

The complete guide to installing, configuring and getting the most out of Nagios.

# The NagiosBook



Chris Burgess

---

# **The Nagios Book**

Chris Burgess

Copyright © 2005 Chris Burgess

The following material contained in this document cannot be reproduced in any form without prior written permission. Nagios and the Nagios logo are registered trademarks of Ethan Galstad. Other trademarks mentioned within this document are the property of their respective owners.

---

---

# Table of Contents

Preface .....	vi
1. Introduction .....	1
1. Welcome to the World of Nagios .....	1
2. Useful Nagios Resources .....	1
3. Other Tutorials .....	1
2. What is Nagios? .....	2
1. About Nagios .....	2
2. Other Monitoring Options .....	3
3. Why Use Nagios? .....	3
4. Nagios Demo .....	4
3. How Nagios Works .....	5
1. A Typical Scenario .....	5
4. Preparing for Installation .....	6
1. Getting Ready .....	6
2. Selecting an Operating System .....	6
3. Setting Up The Server - FreeBSD .....	6
4. Useful FreeBSD Commands .....	8
5. Installing Apache .....	8
5. Installing Nagios .....	10
1. Obtaining Nagios .....	10
2. Knowing Where Nagios Lives .....	11
3. Telling Apache about Nagios .....	12
6. Configuring Nagios .....	13
1. Editing Config Files Manually .....	13
2. Configuration Front Ends .....	13
3. Getting Started .....	13
7. Starting Nagios .....	22
1. Making Sure Nagios Starts Automatically .....	22
2. The Pre-Flight Check .....	22
3. The -s Switch .....	23
4. Help! .....	23
8. Nagios Plugins .....	24
1. The Standard Plugins .....	24
2. The Contrib Plugins .....	24
3. Other Plugins .....	24
4. Writing Your Own Plugins .....	25
9. Monitoring Remote Unix Servers .....	26
1. NPRE .....	26
10. Monitoring Remote Windows Servers .....	27
1. NSClient .....	27
11. Configuring Alerts and Notifications .....	28
1. Email Notifications .....	28
2. Sendmail .....	28
3. SSMTP .....	28
4. SMS Alerts .....	29
5. Pagers, Popups and Other Messaging .....	29
12. Optimizing and Enhancing Nagios .....	30
1. Organizing Your Config Files .....	30
2. Configuring Layout and Dependencies .....	31
3. Improving Logical Layout .....	31
13. Customizing Nagios .....	32
1. Replacing the Default Icons .....	32
2. Custom Headers and Footers .....	34
3. Editing the Navigation Bar .....	35
14. Add-Ons and Related Projects .....	36
1. The A-Z of Nagios Related Projects .....	36
15. Troubleshooting .....	37

1. Checking Logs .....	37
2. Checking the Nagios Process .....	37
3. Permissions .....	37
4. Plugins .....	37
5. Plugins and Dependencies .....	37
16. More Information .....	38
1. Glossary .....	38
2. Index .....	38

---

# Preface

## What This Book Will Cover

### Warning

This is a preview pre-release only. There are several sections missing and a few that need lots of work. This document is only intended for experienced Nagios users for review and feedback.

Stuck? Try Google for that particular keyword or error message. If you still need help, I will try to help if I have time.

You can email me at [chris.burgess@nagiosbook.org](mailto:chris.burgess@nagiosbook.org)

### Intended Audience

The intended audience is anyone wishing to set up a monitoring solution using Nagios.

### Conventions

Commands or text that must be edited will be in bold, the example below shows that you must type `ls -al`.

```
# ls -al
-rw-r--r--  1 chris  chris           298 Jul  24 17:41 search.php
-rw-r-----  1 chris  chris           635 Dec  24 18:24 header.extlib.php
-rw-r--r--  1 chris  chris        37202 Dec  10 17:45 xmlrpc.extlib.php
-rw-r--r--  1 chris  chris        37687 Nov  11 17:21 xmlrpc.php
```

### Obtaining the Latest Version

The latest version of this book can be found at:

<http://nagiosbook.org>

It is currently being edited on an almost daily basis, so make sure you're reviewing the most current copy.

### Comments and Corrections

For now, please email: [chris.burgess@nagiosbook.org](mailto:chris.burgess@nagiosbook.org)

### Acknowledgements

Thanks the helpful folks on the nagios-users mail list and to the following individuals.

Contributors:

- David Field

Technical Reviewers:

- Jeffery Fernandez

Suggestions and Corrections:

---

# Chapter 1. Introduction

## 1. Welcome to the World of Nagios

If you're reading this, the chances are that you've heard about Nagios and would like to jump right in and set up a monitoring system. If this is the case, you've come to right place. This document aims to assist users in installing, configuring, extending, troubleshooting and generally getting the most out of your Nagios system.

It must be said that the documentation that ships with Nagios is excellent, and there are many of good tutorials that already exist. This book was written to complement the existing documentation that is already out there, and even go a little further in some areas - especially for the beginners.

For the true beginners in the audience, you'll find I have created a nice set of step by step instructions, walking you through the process of building a server from scratch, installing Nagios and configuring Nagios to monitor a typical network.

Arguably the most useful source of Nagios information is the mailing list nagios-users. This list is frequented by expert Nagios developers and users alike and is *the* place to be if you're serious about Nagios. A great deal of information in this book has been obtained through the help of people on this list, I highly recommend subscribing and participating - to learn, and to help others learn. Almost every Nagios related question I've ever had has been answered and documented in the list archives. It's a good idea to check there before posting.

## 2. Useful Nagios Resources

There are several places that you can find information on Nagios. Here are some places you should bookmark.

- [Nagios.org Documentation](#)
- [Nagios.org FAQ's](#)
- [Nagios Mail Lists](#)

## 3. Other Tutorials

If the steps contained in this document don't work for you, or you're having trouble (in either case, I'd like to know!), you might find the following tutorials helpful.

<http://www.onlamp.com/pub/a/onlamp/2002/09/05/nagios.html>

[http://www.section6.net/wiki/index.php/Setting\\_up\\_Nagios\\_in\\_FreeBSD](http://www.section6.net/wiki/index.php/Setting_up_Nagios_in_FreeBSD)

---

# Chapter 2. What is Nagios?

## 1. About Nagios

The complexity of modern networks and systems is somewhat astounding, as any experienced System Administrator will tell you. Even seemingly small networks found in many Small/Medium Enterprises (SME's) can have extremely high levels of complexity in the systems they run.

Nagios was designed as a rock solid framework for monitoring, scheduling and alerting. Nagios contains some very powerful features, harnessing them is not only a matter of understanding how Nagios works, but also how the system you're monitoring also works. This is an important realization. Nagios can't automatically teach you about complex systems, but it will be an valuable tool to help you in your journey.

So what are the sorts of things Nagios can do? Nagios can do much more than this, but nevertheless here's a list of common things that Nagios is used for.

- Check to see if a server is up and running
- Notify you if a server is down (by email/pager/SMS)
- Check to see if a service is running (mail, http, pop, ssh)
- Check to see if a process (or Windows service) is running
- Gather performance statistics on a server
- Allow specific alerts to only go to particular groups/individuals
- Get reports of downtime on your servers

A more technically accurate and complete list of Nagios' features can be found in the official documentation (<http://www.nagios.org/docs/>).

The Nagios package doesn't contain any checking tools (called plugins) at all. Does that statement sound crazy? Sure, but let me explain. Nagios focuses on doing what it does best - providing a robust, reliable and extensible framework for any type of check that a user can come up with.

So how does Nagios perform it's checking? A huge number of plugins already exist that extend Nagios to perform every type of check imaginable. And if there isn't an existing check that already exists, you're free to write your own. The nagios-plugins package is separately maintained and can be downloaded from various sources. We will cover the Nagios plugins later on.



Ethan Galstad is the creator of Nagios. Karl DeBisschop, Subhendu Ghosh, Ton Voon, and Stanley Hopcroft are the main plugin developers. Many other people have contributed to the project over the years by submitting bug reports, patches, ideas, suggestions, add-ons, plugins, etc. A list of some of the contributors can be found at the Nagios website.

## 2. Other Monitoring Options

As with anything, each tool has its own set of strengths and weaknesses. Some of the applications may seem similar, but are very different and range from full blown SNMP management solutions, to simple applications with not much flexibility. Big Brother, OpenNMS, OpenView and SysMon (there are dozens more) are often compared to Nagios, however they are quite different in many respects. In my travels as an IT professional, Nagios is the most commonly used monitoring tool by far. There are lots of specialist companies that offer monitoring as a service. A large number of these use Nagios.

## 3. Why Use Nagios?

Nagios is an excellent choice if you want to perform any sort of monitoring. Nagios' main strengths are:

- Open Source
- Robust and Reliable
- Highly Configurable
- Easily Extensible
- Active Development
- Active Community
- Nagios runs on many Operating System

Nagios can be used to monitor all sorts of things, here are some common things are typically monitored:

- Ping to see if host is reachable
- Services such as DHCP, DNS, FTP, SSH, Telnet, HTTP, NTP, POP3, IMAP, SMTP etc.
- Database servers such as MySQL, Postgres, Oracle, SQL Server etc.

- Application level information (Apache, Postfix, LDAP, Citrix etc.)

## 4. Nagios Demo

The Nagios site maintains numerous screenshots that gives you a feel for the interface, however if you'd like to see Nagios in action, I've recorded a session where I shutdown an Apache service to simulate a problem. Nagios detects this problem and sends an email alert accordingly. It's around 5 minutes long and will definitely give you a feel for Nagios.

Watch the demo now! (coming soon)

---

# Chapter 3. How Nagios Works

## 1. A Typical Scenario

It's very useful to conceptually understand how Nagios works. The following is a very simplified view on how Nagios works.

Nagios runs on a server, usually as a daemon (or service). Nagios periodically run plugins residing (usually) on the same server, they contact (PING etc.) hosts and servers on your network or on the Internet. You can also have information sent to Nagios. You then view the status information using the web interface. You can also receive email or SMS notifications if something happens. Event Handlers can also be configured to "act" if something happens.

IMAGE

---

# Chapter 4. Preparing for Installation

## 1. Getting Ready

For the cost of hardware these days, I usually recommend running a dedicated server.

One complaint I often hear from users (usually new to Linux/Unix or Open Source), is that Nagios is difficult to configure. The Nagios Book aims to dispel this myth, showing you how to build a Nagios server from scratch monitoring several hosts and sending alerts in a matter of hours. Note that it's not a competition when it comes to setting up any server, however I've heard reports of people complaining of taking days and even weeks to set up a Nagios server. This is simply not true, as you will find out!

Please note that this document does not explain the how's and why's of how to build a server, that is beyond the scope of this material. My goal is to try and get users up and running quickly so they can experience the power of Nagios. Having said that, I have created some basic steps to build a FreeBSD server from scratch.

These instructions were written and tested using FreeBSD 6.0, however they should also work just as well on earlier versions. If you're using Linux, some steps may be slightly different. I hope to add steps for each popular distro very soon.

I plan to add some other distro specific information here eventually. If you have installation notes for another platform, I'll gladly add it.

## 2. Selecting an Operating System

## 3. Setting Up The Server - FreeBSD

### Tip

The FreeBSD project already have a very thorough and complete set of documentation covering how to install FreeBSD. I know the following section is rough, it's only included since it was requested. I highly recommend reading the fine documentation at <http://www.freebsd.org/docs.html>.

Burn a copy of the boot CD (<http://www.freebsd.org>).

Reboot with the CD in your drive, make sure your BIOS is set to boot off of CDROM first.

Sysinstall (the FreeBSD System Installer) will run.

Select "Standard"

FDISK Partition Editor - Slices

d - Delete any existing "slices" (any data will be destroyed).

c - Create a new slice, the defaults will be fine.

s - Set that slice to be bootable.

q - Save the changes and quit.

Boot Manager

Select standard, as long as you don't have any other OS' on that disk. This means the disk will boot straight into FreeBSD.

Disk Label Editor

c - Create partitions

a - Use auto defaults (you are free to enter your own values if you wish)

q - Save the changes and quit.

Choose Distribution

Select Minimal.

Install Method?

FTP

Choosing a local mirror. Since you'll be installing via FTP, a quick connection means everything.

Choose Network Card. The default should be ok.

IP 6 = No

Perform as Gateway = No

Enter your network settings here, you must know your gateway (probably your router) and your DNS server (probably provided by your ISP).

SSH = Yes

Inetd = No

DHCP = No

FTP = No

NFS Server = No

NFS Client = No

System Console = No

Time Zone = Yes

Linux Binary Compatibility = No

Mouse = No

Packages = No

Create User = Yes

Exit Install

Remove CDROM and reboot.

Then you MUST install the port tree.

Login as root, then run sysinstall.

Configure > Distributions > Ports (and perhaps the Man pages)

Go grab a coffee!

Make sure you update the locate database (this is mentioned in the next chapter) once the ports are installed, this will make your life much easier.

You now have a fully functioning FreeBSD server with the latest ports ready to be easily installed. I highly recommend that you make sure you keep your FreeBSD system updated.

Upgrading the System

(coming soon)

Portupgrade

(coming soon)

[I'm still working on the most efficient and easiest way to display this]

## 4. Useful FreeBSD Commands

If you've come from a Windows background, here are a few commands that will help you find your way around.

Finding Files Using FreeBSD

Make sure the locate database is updated after you install packages. This will let you find files that have been recently installed.

Update the locate database:

```
#/usr/libexec/locate.updatedb
```

Now you can find files by running the locate command, an example is listed below.

```
# locate check_oracle
/usr/local/libexec/nagios/check_oracle
/usr/ports/net-mgmt/nagios-plugins/work/nagios-plugins-1.4/contrib/check_oracle_instance.pl
/usr/ports/net-mgmt/nagios-plugins/work/nagios-plugins-1.4/contrib/check_oracle_tbs
/usr/ports/net-mgmt/nagios-plugins/work/nagios-plugins-1.4/plugins-scripts/check_oracle
/usr/ports/net-mgmt/nagios-plugins/work/nagios-plugins-1.4/plugins-scripts/check_oracle.sh
```

You probably won't have check\_oracle on your system yet, this is just to illustrate how the command works.

## 5. Installing Apache

You can install Apache 2 or 1.3, both work equally well. I have chosen to install Apache 2.

```
cd /usr/ports/www/apache2
make
make install
make clean
```

Edit ServerName, use an IP if not using a real (DNS listed, or FQDN) domain name.

```
ee /usr/local/etc/apache2/httpd.conf
```

*Some users might have to comment out the following line referencing "mod\_unique\_id" if Apache won't load (check your logs). More info coming!*

```
monitor# /usr/local/sbin/apachectl start
/usr/local/sbin/apachectl start: httpd started

monitor# ps -aux | grep httpd
root  8368  0.0  0.2  2500  2100  ??  Ss   12:37PM  0:00.02 /usr/local/sbin/httpd
www   8369  0.0  0.2  2884  2320  ??  I   12:37PM  0:00.00 /usr/local/sbin/httpd
www   8370  0.0  0.2  2512  2124  ??  I   12:37PM  0:00.00 /usr/local/sbin/httpd
```

```
www 8371 0.0 0.2 2512 2124 ?? I 12:37PM 0:00.00 /usr/local/sbin/httpd
www 8372 0.0 0.2 2512 2124 ?? I 12:37PM 0:00.00 /usr/local/sbin/httpd
www 8373 0.0 0.2 2512 2124 ?? I 12:37PM 0:00.00 /usr/local/sbin/httpd
www 8374 0.0 0.2 2512 2124 ?? I 12:37PM 0:00.00 /usr/local/sbin/httpd
```

Good, it's running. Now you must add Apache to rc.conf so it's starts at boot time.

```
ee /etc/rc.conf
```

Then add:

```
apache2_enable="YES"
```

Visit the IP address of your server (i.e. <http://192.168.1.10> or whatever IP you used) , you should be greeted with the default Apache welcome page as shown below.



---

# Chapter 5. Installing Nagios

## 1. Obtaining Nagios

OK, so you know all about what Nagios can do, it's time to get on with it and choose an installation method.

You have two main choices when installing Nagios:

- Tarball
- Package

For most users, your OS' native package management tool will be the easiest method of installing Nagios. You will find Nagios in all common package repositories such as RPM, apt-get, Ports etc. Note that some package collections will have several versions of Nagios, you may have to specify which version you want. Most common package collections will be fairly current.

Always consult your package documentation, this will usually contain information specific to your OS (both flavour and version!). No tutorial or guide could possibly cover every version, installation and configuration.

You can download the latest tarball and the RPM's from:

<http://www.nagios.org/download>

But wait, if you're using FreeBSD, you don't have to. You can use the Ports Collection.

### Tip

The FreeBSD Ports Collection provides an easy way of installing and updating software. Visit [www.freebsd.org/docs](http://www.freebsd.org/docs) for more information on using the Ports system.

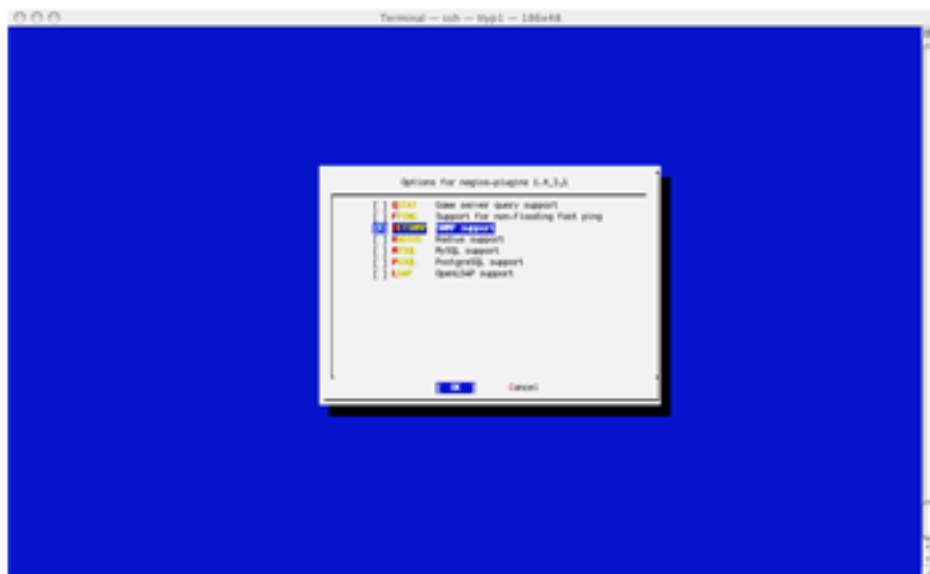
If you're using another Operating System and you've installed Nagios, you may also need to install the Nagios-Plugins package. Using FreeBSD you can install Nagios and the Nagios Plugins package from the ports collection by running the following commands. *With previous FreeBSD ports of Nagios, you had to install the plugins separately, now it's all installed at once.*

```
# cd /usr/ports/net-mgmt/nagios/  
# make  
# make install  
# make clean
```

Note, this will install Nagios version 2 (at the time of writing, it's still in Beta).

I recommend selecting SNMP when prompted as shown below.





If you are using a package management tool, it should be fairly straightforward, but please read the package notes, they are there for a reason. If you're installing from source, read the documentation. It's very thorough.

## 2. Knowing Where Nagios Lives

One thing I highly recommend (especially for new BSD users) is to update your locate database as described earlier, and run `locate nagios`. This will show you the exact paths where Nagios is installed. You will need this information later. The following commands will save this information into text files for your reference; perhaps you might find it useful to print them.

List all file locations:

```
locate nagios | grep -v /usr/ports | grep -v /var/db > nagios_file_locations_FULL.txt
```

List a trimmed down version of locations:

```
locate nagios | grep -v /usr/ports | grep -v /var/db | grep -v /usr/local/share/nagios/ | grep -v /usr/lo
```

Here is the output of the trimmed version:

```
/usr/local/bin/nagios
/usr/local/bin/nagiostats
/usr/local/etc/nagios
/usr/local/etc/nagios/cgi.cfg
/usr/local/etc/nagios/checkcommands.cfg
/usr/local/etc/nagios/minimal.cfg
/usr/local/etc/nagios/misccommands.cfg
/usr/local/etc/nagios/nagios.cfg
/usr/local/etc/nagios/resource.cfg
/usr/local/etc/nagios/samples
/usr/local/etc/nagios/samples/bigger.cfg-sample
/usr/local/etc/nagios/samples/cgi.cfg-sample
/usr/local/etc/nagios/samples/checkcommands.cfg-sample
/usr/local/etc/nagios/samples/minimal.cfg-sample
/usr/local/etc/nagios/samples/misccommands.cfg-sample
/usr/local/etc/nagios/samples/nagios.cfg-sample
/usr/local/etc/nagios/samples/resource.cfg-sample
/usr/local/etc/rc.d/nagios.sh
```

```
/usr/local/libexec/nagios
/usr/local/share/nagios
/var/mail/nagios
/var/spool/nagios
/var/spool/nagios/archives
/var/spool/nagios/comments.dat
/var/spool/nagios/downtime.dat
/var/spool/nagios/nagios.lock
/var/spool/nagios/nagios.log
/var/spool/nagios/objects.cache
/var/spool/nagios/retention.dat
/var/spool/nagios/rw
/var/spool/nagios/status.dat
/var/spool/nagios/status.sav
```

### 3. Telling Apache about Nagios

Note: If you installed Nagios by a package manager, or it is preinstalled by your distro you may not have to do this step.

Add the following to httpd.conf (your path might be different depending on the OS and version of Apache, remember to use locate if you get stuck):

```
# ee /usr/local/etc/apache2/httpd.conf
```

Then copy and paste the text below:

```
ScriptAlias /nagios/cgi-bin /usr/local/share/nagios/cgi-bin
<Directory "/usr/local/share/nagios/cgi-bin">
    AllowOverride AuthConfig
    Options ExecCGI
    Order allow,deny
    Allow from all
</Directory>

Alias /nagios /usr/local/share/nagios
<Directory "/usr/local/share/nagios">
    Options None
    AllowOverride AuthConfig
    Order allow,deny
    Allow from all
</Directory>
```

Please note the paths for FreeBSD are slightly different from the Nagios docs (it's a Linux/BSD thing).

```
# /usr/local/sbin/apachectl restart
```

You should now be able to visit the IP/nagios (eg. <http://192.168.0.10/nagios>) of your server and you should see the Nagios navigation. You will (and should) note that Nagios isn't actually running yet. We still have to configure Nagios.

One the plus side, you should now be able to access the Nagios documentation locally on your Nagios server.

---

# Chapter 6. Configuring Nagios

## 1. Editing Config Files Manually

The standard way most people configure Nagios is by simply copying the samples included and modify them with a text editor as needed. Setting up Nagios to monitor even a modest sized network is where most of the work lies. Even though it does take some work, the Nagios configuration files can be "streamlined" using includes and object definitions, this is covered in a separate topic.

I will be covering how to configure Nagios using a text editor. Since we'll be using FreeBSD, my editor of choice is ee. You can use vi, joe, pico or even Notepad on Windows (via FTP or SCP) if you wish.

## 2. Configuration Front Ends

### Configuration Front Ends

I'm currently trying several third party tools that exist, below are my initial findings. Keep in mind some of them are early on in development and are a little rough. Some of these tools require extra software on your server and take some work themselves to configure. My instructions don't cover using these tools, but I hope to expand this chapter once I have time to experiment.

#### Nagmin

#### Monarch

Monarch is a very slick looking interface for Nagios. I haven't installed it yet, I hope to soon. One appealing factor in choosing this tool would be the excellent documentation.

#### NagiosWeb

The tool NagiosWeb looks very promising. It uses a MySQL database to store your host/service information, and then writes that to the config files. NagiosWeb adds a few extra menu items to the main Nagios navigation, so access is very handy. I ran into a few problems with the code, so I have abandoned this tool for now. I might come back to it at a later date.

#### NagioSQL

NagioSQL is another web configuration interface for Nagios. I haven't installed this yet, however it seems to be popular among users.

#### Simple Config

This is a little PHP application that builds Nagios configurations for you to copy and paste into your config files. Perhaps not as sophisticated as the other options, but it's simple and it works!

If there are any other tools that you come across not listed here, let me know and I'll try to get around to trying them.

## 3. Getting Started

Some people freak out at this part, but this is where the fun starts. It's where you get to configure the actual server and hosts (among other things).

## Tip

If you're really against editing the files manually (or have a very large network), there is a Perl script called `nmap2nagios.pl` that can be downloaded from SourceForge. If you're installing `nmap2nagios.pl`, you may have to force install the XML::Simple Perl module (well I have to on FreeBSD 5.4, 6.0 and OS X) which is a dependency.

```
monitor# cd /usr/local/etc/nagios/
monitor# ls -al
total 104
drwxr-xr-x  2 root  wheel   512 Nov  7 17:56 .
drwxr-xr-x  6 root  wheel   512 Nov  8 12:17 ..
-rw-r--r--  1 root  wheel  30046 Nov  7 17:56 bigger.cfg-sample
-rw-r--r--  1 root  wheel   9569 Nov  7 17:56 cgi.cfg-sample
-rw-r--r--  1 root  wheel   4475 Nov  7 17:56 checkcommands.cfg-sample
-rw-r--r--  1 root  wheel  13602 Nov  7 17:56 minimal.cfg-sample
-rw-r--r--  1 root  wheel   4297 Nov  7 17:56 misccommands.cfg-sample
-rw-r--r--  1 root  wheel  30735 Nov  7 17:56 nagios.cfg-sample
-rw-r--r--  1 root  wheel   1335 Nov  7 17:56 resource.cfg-sample
```

You will notice that all the config files are there ready to use, all you have to do is rename them to `.cfg` rather than `sample`. The structure and use of the config files is covered thoroughly in the Nagios documentation. To keep things simple, we'll be using a basic configuration using the `minimal.cfg`.

Manually backup files into a separate directory, to reduce clutter and rename the samples to make them the live config files.

```
monitor# mkdir samples
monitor# cp * /samples/*
monitor# ls
bigger.cfg-sample          checkcommands.cfg-sample  misccommands.cfg-sample  resource
cgi.cfg-sample            minimal.cfg-sample        nagios.cfg-sample        samples
monitor# mv bigger.cfg-sample bigger.cfg
monitor# mv cgi.cfg-sample cgi.cfg
monitor# mv checkcommands.cfg-sample checkcommands.cfg
monitor# mv minimal.cfg-sample minimal.cfg
monitor# mv misccommands.cfg-sample misccommands.cfg
monitor# mv nagios.cfg-sample nagios.cfg
monitor# mv resource.cfg-sample resource.cfg
monitor# ls -al
total 76
drwxr-xr-x  3 root  wheel   512 Nov  9 16:24 .
drwxr-xr-x  6 root  wheel   512 Nov  8 12:17 ..
-rw-r--r--  1 root  wheel  30046 Nov  7 17:56 bigger.cfg
-rw-r--r--  1 root  wheel   9569 Nov  7 17:56 cgi.cfg
-rw-r--r--  1 root  wheel   4475 Nov  7 17:56 checkcommands.cfg
-rw-r--r--  1 root  wheel  13602 Nov  7 17:56 minimal.cfg
-rw-r--r--  1 root  wheel   4297 Nov  7 17:56 misccommands.cfg
-rw-r--r--  1 root  wheel  30735 Nov  7 17:56 nagios.cfg
-rw-r--r--  1 root  wheel   1335 Nov  7 17:56 resource.cfg
drwxr-xr-x  2 root  wheel   512 Nov  9 16:22 samples
```

Now we have all the files in place, we're almost there. All we have to do is edit a few files to configure our system and to tell Nagios about what we want to monitor. There are only 3 files you need to edit to get a minimal system up and running.

## Tip

If you're using FreeBSD, all of the Nagios config files are located in: `/usr/local/etc/nagios`

### nagios.cfg

Open `nagios.cfg` and make the following changes:

Comment out the following lines.

```
#cfg_file=/usr/local/etc/nagios/checkcommands.cfg
#cfg_file=/usr/local/etc/nagios/misccommands.cfg
```

### minimal.cfg

Open minimal.cfg and make the following changes:

```
#####
# MINIMAL.CFG
#
# MINIMALISTIC OBJECT CONFIG FILE (Template-Based Object File Format)
#
# Last Modified: 03-23-2005
#
#
# NOTE: This config file is intended to be used to test a Nagios installation
# that has been compiled with support for the template-based object
# configuration files.
#
# This config file is intended to servce as an *extremely* simple
# example of how you can create your object configuration file(s).
# If you're interested in more complex object configuration files for
# Nagios, look in the sample-config/template-object/ subdirectory of
# the distribution.
#
#####

#####
#####
#
# TIME PERIODS
#
#####
#####

# This defines a timeperiod where all times are valid for checks,
# notifications, etc. The classic "24x7" support nightmare. :-)

define timeperiod{
    timeperiod_name 24x7
    alias            24 Hours A Day, 7 Days A Week
    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
}

#####
#####
#
# COMMANDS
#
#####
#####

# This is a sample service notification command that can be used to send email
# notifications (about service alerts) to contacts.

define command{
    command_name notify-by-email
    command_line /usr/bin/printf "%b" "***** Nagios @VERSION@ *****\n\nNotification Type: $NOTIFICATI
}

# This is a sample host notification command that can be used to send email
# notifications (about host alerts) to contacts.

define command{
    command_name host-notify-by-email
    command_line /usr/bin/printf "%b" "***** Nagios @VERSION@ *****\n\nNotification Type: $NOTIFICATI
}
```

```

# Command to check to see if a host is "alive" (up) by pinging it
define command{
    command_name    check-host-alive
    command_line    $USER1$/check_ping -H $HOSTADDRESS$ -w 99,99% -c 100,100% -p 1
}

# Generic command to check a device by pinging it
define command{
    command_name    check_ping
    command_line    $USER1$/check_ping -H $HOSTADDRESS$ -w $ARG1$ -c $ARG2$ -p 5
}

# Command used to check disk space usage on local partitions
define command{
    command_name    check_local_disk
    command_line    $USER1$/check_disk -w $ARG1$ -c $ARG2$ -p $ARG3$
}

# Command used to check the number of currently logged in users on the
# local machine
define command{
    command_name    check_local_users
    command_line    $USER1$/check_users -w $ARG1$ -c $ARG2$
}

# Command to check the number of running processing on the local machine
define command{
    command_name    check_local_procs
    command_line    $USER1$/check_procs -w $ARG1$ -c $ARG2$
}

# Command to check the load on the local machine
define command{
    command_name    check_local_load
    command_line    $USER1$/check_load -w $ARG1$ -c $ARG2$
}

#####
#####
#
# CONTACTS
#
#####
#####

# In this simple config file, a single contact will receive all alerts.
# This assumes that you have an account (or email alias) called
# "@nagios_user-admin" on the local host.

define contact{
    contact_name    chrisb
    alias           Chris Burgess
    service_notification_period    24x7
    host_notification_period      24x7
    service_notification_options  w,u,c,r
    host_notification_options     d,r
    service_notification_commands  notify-by-email
    host_notification_commands    host-notify-by-email
    email           chris.burgess@nagiosbook.org
}

#####
#####
#
# CONTACT GROUPS

```

```

#####
#####

# We only have one contact in this simple configuration file, so there is
# no need to create more than one contact group.

define contactgroup{
    contactgroup_name    admins
    alias                Nagios Administrators
    members              chrisb
}

#####
#####
#
# HOSTS
#
#####
#####

# Generic host definition template - This is NOT a real host, just a template!

define host{
    name                generic-host    ; The name of this host template
    notifications_enabled    1        ; Host notifications are enabled
    event_handler_enabled    1        ; Host event handler is enabled
    flap_detection_enabled    1        ; Flap detection is enabled
    failure_prediction_enabled    1    ; Failure prediction is enabled
    process_perf_data        1        ; Process performance data
    retain_status_information    1     ; Retain status information across program restarts
    retain_nonstatus_information    1  ; Retain non-status information across program restarts
    register              0        ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL HOST, JU
}

# Since this is a simple configuration file, we only monitor one host - the
# local host (this machine).

define host{
    use                generic-host    ; Name of host template to use
    host_name          localhost
    alias              localhost
    address            127.0.0.1
    check_command      check-host-alive
    max_check_attempts    10
    notification_interval    120
    notification_period    24x7
    notification_options    d,r
    contact_groups     admins
}

define host{
    use                generic-host    ; Name of host template to use
    host_name          intranet
    alias              Intranet Server
    address            192.168.1.99
    check_command      check-host-alive
    max_check_attempts    10
    notification_interval    120
    notification_period    24x7
    notification_options    d,r
    contact_groups     admins
}

#####
#####
#
# HOST GROUPS
#
#####
#####

# We only have one host in our simple config file, so there is no need to
# create more than one hostgroup.

define hostgroup{
    hostgroup_name    test
    alias            Test Servers
}

```

```

members      localhost,intranet
}

#####
#####
#
# SERVICES
#
#####
#####

# Generic service definition template - This is NOT a real service, just a template!

define service{
    name                generic-service ; The 'name' of this service template
    active_checks_enabled 1           ; Active service checks are enabled
    passive_checks_enabled 1          ; Passive service checks are enabled/accepted
    parallelize_check     1           ; Active service checks should be parallelized (disabling
    obsess_over_service   1           ; We should obsess over this service (if necessary)
    check_freshness       0           ; Default is to NOT check service 'freshness'
    notifications_enabled 1           ; Service notifications are enabled
    event_handler_enabled 1           ; Service event handler is enabled
    flap_detection_enabled 1          ; Flap detection is enabled
    failure_prediction_enabled 1      ; Failure prediction is enabled
    process_perf_data     1           ; Process performance data
    retain_status_information 1       ; Retain status information across program restarts
    retain_nonstatus_information 1    ; Retain non-status information across program restarts
    register              0           ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL SERVICE,
}

# Define a service to "ping" the local machine

define service{
    use                generic-service ; Name of service template to use
    host_name          localhost, intranet
    service_description PING
    is_volatile        0
    check_period       24x7
    max_check_attempts 4
    normal_check_interval 5
    retry_check_interval 1
    contact_groups     admins
    notification_interval 960
    notification_period 24x7
    check_command       check_ping!100.0,20%!500.0,60%
}

# Define a service to check the disk space of the root partition
# on the local machine. Warning if < 20% free, critical if
# < 10% free space on partition.

define service{
    use                generic-service ; Name of service template to use
    host_name          localhost
    service_description Root Partition
    is_volatile        0
    check_period       24x7
    max_check_attempts 4
    normal_check_interval 5
    retry_check_interval 1
    contact_groups     admins
    notification_interval 960
    notification_period 24x7
    check_command       check_local_disk!20%!10%!/
}

# Define a service to check the number of currently logged in
# users on the local machine. Warning if > 20 users, critical
# if > 50 users.

define service{
    use                generic-service ; Name of service template to use
    host_name          localhost
    service_description Current Users
    is_volatile        0
    check_period       24x7
    max_check_attempts 4

```



```

    normal_check_interval      5
    retry_check_interval      1
    contact_groups             admins
    notification_interval      960
    notification_period        24x7
    check_command               check_local_users!20!50
}

# Define a service to check the number of currently running procs
# on the local machine. Warning if > 250 processes, critical if
# > 400 users.

define service{
    use                         generic-service           ; Name of service template to use
    host_name                   localhost
    service_description         Total Processes
    is_volatile                  0
    check_period                 24x7
    max_check_attempts           4
    normal_check_interval        5
    retry_check_interval         1
    contact_groups               admins
    notification_interval        960
    notification_period          24x7
    check_command                check_local_procs!250!400
}

# Define a service to check the load on the local machine.

define service{
    use                         generic-service           ; Name of service template to use
    host_name                   localhost
    service_description         Current Load
    is_volatile                  0
    check_period                 24x7
    max_check_attempts           4
    normal_check_interval        5
    retry_check_interval         1
    contact_groups               admins
    notification_interval        960
    notification_period          24x7
    check_command                check_local_load!5.0,4.0,3.0!10.0,6.0,4.0
}

# EOF

```

### cgi.cfg (optional)

Turn off Authentication for the CGI's when getting started. After you can view the web interface and everything is working, then turn Authentication on. This section is totally optional, however it does eliminate authentication problems (which are common for some beginners) during the configuration stage.

Open cgi.cfg and make the following changes:

```
use_authentication=1
```

change to

```
use_authentication=0
```

### Warning

It is a very bad idea to permanently disable authentication. The documentation covers the topic in an incredibly thorough fashion. It is however, worth knowing that this option exists when you're initially configuring Nagios.

### resources.cfg

```
$USER1$=/usr/local/libexec/nagios
```

You should now be able to run a pre-flight check with success. You should get something like the following.

```
monitor# /usr/local/bin/nagios -v /usr/local/etc/nagios/nagios.cfg
Nagios 2.0b3
Copyright (c) 1999-2005 Ethan Galstad (www.nagios.org)
Last Modified: 04-03-2005
License: GPL

Reading configuration data...

Running pre-flight check on configuration data...

Checking services...
  Checked 6 services.
Checking hosts...
  Checked 2 hosts.
Checking host groups...
  Checked 1 host groups.
Checking service groups...
  Checked 0 service groups.
Checking contacts...
  Checked 1 contacts.
Checking contact groups...
  Checked 1 contact groups.
Checking service escalations...
  Checked 0 service escalations.
Checking service dependencies...
  Checked 0 service dependencies.
Checking host escalations...
  Checked 0 host escalations.
Checking host dependencies...
  Checked 0 host dependencies.
Checking commands...
  Checked 8 commands.
Checking time periods...
  Checked 1 time periods.
Checking extended host info definitions...
  Checked 0 extended host info definitions.
Checking extended service info definitions...
  Checked 0 extended service info definitions.
Checking for circular paths between hosts...
Checking for circular host and service dependencies...
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
```

Now, let's start nagios and see what happens!

```
monitor# /usr/local/bin/nagios /usr/local/etc/nagios/nagios.cfg &
[1] 13274
monitor#
Nagios 2.0b3
Copyright (c) 1999-2005 Ethan Galstad (www.nagios.org)
Last Modified: 04-03-2005
License: GPL

Nagios 2.0b3 starting... (PID=13274)
```

Let's run ps and grep to make sure it's running.

```
monitor# ps -aux | grep nagios
nagios 13274  0.0  0.2 3348 2456 p0 S   4:54PM  0:00.05 /usr/local/bin/nagios /usr/local/etc/nagio
```

Tail the nagios log to check for errors.

```
monitor# tail -f /var/spool/nagios/nagios.log
[1131555251] Nagios 2.0b3 starting... (PID=13274)
[1131555251] LOG VERSION: 2.0
[1131555261] HOST ALERT: intranet;UP;HARD;1;PING OK - Packet loss = 0%, RTA = 0.24 ms
[1131555261] HOST NOTIFICATION: chrisb;intranet;UP;host-notify-by-email;PING OK - Packet loss = 0%, RTA =
[1131555261] SERVICE ALERT: intranet;PING;OK;SOFT;1;PING OK - Packet loss = 0%, RTA = 0.25 ms
```

All looks good! Now let's visit the URL for the web interface. Fire up Firefox and go to your IP address.

Someone pointed out this isn't necessary, although it was for me. Check the permissions carefully.

chown and chgrp the /var/spool/nagios dir and files

and the plugins...

```
monitor# chmod 755 *
```

```
monitor# chgrp wheel *
```

```
monitor# chown root *
```

More on this soon...

---

# Chapter 7. Starting Nagios

## 1. Making Sure Nagios Starts Automatically

You will probably want Nagios to start automatically when your system boots. You do this by editing `/etc/rc.conf`.

Add Nagios to `rc.conf` (this means the file `/usr/local/etc/rc.d/nagios.sh` gets loaded at boot time):

```
# ee /etc/rc.conf
```

Then add the following entry:

```
nagios_enable="YES"
```

### Warning

If you're working on a remote system, be careful editing your `rc.conf` file. Errors could cause the system to fail to boot. It's happened to me more than once!

## 2. The Pre-Flight Check

(I've already covered some of these commands, I plan to go over this section and tidy it up!)

Run a pre-flight check to verify configuration:

```
# /usr/local/bin/nagios -v /usr/local/etc/nagios/nagios.cfg
```

It's also worth mentioning that if something goes wrong and you need to make changes, there is no need to restart the server, you can start Nagios manually by running the following command:

```
# /usr/local/bin/nagios /usr/local/etc/nagios/nagios.cfg &
```

Check to make sure Nagios is running:

```
monitor# ps -aux | grep nagios
```

You can kill Nagios which will flush the config as an alternative to rebooting. You must replace PID with the actual process ID obtained from `ps -aux`.

```
monitor# kill -HUP PID
```

Running Nagios without any switches displays the following output.

```
monitor# nagios
Nagios 2.0b3
Copyright (c) 1999-2005 Ethan Galstad (www.nagios.org)
Last Modified: 04-03-2005
License: GPL

Usage: nagios [option] <main_config_file>

Options:
```

- v Reads all data in the configuration files and performs a basic verification/sanity check. Always make sure you verify your config data before (re)starting Nagios.
- s Shows projected/recommended check scheduling information based on the current data in the configuration files.
- d Starts Nagios in daemon mode (instead of as a foreground process). This is the recommended way of starting Nagios for normal operation.

Visit the Nagios website at <http://www.nagios.org> for bug fixes, new releases, online documentation, FAQs, information on subscribing to the mailing lists, and commercial and contract support for Nagios.

### 3. The -s Switch

```
monitor# /usr/local/bin/nagios -s /usr/local/etc/nagios/nagios.cfg
```

```
Nagios 2.0b3
Copyright (c) 1999-2005 Ethan Galstad (www.nagios.org)
Last Modified: 04-03-2005
License: GPL
```

```
Projected scheduling information for host and service
checks is listed below. This information assumes that
you are going to start running Nagios with your current
config files.
```

#### HOST SCHEDULING INFORMATION

-----

```
Total hosts:                11
Total scheduled hosts:      0
Host inter-check delay method: SMART
Average host check interval: 0.00 sec
Host inter-check delay:    0.00 sec
Max host check spread:     30 min
First scheduled check:     N/A
Last scheduled check:      N/A
```

#### SERVICE SCHEDULING INFORMATION

-----

```
Total services:            15
Total scheduled services:   15
Service inter-check delay method: SMART
Average service check interval: 300.00 sec
Inter-check delay:         20.00 sec
Interleave factor method:  SMART
Average services per host:  1.36
Service interleave factor:  2
Max service check spread:   30 min
First scheduled check:     Mon Nov 21 11:21:52 2005
Last scheduled check:      Mon Nov 21 11:26:32 2005
```

#### CHECK PROCESSING INFORMATION

-----

```
Service check reaper interval: 10 sec
Max concurrent service checks:  Unlimited
```

#### PERFORMANCE SUGGESTIONS

-----

```
I have no suggestions - things look okay.
```

### 4. Help!

If Nagios won't start, fear not. Go to the the chapter titled *Troubleshooting*. If you're still stuck, try asking the nagios-users list. As always, copying the error message into Google usually always pays off.

---

# Chapter 8. Nagios Plugins

## 1. The Standard Plugins

Check SSH

Add the following command definition:

```
# check_ssh
define command{
command_name check_ssh
command_line $USER1$/check_ssh -H $HOSTADDRESS$
}
```

Then add the following service definition:

```
define service{
use generic-service
host_name *
service_description SSH
is_volatile 0
check_period 24x7
max_check_attempts 3
normal_check_interval 5
retry_check_interval 1
contact_groups admins
notification_interval 120
notification_period 24x7
notification_options w,u,c,r
check_command check_ssh
}
```

## 2. The Contrib Plugins

## 3. Other Plugins

## 4. Writing Your Own Plugins

---

# Chapter 9. Monitoring Remote Unix Servers

## 1. NPRE



---

# Chapter 10. Monitoring Remote Windows Servers

## 1. NSClient

There are several ways to achieve this, however I choose to use NSClient. Each method has it's own set of pros and cons and it really depends on what you want to monitor and what you are familiar with. Install NSClient on the Windows server. The NSClient documentation lists the text you have to add to your service checks. (more info coming)

You can then monitor any service, along with typical local resources such as disk, cpu and memory usage.

---

# Chapter 11. Configuring Alerts and Notifications

## 1. Email Notifications

There are several ways get your mail alerts to work. It all depends what you're more comfortable with and whether this particular server is going to be used for anything else.

Your options are (to name a few)

- \* Configure the default install of Sendmail
- \* Replace Sendmail with Nullmailer or SSMTP
- \* Install another MTA like Postfix, QMail or Exim

## 2. Sendmail

I will not cover configuring Sendmail, however the FreeBSD website has some information in the FreeBSD Handbook that might be useful.

[http://www5.ua.freebsd.org/doc/en\\_US.ISO8859-1/books/handbook/outgoing-only.html](http://www5.ua.freebsd.org/doc/en_US.ISO8859-1/books/handbook/outgoing-only.html)

## 3. SSMTP

I choose to replace Sendmail with SSMTP.

```
monitor# cd /usr/ports/mail/ssmtp
monitor# make install replace clean
monitor# cd /usr/local/etc/ssmtp
monitor# ls
revaliases.sample          ssmtp.conf.sample
monitor# cp revaliases.sample revaliases
monitor# cp ssmtp.conf.sample ssmtp.conf
monitor# ls
revaliases                 revaliases.sample      ssmtp.conf             ssmtp.conf.sample
```

Now edit the SSMTP configuration file.

```
monitor# ee /usr/local/etc/ssmtp/ssmtp.conf
```

Below is my sample configuration, the config file is commented if you need another example.

```
mailhub=192.168.0.2 (this will be a mail server you can send mail through, can be IP or FQDN)
rewriteDomain=nagiosbook.org
hostname=monitor
```

Make sure you disable Sendmail by editing /etc/rc.conf and adding the following directives. (This is for FreeBSD version 5 or later)

```
# ee /etc/rc.conf
```

```
sendmail_enable="NO"
sendmail_submit_enable="NO"
sendmail_outbound_enable="NO"
sendmail_msp_queue_enable="NO"
```

Test to make sure you can send mail by:

```
mail -s "Test" root
<ctrl-d>
```

You should now be able to send email from your Nagios server.

## 4. SMS Alerts

Like email alerts, you have several options to getting this working.

Your options are:

Enabling an SMS service on your Nagios server

Using your own SMS gateway on another server

Using a commercial third party SMS gateway

There are several tools you can use to enable a server to send SMS's. Each tool has its own strengths and weaknesses. The following list should help you decide what to use.

\* Gnokii - Gnokii is very popular and can be used as a full-featured 2-way SMS gateway.

\* SMSTools - An alternative to Gnokii, also popular and heavily documented.

\* SMSClient - Calls a provider by modem and sends SMS's only.

If you'd like to build your own SMS gateway using Gnokii, I wrote a "SMS Gateway How To" a few years ago. You can read the original article here (<http://www.chrisburgess.com.au/sms-gateway-how-to/>), or read on for the simplified steps. I am assuming you are using the same server to run both Nagios and Gnokii.

Installing and Configuring Gnokii for SMS Alerts

```
cd /usr/ports/comms/gnokii
```

```
make install clean
```

MORE STEPS FOR GNOKII COMING SOON

## 5. Pagers, Popups and Other Messaging

If you'd prefer another messaging type, the Nagios mail list archives mention all sorts of methods people use.

---

# Chapter 12. Optimizing and Enhancing Nagios

## 1. Organizing Your Config Files

So far, we've modified the sample file called `minimal.cfg` (originally `minimal.sample`). Using the minimal sample is ideal for getting started, and by all means, you're free to use run Nagios in that configuration if you wish. At this stage however, it's worth pointing out that you can split your configurations into separate config files to help make things more manageable.

Straight from the documentation, we can see how this works:

```
# You can also tell Nagios to process all config files (with a .cfg
# extension) in a particular directory by using the cfg_dir
# directive as shown below:
#cfg_dir=/usr/local/etc/nagios/servers
#cfg_dir=/usr/local/etc/nagios/printers
#cfg_dir=/usr/local/etc/nagios/switches
#cfg_dir=/usr/local/etc/nagios/routers
```

Here are some other examples that should give you some ideas:

### Device Types

```
/usr/local/etc/nagios/routers
/usr/local/etc/nagios/switches
/usr/local/etc/nagios/printers
```

### Function Type

```
/usr/local/etc/nagios/mailservers
/usr/local/etc/nagios/winservers
/usr/local/etc/nagios/unixservers
```

### Topological Location

```
/usr/local/etc/nagios/intranet
/usr/local/etc/nagios/extranet
/usr/local/etc/nagios/border
/usr/local/etc/nagios/dmz
```

### Physical Location

/usr/local/etc/nagios/building-1  
/usr/local/etc/nagios/building-2  
/usr/local/etc/nagios/building-3  
/usr/local/etc/nagios/building-4

#### Departmental Layout

/usr/local/etc/nagios/creative  
/usr/local/etc/nagios/accounts  
/usr/local/etc/nagios/sales  
/usr/local/etc/nagios/it  
/usr/local/etc/nagios/admin  
/usr/local/etc/nagios/callcentre

#### Area of Responsibility

/usr/local/etc/nagios/support-team  
/usr/local/etc/nagios/network-team  
/usr/local/etc/nagios/dev-team

## 2. Configuring Layout and Dependencies

```
define host{
    use                generic-host           ; Name of host template to use
    host_name          adl
    alias              Primary AD Server
    address             192.168.1.27
    parents             router1
    check_command       check-host-alive
    max_check_attempts 10
    notification_interval 120
    notification_period 24x7
    notification_options d,r
    contact_groups      admins
}
```

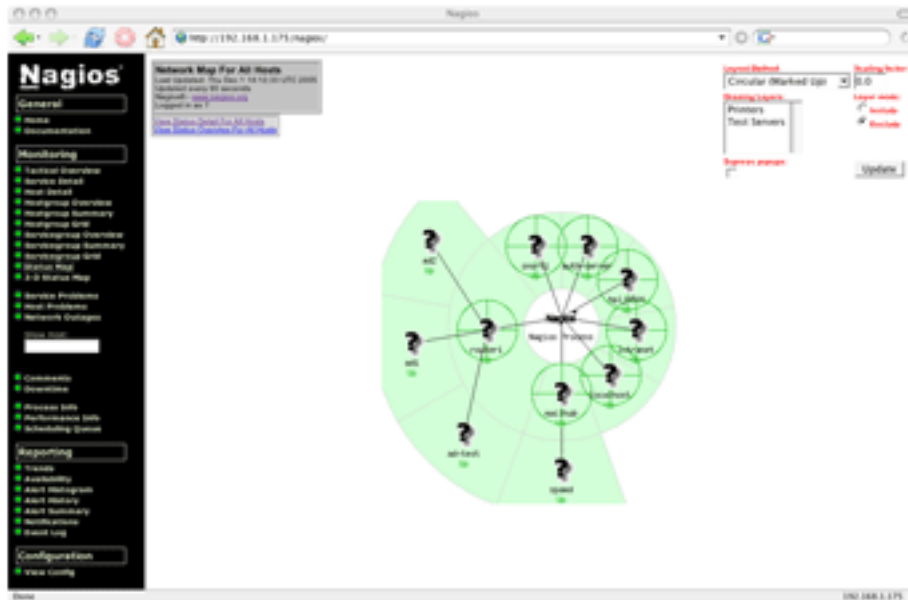
## 3. Improving Logical Layout

You can use Dummy Hosts to improve the logical layout of your status map. Search the nagios-users mail list archives for "dummy host".

# Chapter 13. Customizing Nagios

## 1. Replacing the Default Icons

By default, Nagios looks like this:



Here's how you get custom icons working. It's quite easy. In my example, I will be using the standard icons, however there are lots of "icon packs" that others have created. You could also create your own if you wish.

First, check to make sure you have the images on your system.

```
monitor# ls /usr/local/share/nagios/images/logos/
aix.gd2      caldera.gif      hp-printer40.png  mac40.png      novell140.
aix.gif      caldera.jpg      hpux.gd2           mandrake.gd2   novell140.
aix.jpg      caldera.png     hpux.gif           mandrake.gif   novell140.
aix.png      cat1900.gd2     hpux.jpg          mandrake.jpg   opensbd.g
amiga.gd2    cat2900.gd2     hpux.png          mandrake.png  opensbd.g
amiga.gif    cat5000.gd2     irix.gd2          nagios.gd2     opensbd.j
amiga.jpg    debian.gd2      irix.gif          nagios.gif     opensbd.p
amiga.png    debian.gif      irix.jpg          nagiosvrml.png redhat.gd
apple.gd2    debian.jpg      irix.png          next.gd2       redhat.gi
apple.gif    debian.png     linux40.gd2       next.gif       redhat.jp
apple.jpg    freebsd40.gd2  linux40.gif       next.jpg       redhat.pn
apple.png    freebsd40.gif  linux40.jpg       next.png      router40.
beos.gd2    freebsd40.jpg  linux40.png      ng-switch40.gd2 router40.
beos.gif    freebsd40.png  logo.gd2          ng-switch40.gif router40.
beos.jpg    hp-printer40.gd2 mac40.gd2         ng-switch40.jpg router40.
beos.png    hp-printer40.gif mac40.gif         ng-switch40.png slackware
caldera.gd2 hp-printer40.jpg mac40.jpg         novell140.gd2 slackware
```

Yes, they are there. Next, edit the file `hostextinfo.cfg`.

```
ee /usr/local/etc/nagios/nagios.cfg
```

This contains the extended host information. You also must make sure that `nagios.cfg` references `hostextinfo.cfg`, on my system I had to comment out the following line.

```
cfg_file=/usr/local/etc/nagios/hostextinfo.cfg
```

After you've done that, now edit hostextinfo.cfg.

```
ee /usr/local/etc/nagios/hostextinfo.cfg
```

I've edited my file like this.

```
define hostextinfo{
    host_name      localhost,intranet,auth-server,spamd,mailhub,router1,snort1
    notes_url      http://webserver/hostinfo.pl?host=you_can_edit_this
    icon_image     freebsd40.png
    icon_image_alt FreeBSD
    vrmI_image     freebsd40.png
    statusmap_image freebsd40.gd2
#    2d_coords     100,250
#    3d_coords     100.0,50.0,75.0
}

define hostextinfo{
    host_name      ad1,ad2,ad-test
    notes_url      http://webserver/hostinfo.pl?host=you_can_edit_this
    icon_image     win40.png
    icon_image_alt Windows
    vrmI_image     win40.png
    statusmap_image win40.gd2
#    2d_coords     100,250
#    3d_coords     100.0,50.0,75.0
}
```

Note: You can add your own coordinates for the status map.

After editing the file, save your changes, then restart Nagios.

Check to find the Process ID

```
monitor# ps -aux | grep nagios
```

```
nagios 1978 0.0 0.2 3388 2504 p0 S 6:10PM 0:00.06 /usr/local/bin/nagios /usr/local/etc/nagios/nagios.cfg
```

Send the Kill signal

```
monitor# kill 1978
```

```
monitor# Successfully shutdown... (PID=1978)
```

```
[1] Done /usr/local/bin/nagios
```

```
/usr/local/etc/nagios/nagios.cfg
```

Start Nagios

```
monitor# /usr/local/bin/nagios /usr/local/etc/nagios/nagios.cfg &
```

```
[1] 2062
```

```
monitor#
```

Nagios 2.0b3

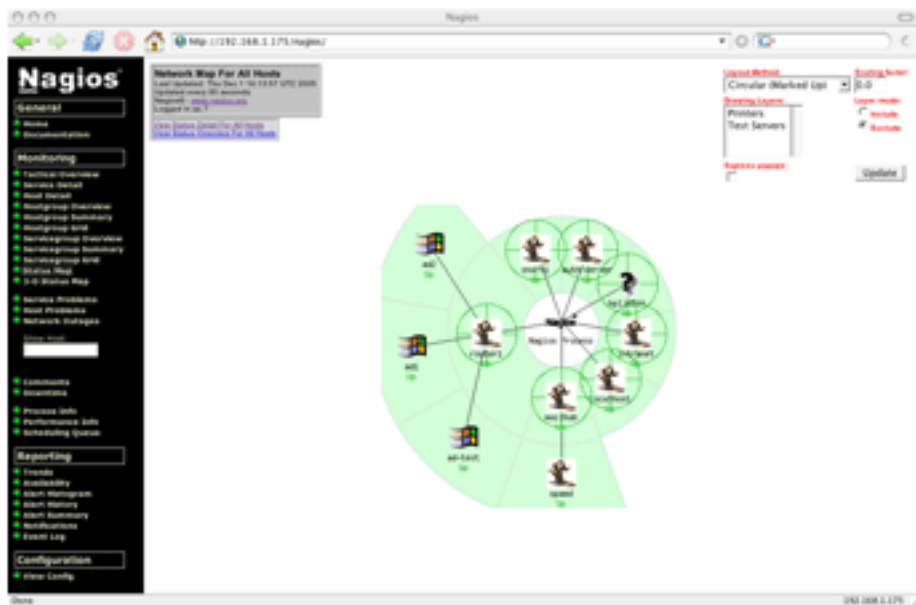
Copyright (c) 1999-2005 Ethan Galstad (www.nagios.org)

Last Modified: 04-03-2005

License: GPL

Nagios 2.0b3 starting... (PID=2062)

You should now have custom icons. My example looks like this:



## 2. Custom Headers and Footers

```
cd /usr/local/share/nagios/ssi
```

```
ee common-header.ssi
```

```
<br>
```

```
<strong>
```

```
Your Custom Header | Custom Link 1 | Custom Link 2
```

```
</strong>
```

```
<br>
```

```
<hr>
```

```
<br>
```

Save this file.

```
ee common-footer.ssi
```

```
<br>
```

```
<strong>
```

```
Your Custom Footer | Custom Link 1 | Custom Link 2
```

```
</strong>
```

```
<br>
```

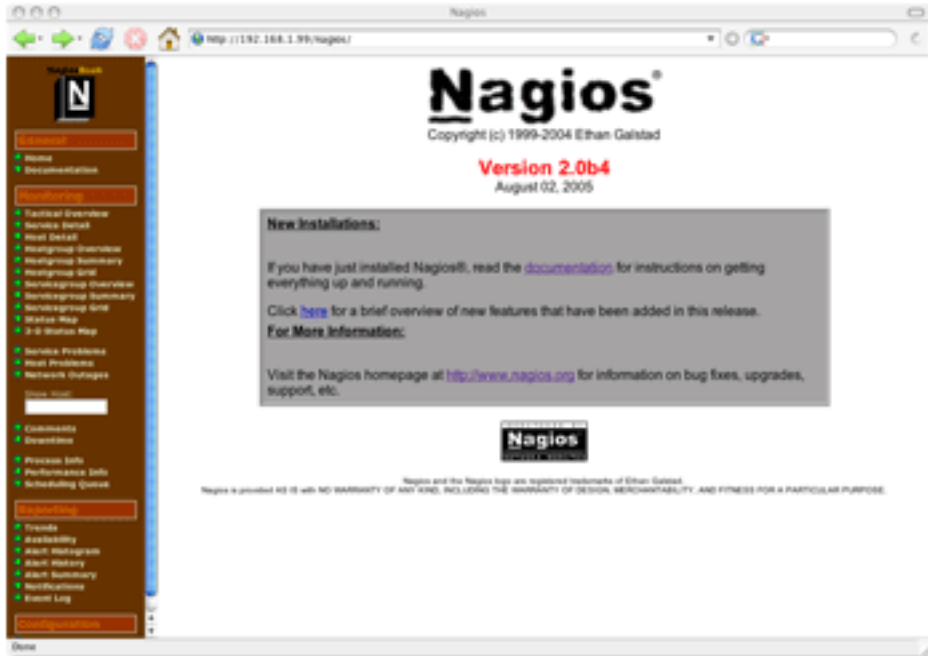
Naturally you would replace my sample code with your own HTML.



Nagios will automatically include your newly created files in it's display.

### 3. Editing the Navigation Bar

To edit the navigation, along with changing the look of your Nagios interface, all you have to do is open /usr/local/share/nagios/side.html and make your changes. Simply open the file in your HTML editor of choice, or a text editor if you choose.



The HTML below replaced the Nagios logo in this example. After making some changes to the inline CSS, I achieved the look seen above.

```
<table width="150" border="0">
  <tr>
    <td>
      <div align="center"><a href="http://www.nagiosbook.org" target="_blank"><img src="images/nagios_boo
    </td>
  </tr>
</table>
```

---

# Chapter 14. Add-Ons and Related Projects

## 1. The A-Z of Nagios Related Projects

There are literally dozens of add-on projects to enhance and extend Nagios even more. For a very updated list of what is available, I highly recommend visiting [NagiosExchange.org](http://NagiosExchange.org).

- \* Apan
- \* PerfParse
- \* Cacti Plugin
- \* Nagvis

This list is still being compiled.

The number of projects that extend and enhance Nagios is staggering. If you've read the list above and have a need for something not listed, try searching the following sites for Nagios add-ons and enhancements. New projects appear every day.

- \* [NagiosExchange.org](http://NagiosExchange.org)
- \* [SourceForge](http://SourceForge)
- \* [Google](http://Google)

---

# Chapter 15. Troubleshooting

## 1. Checking Logs

OK, so something isn't working, the first thing you do when troubleshooting any problem is to try and isolate the cause. Looking at the error logs is usually helps pinpoint the problem. Useful logs worth checking out:

```
/var/log/messages          - General system error log
/var/log/maillog           - Mail server log
/var/spool/nagios/nagios.log - The Nagios log
```

The tail command with appended with the -f switch is also very helpful since you can see what's going on in real-time. I usually have one SSH/Terminal/PuTTY session open just for tailing logs, it's very useful.

```
# tail -f /var/spool/nagios/nagios.log
```

## 2. Checking the Nagios Process

You can (and should) check the processes are running by running the ps command. It is usually helpful to filter out all of the other system processes and spear in on exactly what we're looking for.

```
# ps -aux | grep nagios
```

## 3. Permissions

You can check the file and directory permissions by running the ls -al command, please note this a very common problem seen on the lists. It is imperative that all the permissions be set correctly.

## 4. Plugins

First, make sure you can run the check directly. If you can't run the plugin from the command line (i.e. cd /location/of/your/plugins, then ./whatever\_check), then Nagios won't be able to either. If you can run it, but Nagios can't, check the ownership and permissions.

## 5. Plugins and Dependencies

Many plugins require certain libraries or applications exist on your system. If the dependencies don't exist, the plugins might not be installed. It's worth watching the console output carefully when compiling plugins. Always run ./check\_yourplugin -h for more information on that particular plugin.

---

# Chapter 16. More Information

## 1. Glossary

Not compiled yet.

## 2. Index

Not compiled yet.