Conficker spread, or other worms, he just restores his image backup and the system is up once again. Well, the main problem with that is the time it takes to restore the image, and of course, network downtime. While the server is being setup and configured, the DHCP and other services are down, and so networking is down also.

So I told him that an operating system named FreeBSD could be installed in less than 20 minutes and run network

services which are not platform specific such as DHCP service and does not require gigabytes of memory. He couldn't believe such a thing was possible since it took him hours to do fresh install of a Windows server, and even some Linux distributions. At that point I said, *If you would let me, then I will demo it*. He gave me an old AMD Athlon computer with a 128MB of memory and watched me. I had the FreeBSD 8.0-RELEASE i386 disc 1 with me, and I started the installation. In less than 15 minutes, the base system was installed. After that, I downloaded and installed the package ISC DHCP 3.1 Server, edited the configuration file and connected it to his network. All in all, it took us 18 minutes to setup and run a DHCP server.

He was amazed at what I accomplished in such a short time. From then on, he studied the FreeBSD operating system through the BSD handbook 2 (<http://www.freebsd.org/doc/en/books/handbook/>), read the BSD Magazine, and slowly migrated some servers to FreeBSD. He also spread the word to his friends who are system administrators as well who are all willing to try FreeBSD.

It is hard to advocate system administrators and computer users to use FreeBSD. But slowly, each one of us can do small things that will make the *Unknown Giant* visible and known without sacrificing quality over quantity and assist other users to understand their systems better.

JOSHUA EBARVIA

Joshua Ebarvia is a java programmer, systems administrator and college lecturer. His passion is working and using operating systems specially UNIX-based and UNIX-cloned systems. You can reach him at joshua.ebarvia@gmail.com

MAGAZINE

BSD

In the next issue:

- Network Monitoring With Nagios and OpenBSD (PART2)
- BSD and Linux

Next issue is coming in September!

Customize Your Server Flavors

Made to Order
Server Systems

High-Density
Storage Solutions

Intel® Xeon®
5600/5500 Series
Processors

Open Source
Configurations



Served **Exactly** How You Like!

Tired of being able to choose from only **chocolate**, **strawberry**, or **vanilla**? At iXsystems, we understand your need for custom-made servers.

“Open Source Hardware Design” is the iXsystems trademark. iXsystems provides an assortment of pre-configured servers and storage solutions, but our true pride rests on our ability to customize our products to meet your specific tastes and needs. iXsystems mixes in the raw power of Intel® Xeon® 5600/5500 Series Processors for a truly delicious treat. Our Professional Enterprise Service Level packages and desktop support offering also enables us to ensure you get the most from your FreeBSD® and PC-BSD™ systems, adding the perfect toppings to your order.

Call iXsystems toll free or visit our website today!

+1-800-820-BSDi | www.iXsystems.com

