

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## Table Of Contents

Introduction.....	2
Section One – Installation of the Operating System.....	3
Section Two – Installation of Host Server Programs.....	5
Section Three – Installation of the Linux Client Software (NRPE).....	14
Section Four – Installation of the Windows Client Software (NSClient).....	18
Section Five – Using the check_snmp plug-in for non-HP Printers.....	20
Section Six – Configuration of the check_flexlm plug-in.....	22
Section Seven – Configuration of the 2D and 3D Status Maps.....	25
Section Eight – Remote Host Monitoring via SSH.....	29
Section Nine – Monitoring HP-UX Workstations and Servers.....	31
Appendix A.....	34
Appendix B.....	40
Appendix C.....	43
Appendix D.....	44
Appendix E.....	62
Appendix F.....	80
Appendix G.....	81
Appendix H.....	95
Appendix I.....	99
Appendix J.....	104
Appendix K.....	106

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## Introduction:

This document was first conceived in late 2002 as a means to facilitate the installation of Nagios. At that time, Nagios, still in beta release, was being tested at the company where the author works. Since that time, several updates have occurred to the application as well as the operating system on which it resides. As of this writing, Nagios is currently available as version 1.0. The operating system on which the program runs at the author's site is now Red Hat 8.0.

The purpose of this document is similar to that of its predecessor: to provide a step-by-step procedure for new users of the application. It is not an all-encompassing paper that will answer all questions that are asked of it. Rather, it is one person's experience with the installation and various configuration issues associated with the program. What has been documented here may or may not have occurred in other instances. Continuing the spirit and tradition of its predecessor, this document is available for all to peruse, and critique. If there are errors in what is being described, please feel free to contact the author via the Nagios Users groups to notify him of the error(s).

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## Section One: Installation of the Operating System

*Note: This section is included simply to provide an example configuration. The description that follows has been successfully installed and tested as a viable platform for running Nagios. At the same time, this method of configuration is only one of a wide variety of setups that can be used, and is therefore NOT exhaustive.*

1. Obtain or download onto CD the ISO images for Red Hat 8.0 There will be a total of five CD's, but only three will be used during the installation procedure.
2. Make sure the workstation or server that will be hosting Nagios has at least 256 megabytes of RAM. Hard drive size and type, i.e.: IDE or SCSI can vary, but a good starting point is twenty gigabytes.
3. Obtain, if possible, a static ip address for the Nagios server. Additionally, add the hostname and fully qualified domain name (FQDN) with the associated address to the DNS server of the network. This will allow connection to the server via its hostname rather than having to rely on the ip address for connectivity.
4. Begin the installation of the operating system onto the workstation. Red Hat 8.0 has a utility that will scan the operating system CD's for any defects before proceeding with the loading of the software. It is entirely at the administrator's discretion whether a scan should be done.
5. Select Custom Installation when the appropriate screen appears. Using this method will allow the administrator to have complete control over what is and is not installed onto the computer and how it is configured.
6. The configuration of the mount-points for the server can be accomplished via the Disk Druid utility. One example layout could be the following:

<b>Filesystem</b>	<b>1K-blocks</b>	<b>Used</b>	<b>Available</b>	<b>Use%</b>	<b>Mounted on</b>
/dev/hda8	1004024	278896	674124	30%	/
/dev/hda1	1004024	44036	908984	5%	/boot
/dev/hda7	1004024	191486	761572	21%	/home
/dev/hda2	5036316	32828	4747656	1%	/opt
/dev/hda9	537572	17068	493196	4%	/tmp
/dev/hda5	5036316	2959336	1821116	62%	/usr
/dev/hda3	5036316	122224	4658260	3%	/var

7. The following screens include the various components that can be installed onto the system. Included here are the Gnome and KDE desktop environments as well as the individual packages. Depending on what the needs are, some or all of the files can be slated for installation. Several items that are available on the Red Hat CD's but do not need to be installed are the Apache and MySQL programs. Each of these has a later version that is freely available on the Internet. The various shells, i.e.:

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

bash, tcsh, can be installed at the administrator's discretion, although it is a very good rule of thumb to install the bash shell.

8. Configure the static ip address for the soon-to-be Nagios server. Along with the address, implement the hostname that has been assigned to the computer. If the server is to become part of an NIS domain, configure the machine in the appropriate manner.
9. After installation of the operating system is complete, install the appropriate errata packages that coextend with the base operating system.

Installation of the base system is complete. The Nagios application can be installed.

*Note: It is probably quite apparent the above procedure is, at best, an overview of the installation procedure. There were several important steps not mentioned here. The purpose here was to provide one example configuration that can be used as a reference for whatever configuration the administrator setting up Nagios chooses to use.*

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## Section Two – Installation of Host Server Programs

### 1. Obtain Nagios

- Download the Nagios core program from the appropriate website. At this writing, the URL for the website is <http://www.nagios.org>.
- Obtain the plug-ins that Nagios uses to monitor the network. *This is an important point:* The core program is strictly the engine of the program, and does not have any monitoring functionality of its own. *The plug-in's are necessary for the network monitoring that will be taking place.*
- The various nodes on the network will need a client program in order for them to be able to communicate with the Nagios application server. There are two such programs: NSClient (Windows) and NRPE (Linux). The homepage of NSClient is <http://nsclient.ready2run.nl>, while NRPE can be downloaded from the Nagios homepage. Refer to Appendix A for the documentation associated with the client software.
- Periodically, there will be updates to the core program and various add-ons to the application.
- After installation is complete, regular checks on the Nagios and its associated websites is recommended.

### 2. Prerequisite Installs

- Download and install the Apache web server from the [www.apache.org](http://www.apache.org) website. Nagios uses a web interface for the front-end of the program. Additionally, be prepared to make modifications to the httpd.conf file to accommodate Nagios. An additional program that can be downloaded is the Comanche user-interface that serves as a front-end for the Apache program. The program can be downloaded from [www.comanche.org](http://www.comanche.org). Refer to the documentation that comes with the download for proper installation.
- Download and install the open-source MySQL database program from [www.mysql.org](http://www.mysql.org). Certain aspects of Nagios use MySQL as a means to better organize data for the program. Refer to the documentation that comes with the download for proper installation.

Nagios Plugin Requirements (*taken from the REQUIREMENTS file*)

-----

Some plugins require that you have additional programs and/or libraries installed on your system before they can be used. Plugins that are dependent on other programs/libraries that are missing are usually not compiled. Requirements for various plugins are listed below:

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

check\_fping:

- Requires the fping utility distributed with SATAN. Either download and install SATAN or grab the fping program from

<http://www.stanford.edu/~schemers/docs/fping/fping.html>  
<ftp://ftp.redhat.com/pub/contrib/libc6/SRPMS/fping-2.2b1-1.src.rpm>  
<ftp://ftp.redhat.com/pub/contrib/libc6/RPMS/fping-2.2b1-1.i386.rpm>

Note that the fping command must be setuid root to function.

check\_game:

- Requires the qstat utility available from

<http://www.activesw.com/people/steve/qstat.html>

Last tested on qstat 2.3d BETA

check\_hpjd:

- Requires the UCD-SNMP package available from

<http://ucd-snmp.ucdavis.edu>

The snmpget binary is all that is required.

check\_ldap:

- Requires the LDAP libraries available from

<http://www.openldap.org>

Lib: libldap, liblber

Redhat Source: openldap-1.2.9-6.i386.rpm, openldap-devel-1.2.9-6.i386.rpm

check\_mysql:

- Requires the MySQL libraries available from

<http://www.mysql.org>

Lib: libmysql, libmysqlclient

Redhat Powertools Source:

mysql-3.20.32a-3.i386.rpm,

mysql-devel-3.20.32a-3.i386.rpm

(these seem to be broken, however, RPMS from www.mysql.org work better)

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

check\_pqsql:

- Requires the PostgreSQL libraries available from

<http://www.postgresql.org>

check\_radius:

- Requires the radiusclient library available from

<http://www.cityline.net/~lf/radius/>

RPM (rpmfind): radiusclient-0.3.1-1, radiusclient-devel-0.3.1-1

check\_snmp:

- Requires the UCD-SNMP package available from

<http://ucd-snmp.ucdavis.edu>

## 3. Install Nagios

- Unpack the distribution file that was downloaded from the Internet. There are two commands.

**gunzip nagios- version number.tar.gz tar or -xvf nagios- version number.tar**

Note: If the zip version of the program was downloaded, the syntax would be the following:

**Unzip nagios- version number.zip**

- Create the installation directory for the program. The root or sudo user account should be used when this and future steps are done to insure there are no permissions issues later during the installation. The syntax to use is:

**mkdir /usr/local/nagios**

- Create the User and Group associated with the program. Nagios has its own user account and private group. The command shown will accomplish both:

**adduser nagios**

- Run the configure script. This shell script will automatically create the necessary directory structure for the program. The generic version of the script is the following

**./configure --prefix= prefix --with-cgiurl= cgiurl --with-htmurl= htmurl --with-nagios-user= someuser --with-nagios-grp= somegroup**

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

- Replace **prefix** with the installation directory that was created previously. (the default is /usr/local/nagios)
  - Replace **cgiurl** with the actual URL that will be used for accessing the CGI's (the default is /cgi-bin/nagios) **Do not append a slash at the end of the URL.**
  - Replace **htmurl** with the actual URL to be used when accessing the html for the main interface and documentation (the default is /nagios)
  - Replace **someuser** with the name of the user on the system that will be used for setting permissions on the installed files. (the default is nagios)
  - Replace **somegroup** with the name of the group on the system that will be used for setting permissions on the installed files. (the default is nagios)
- Compile the binaries. These are the executable files needed for Nagios' operation. The command syntax is:

**make all**

- Install the Binaries and HTML Files. After the binaries are compiled (created) the next step is to actually install them in their appropriate directories. This is coupled with doing the same for the html files. The command to accomplish this task is:

**make install**

- Install the Init Script. This is the startup script used by Nagios to automatically start the program upon system boot. The utility program is created in the /etc/rc.d/init.d/nagios directory, and can be modified to reflect the correct paths for the operating system and Nagios installation. The command syntax is:

**make install-init**

Refer to Appendix B for the complete syntax of the script.

- Verify the directory structure of Nagios is correct. The location of the program, by default, is /usr/local/nagios. Executing the command `ls -l` will provide a detailed view of the infrastructure. The directory structure for Nagios should be the following:

**/usr/local/nagios**  
**/bin – Nagios core program**  
**/etc – Main, Resource, Object, and CGI configuration files**  
**/libexec – Plug-In's**  
**/sbin - CGI's**  
**/share – HTML files (for web interface and online documentation)**  
**/var – log files**



# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

- Install the plug-in's that are associated with Nagios. The procedure to install the plug-ins is similar to that of the core program. Refer to Appendix C for the documentation associated with the plug-ins.
- Set up the Web Interface for Nagios. As mentioned previously, the httpd.conf, main configuration, file for Apache will need to be modified in order for Nagios to work. It is recommended that an original file be kept intact in the event of a problem with Apache and/or Nagios. The simplest way to save an original version of the file is to copy the file with the same name and add an .orig suffix at the end. This can be accomplished by going to the appropriate directory where the conf file is located, for Apache 2 the location is /usr/local/apache2/conf, and make the change there. For example:

```
cp -pr httpd.conf httpd.conf.orig
```

- After an original copy is made an alias for the CGI's needs to be made. The default installation expects to find them accessible at **http://machinename/nagios/cgi-bin/** The following lines should be added to the httpd.conf file for this to happen:

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

- The following entries should be added to the httpd.conf file in order for the HTML files to be accessible via the web server.

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

- The entries shown above will allow for the viewing of the HTML web interface and documentation. The alias should be the same value that was entered for the --with-httmurl argument in the configure script. *The default is /nagios/.*

**Important: The Alias directive must come after the ScriptAlias directive for the CGI's. If it doesn't users will be confronted with a 404 error when attempting to access the CGI's.**

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

- After the directive entries have been entered, it is necessary for Apache to default to the Nagios CGI's as opposed to its defaults. The httpd.conf file contains a section for Apache's CGI's. Included in the file is the notation that the default setting /usr/local/apache2/cgi-bin should be changed to whatever the ScriptAliased CGI directory is if it is so configured for an application. The simplest way to insure the Nagios CGI's are referenced is to comment out the entire default directive. See below:

Excerpt from a typical httpd.conf file:

```
# "/usr/local/apache2/cgi-bin" should be changed to whatever your
# ScriptAliased CGI directory exists, if you have that configured.
#
# <Directory "/usr/local/apache2/cgi-bin">
# AllowOverride None
# Options None
# Order allow,deny
# Allow from all
# </Directory>
```

- The above excerpt shows the default cgi-bin directive commented out. Appendix D provides the entire httpd.conf file with the changes made while the subsequent Appendix E lists the file in its original format.
- After the changes have been made to the httpd.conf file, the web server will need to be rebooted for the changes to go into effect. There are two commands that can be used to do this:

```
/usr/local/apache2/bin/apachectl restart or /etc/rc.d/init.d/httpd restart
```

- Verify the changes that have been made are correct by pointing a web browser at <http://yourmachine/nagios>. If the web interface is displayed, then the configurations are correct. It is important to note the CGI's will not display information at this point.
- Configure Web Authentication by enabling access for specific users and/or groups to the CGI's. The first step is to make sure the httpd.conf file contains the AuthOverride AuthConfig statement for the Nagios CGI-BIN directory. If it does not, the following entry, shown on the next page, should be added and the web server should be subsequently rebooted.

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
<Directory /usr/local/nagios/sbin>  
AllowOverride AuthConfig  
order allow,deny  
allow from all  
Options ExecCGI  
</Directory>
```

- If authentication for access to the HTML pages in Nagios is desired, the following should also be added to the httpd.conf file.

```
<Directory /usr/local/nagios/share>  
AllowOverride AuthConfig  
order allow,deny  
allow from all  
</Directory>
```

- The second step is to create a file named .htaccess in the root of the CGI, and optionally to the HTML, directory. The defaults for Nagios is /usr/local/nagios/sbin and /usr/local/nagios/share respectively. The file(s) should have similar to the following:

```
AuthName "Nagios Access"  
AuthType Basic  
AuthUserFile  
/usr/local/nagios/etc/htpasswd.users  
require valid-user
```

Refer to Appendix F for the complete syntax of the file.

- Set up authenticated users for Nagios. Use the htpasswd command that is bundled with Apache. Running the program will create a new file called htpasswd.users in the /usr/local/nagios/etc directory. While running the program, a password will be requested to be associated with users who will be authenticating to the web server. The example below shows the syntax of the command when entering the nagiosadmin user.

```
htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

- The process can be continued to add more users as needed to the list of authenticated users. The syntax used for adding more users is similar to the one shown above except the -c option should not be used. This option actually creates the file, and using it again would overwrite the existing file with another one. Hence, the correct syntax for additional users would be:

```
htpasswd /usr/local/nagios/etc/htpasswd.users username
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

- Enable authentication and authorization functionality to the CGI files. This is accomplished by editing the cgi.cfg file. The user\_authentication variable should be modified to a non-zero value (typically 1) to activate the basic feature. The correct syntax is the following:

```
use_authentication=1
```

- Once this is complete, it is necessary to remove the comment (#) symbol from the authorized\_for\_servicenames and add the usernames created in the htpasswd.users file at the end of the lines. Each one of these authorization lines allow the user who is logging into the machine to be able to view from the web browser the information that would be displayed on the screen. If the username is not entered, or if the authorization line was left commented out, the screen in question would display a message indicating the permissions were not valid. Refer to Appendix G for the modified and original versions of the CGI.CFG files for further information.

- Verify Nagios is configured properly. Affectionately known as the sanity check, a shell command is executed to verify that all configuration settings are in order. Nagios automatically runs a “pre-flight” check before it starts monitoring, but the option is also available to run this check manually before starting the program. This is accomplished by adding the -v command line argument as follows:

```
/usr/local/nagios/bin/nagios -v <main configuration file>
```

The pathname of the main configuration file typically is:

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

- There are four methods for starting Nagios:
  1. Manually as a foreground process
  2. Manually as a background process
  3. Manually as a daemon
  4. Automatically at system boot

For the purposes of this installation procedure, the focus will be on items C and D. Further information of the other startup methods can be found in the Nagios Documentation guide.

- Running Nagios in daemon mode requires the inclusion of the -d switch on the command line as follows:

```
/usr/local/nagios/bin/nagios -d <main configuration file>
```

As mentioned previously, the main configuration file is normally

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

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

- Running Nagios automatically at system boot requires a startup script file and two symbolic links in the appropriate run level. The startup script is normally found in the `/etc/rc.d/init.d` directory, while the symbolic link is located in `/etc/rc.d/rc x.d`. The default is usually `/etc/rc.d/rc3.d`. The init script for Nagios, known as **Nagios**, can be found in Appendix B. The two symbolic links, Start and Kill, can be created to reference the initialization script. Listed below are the links as they appear in one such configuration:

`S96nagios -> /etc/rc.d/nagios` and `K21nagios -> /etc/rc.d/nagios`

- Stopping and restarting Nagios is accomplished via the Init Script that was described previously, or through the kill command. The focus here will be on using the former. The chart shown below provides the information.

## Desired Action Command Description

---

Stop Nagios <code>/etc/rc.d/init.d/nagios stop</code>	This kills the Nagios process
---	-------------------------------

---

Restart Nagios <code>/etc/rc.d/init.d/nagios start</code>	This kills the current Nagios process and then starts Nagios up again
---	---

---

Reload Configuration Data <code>/etc/rc.d/init.d/nagios reload</code>	Sends a SIGHUP to the Nagios process, causing it to flush its current configuration data, reread the configuration files, and start monitoring again
---	--

---

- Install the net-snmp utility. The files are freely available for download at [www.net-snmp.org](http://www.net-snmp.org). Several plug-in's that Nagios uses, i.e.: `check_hpjd`, `check_snmp`, use the snmp libraries for monitoring nodes on the network. As of this writing, the latest stable version available is 5.0.7. Refer to Appendix I for the installation procedure for the net-snmp utility.

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## Section Three – Installation of the Linux Client Software (NRPE)

### 1. Obtain the NRPE Client Program

- Download the NRPE client program from the appropriate website. At the time of this writing, the URL for the website is <http://www.nagios.org>.

### 2. Installing NRPE

- Unpack the distribution file that was downloaded from the Internet. There are two commands:

```
gunzip nrpe- version number .tar.gz and tar -xvf nagios- version number.tar
```

Note: If the zip version of the program was downloaded, the syntax would be the following:

```
unzip nagios- version number.zip
```

Refer to Appendix H for additional documentation on how to compile and install the program.

- Create the installation directory on the remote host for the program. The root or sudo user account should be used when this and future steps are done to insure there are no permissions issues later during the installation. The syntax to use is:

```
mkdir /usr/local/nagios
```

- Create the libexec directory under the nagios directory created in the previous step. The complete path to the directory should read:

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

- The check\_nrpe plugin should be placed on the Nagios **application server** along with the other plugins. The default location is the /usr/local/nagios/libexec directory.

- The nrpe program and the configuration file (nrpe.cfg) should be placed in the directory that was created in the previous steps on the **remote host**. Additional plugins will need to be installed on the remote host in order for the add-on to function properly. The add-ons can be copied from the application server's libexec directory and pasted into the appropriate directory on the remote host.

- Configure the nrpe client to run under the control of inetd or xinetd. Refer to Appendix H for detailed information on the proper syntax for this operation.

- Configure the proper settings on the Nagios host (application server). Examples

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

for configuring the nrpe daemon are found in the sample nrpe.cfg file included in the distribution. The config file resides on the remote host(s) along with the nrpe daemon. The check\_nrpe plugin is installed on the Nagios host (application server). In order to use the check\_nrpe plugin from within Nagios, several things in the host configuration (hosts.cfg) file need to be defined. An example command definition for the check\_nrpe plugin would look like this:

```
define command{
command_name check_nrpe
command_line /usr/local/nagios/libexec/check_nrpe $HOSTADDRESS$ -c $ARG1$
}
```

- In any service definitions that use the nrpe plugin/daemon to get their results, you would set the service check command portion of the definition to something like this (sample service definition is simplified for this example):

```
define service{
host_name someremotehost
service_description someremoteservice
check_command check_nrpe!yourcommand
... etc ...
}
```

- where "your command" is a name of a command that you define in your nrpe.cfg file on the remote host (see the docs in the sample nrpe.cfg file for more information).
- An example of the correct syntax on the application server for running the check\_disk plug-in on the remote host would be the following:

```
define service{
use generic-service ; name of service template to use
host_name bugzilla.mgh.harvard.edu
service_description Free Space
is_volatile 0
check_period 24x7
max_check_attempts 3
normal_check_interval 5
retry_check_interval 1
contact_groups linux-admins
notification_interval 120
notification_period 24x7
notification_options w,u,c,r
check_command check_nrpe!check_disk1
}
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

- The nrpe.cfg file on the remote machine would have something like or similar to the following to allow the application server to run the above plug-in:

```
command[check_disk1]=/usr/local/nagios/libexec/check_disk 80 95 /dev/hda1
```

- It is important to remember that whatever services are to be monitored from the application server, the appropriate plug-ins must be installed on the remote host(s) libexec directory. Also, the accurate command syntax must be present in the nrpe.cfg file on the remote host as well as the services.cfg file on the application server.

Excerpt from posting on the Nagios Users discussion group:

You only have to provide check\_command check\_nrpe!check\_disk1 in yourservices.cfg. The nrpe on the target machine should translate that into ../check\_disk 80 95 ...

-----Ursprüngliche Nachricht-----

Von: Kaplan, Andrew H. [mailto:AHKAPLAN@PARTNERS.ORG]

Gesendet: Mittwoch, 2. Oktober 2002 15:24

An: nagios-users@lists.sourceforge.net

Betreff: [Nagios-users] Incorrect command line arguments supplied error message

Hi,

I have installed the nrpe plugin on the nagios host and the daemon on a remote host. However,

when I try to monitor the disk space usage on the remote host, I encounter the following error on the web browser:

**INCORRECT COMMAND LINE ARGUMENTS SUPPLIED**

The text that I added on the host machine is as follows:

checkcommands.cfg:

```
# 'check_nrpe' command definition
```

```
define command{
```

```
command_name check_nrpe
```

```
command_line /usr/local/nagios/libexec/check_nrpe $HOSTADDRESS$ -c $ARG1$
```

```
}
```

services.cfg

```
# Service definition
```

```
define service{
```

```
use generic-service ; Name of service template to use
```

```
host_name bugzilla.mgh.harvard.edu
```

```
service_description Free Space
```

```
is_volatile 0
```

```
check_period 24x7
```

```
max_check_attempts 3
```

```
normal_check_interval 1
```

```
contact_groups linux-admins
```

```
notification_interval 120
```

```
notification_period 24x7
```



# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

notification\_options w,u,c,r

check\_command

check\_nrpe!command[check\_disk1]=/usr/local/nagios/libexec/check\_disk 80 95  
/dev/hda5

I installed the nrpe daemon and nrpe.cfg files onto the remote host at /usr/local/nagios. I created the

/usr/local/nagios/libexec directory there as well. I copied the check\_disk, check\_load, check\_procs, and the check\_users

plugins from the host machine into the libexec directory on the remote machine. The nrpe file was created in the

/etc/xinetd.d directory. Also, I added the following line to the nrpe.cfg file:

command[check\_disk1]=/usr/local/nagios/libexec/check\_disk 80 95 /dev/hda5

What arguments did I miss or have the wrong syntax for?

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## Section Four – Installation of the Windows Client Software (NSClient) (based on the documentation for NSClient)

### 1. Obtain the NSClient Program

- Download the NSClient program from the appropriate website. At the time of this writing, the URL for the website is <http://nsclient.ready2run.nl/> .
- Unpack the distribution file that was downloaded from the Internet. Any zip utility will do so long as it successfully expands the executable and library files of the program.

### 2. Installing NSClient

- Log onto the Windows machine as Administrator or as a user that has administrator access to the system.
- On the Windows machine copy pNSClient.exe in any directory on the machine you want to monitor. i.e.. (c:\nsclient).
- Open a command prompt in the installation directory
- Run the following command : >pNSClient /install
- Start the service 'Netsaint NT Agent' in the services applet of the control panel. The installation will create an entry for the service in the registry and create a new key to store parameters. The created key is the following: HKEY\_LOCAL\_MACHINE\SOFTWARE\NSClient
- Add the following lines to the checkcommands.cfg file:

```
command[check_nt_disk]=$USER1$/check_nt -H $HOSTADDRESS$ -p 1248 -v USEDDISKSPACE -I $ARG1$ -w $ARG2$ -c $ARG3$
command[check_nt_cpuload]=$USER1$/check_nt -H $HOSTADDRESS$ -p 1248 -v CPULOAD -I $ARG1$
command[check_nt_uptime]=$USER1$/check_nt -H $HOSTADDRESS$ -p 1248 -v UPTIME
command[check_nt_clientversion]=$USER1$/check_nt -H $HOSTADDRESS$ -p 1248 -v CLIENTVERSION
command[check_nt_process]=$USER1$/check_nt -H $HOSTADDRESS$ -p 1248 -v PROCSTATE -I $ARG1$
command[check_nt_service]=$USER1$/check_nt -H $HOSTADDRESS$ -p 1248 -v SERVICESTATE -I $ARG1$
command[check_nt_memuse]=$USER1$/check_nt -H $HOSTADDRESS$ -p 1248 -v MEMUSE -w $ARG1$ -c $ARG2$
command[check_nt_pagingfile]=$USER1$/check_nt -H $HOSTADDRESS$ -p 1248 -v COUNTER -I "\\Paging File(_Total)\% Usage", "Paging File usage is %.2f%%" -w $ARG1$ -c $ARG2$
```

### 3. Uninstallation

- Go to the installation directory and run the following command: >pNSClient /uninstall. All entries in the registry will be removed as well as the definition of the service.

### 4. Configuration

- There are two parameters you can change: the port (default: 1248) and the password (default: 'None'). These two settings are store in the registry and can

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

only be changed using 'regedit'. Open the following key and change the values if needed : HKEY\_LOCAL\_MACHINE\SOFTWARE\NSClient\Parms

- If you change the password, you will have to use the -s <password> with every request you send to NSClient.
- Refer to Appendix A for the complete documentation for NSClient. Included is the syntax for the plug-in's. Although NSClient was originally written for Netsaint, it has been successfully proven to work with Nagios.

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## Section Five – Using the check\_snmp plug-in for non-HP printers

The check\_snmp plug-in can be used for printers where one or more of the following conditions apply:

- The printer is not a Hewlett-Packard brand
- The printer does not use a Jet-Direct network card
- The network card the printer uses is not Jet-Direct compatible
- The printer has SNMP capability

The check\_snmp plug-in uses standard snmp versions 1 and 2 MIB's (management information blocks) to send out requests to the various clients. It requires the net-snmp package that was mentioned earlier, and it also needs the Net::SNMP perl module. Assuming the net-snmp program has been installed on the Nagios server, what follows is a listing of the steps to install the perl module and also the correct syntax for the check\_snmp command itself. **Note: The Nagios application should be disabled before proceeding.**

1. While logged onto the Nagios server, access the CPAN website for the Net::SNMP module. The url for the package is the following:

<http://search.cpan.org/author/DTOWN/>

2. Click on the download link to copy the file onto the server.

3. Unpack the file using the gunzip (if necessary) and tar utilities. A directory called [Net-SNMP-4.0.3](#) will be created. Change to the directory.

4. There are two approaches for installing the perl module. The first has the installation conducted on-line with all the necessary configurations being automatically completed. The second has the administrator doing a manual installation. The former will be discussed here. Refer to Appendix J for more detailed information concerning the manual installation. To install the Net::SNMP module and all of its dependencies directly from the Comprehensive Perl Archive Network (CPAN) execute the command:

```
perl -MCPAN -e "install Net::SNMP"
```

5. The check\_snmp plug-in should be tested prior to configuring the remote printers to be monitored by Nagios. The root or sudo account can be used to accomplish this task. While in a terminal command shell, go to the appropriate directory where the plug-in is located, which for argument can be the default /usr/local/nagios/libexec, and type the following:

```
./check_snmp -H ip address -C public -o sysDescr.0
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

The printer should return some SNMP information string, which describes its current status, as well as various information about its firmware and other components. If an error message similar to “**SNMP problem – No data returned from host.**” is returned one of several issues may be occurring:

- The command syntax is wrong
- The printer does not have an SNMP stack

If the printer does not have an SNMP stack, the vendor can be contacted to determine if SNMP is either not configured on the system or can be installed onto the printer. However, if the vendor does not have any SNMP solution available, then the printer cannot be monitored via the check\_snmp plug-in. The only recourse available in that case would be to use the check\_ping utility to verify the printer being available on the network.

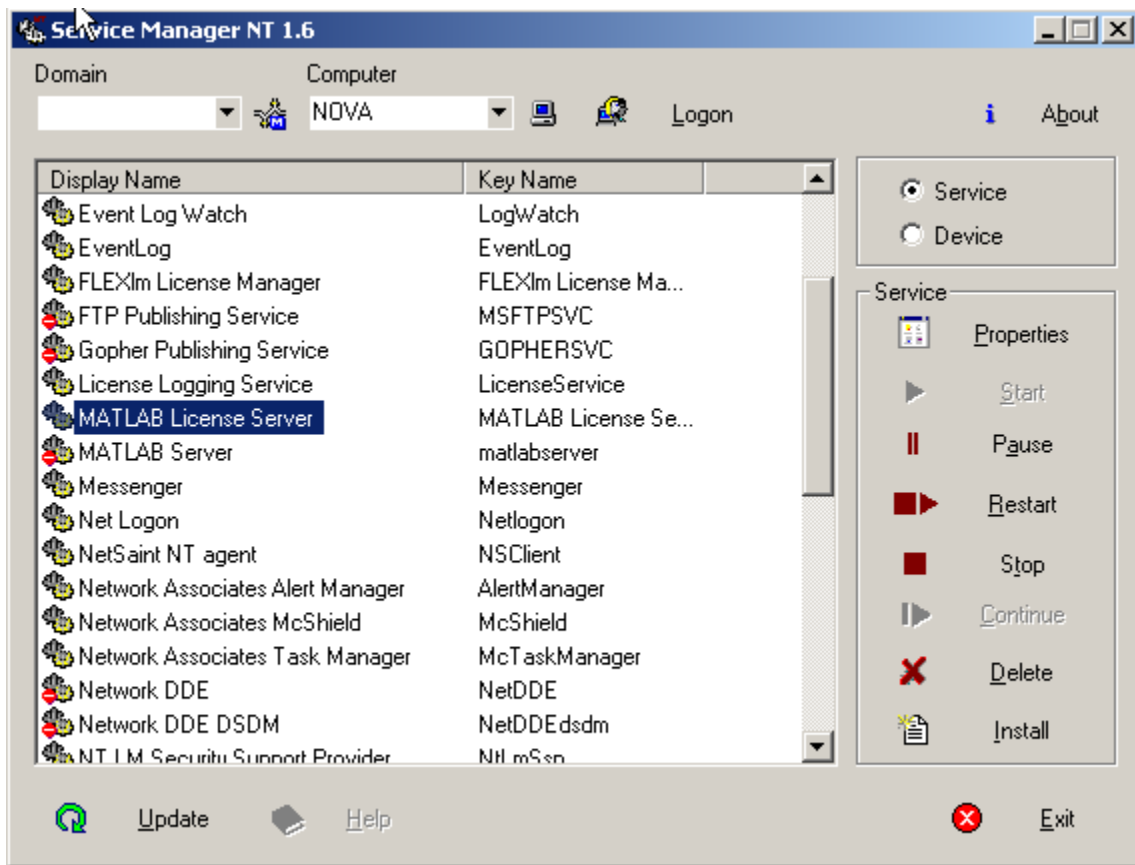
# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## Section Six – Configuration of the check\_flexlm plug-in

The check\_flexlm plug-in is used to monitor the flexlm license server common to various mathematical applications. Several such programs include Matlab and IDL. The license server is available for Windows, Linux, and several other platforms. Monitoring the flexlm service can be accomplished using methods specific to the platform. The Windows approach will be discussed briefly and will be followed by the Linux analysis.

### Windows:

The Windows version of flexlm is monitored via the NSClient or Ntray utility. For the purposes of this document, the NSClient utility and Matlab license server will be used as references. After installing the NSClient program on the license server, the administrator can configure the service to be monitored from the Nagios application server. A tool that provides access to the list of services running on the Windows machine is Service Manager NT. This utility is freely downloadable from the Internet. After connecting to the remote Windows server, locate the application's license server. The figure below is a screenshot of one example:



# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

To configure the license server to be monitored, modify the services.cfg file on the Nagios server. One such example follows:

```
# Service definition
define service{
    use generic-service ; Name of service template to use
    host_name nova
    service_description MATLAB License Server
    is_volatile 0
    check_period 24x7
    max_check_attempts 3
    normal_check_interval 5
    retry_check_interval 1
    contact_groups nt-admins
    notification_interval 120
    notification_period 24x7
    notification_options w,u,c,r
    check_command check_nt_service!'MATLAB License Server'
}
```

The check\_command line defines the executable for monitoring the license server. Special note should be given to the quotation marks. The Service Manager program advertised the Display Name of the service, which happened to be comprised of several words. The inclusion of the quotation marks tells the Linux operating system, where the Nagios application resides, the text that is included within is part of one filename rather than several. This allows the administrator have user-friendly names represent the services on the remote system.<sup>1</sup>

## Linux:

The configuration for check\_flexlm involves three files on the license server, and the services.cfg file on the Nagios server. The three files are:

1. utils.pm
2. check\_flexlm.pl
3. nrpe.cfg

The utils.pm file is the utility drawer or resource file the Nagios plug-ins use for referencing the locations of various daemons that are to be monitored on the client. If the file is not on the license server, it should be copied to the same directory on the client where the plugin's are located. Once the file has been copied and/or located, it should be modified to accommodate the location of the lmstat file, which is the license server daemon. The line in the file that is modified is

---

<sup>1</sup> If there were several dependencies associated with a particular service that needed to be monitored. The above example would apply with addition of commas separating each dependency. For example: check\_nt\_service!'OTG DiskXtender','OTG License Server','OTG MediaStor'

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

**\$PATH\_TO\_LMSTAT = "path to the lmstat file"**

The check\_flexlm.pl file, written by Subhendu Ghosh, is a perl script that will conduct the actual inquiry to the lmstat file. It relies on the utils.pm file for the correct path to the flexlm license daemon. The modification needed in this file occurs on the first line of the file. It is here the location of Perl on the client computer is designated for the script. To find the location of Perl, run the following command:

**which perl**

The output of the command should be incorporated into the script for proper functionality. The nrpe.cfg file should have a line similar to the one shown below added to have the nrpe client program check on the status of the flexlm daemon. The -F option specifies the location of the license.dat file. The full pathname to the file should be provided. After the changes have been made to the client system, restart the xinetd.d services on the client with the following:

**/etc/rc.d/init.d/xinetd restart**

Prior to making changes the services.cfg file on the Nagios application server, disable the application. After that has been done, the services.cfg file should subsequently have a similar entry to the following added to the configuration: The Nagios server should be restarted. After several minutes, the status of the flexlm license server should appear on Services Detail screen.



# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## Section Seven – Configuration of the 2D and 3D Status Maps

The configuration of the 2D and 3D Status maps involves modifying several files as well as downloading the icons needed by the status maps on the Nagios server. The latter requires downloading specific software onto the client machine.

The icons are available at the Nagios home page, [www.nagios.org](http://www.nagios.org), and are located in the Downloads section. The default location of the icon files is the /usr/local/nagios/share/images/logos directory. The screen capture shown below provides an example directory listing.

```
apple.gd2  frechs40.png  mandrake.gd2  my_pc.jpg      my_ws.gd2      openshd.png  sun40.gd2    unicos.jpg
apple.gif  hp-printer40.gd2  mandrake.gif  my_pc.png     my_ws.gif     redhat.gd2   sun40.gif    unicos.png
apple.jpg  hp-printer40.gif  mandrake.jpg  my_router.gd2  my_ws.jpg     redhat.gif   sun40.jpg    unknown.gd2
apple.png  hp-printer40.jpg  mandrake.png  my_router.gif  my_ws.png     redhat.jpg   sun40.png   unknown.gif
beos.gd2  hp-printer40.png  my_cache.gd2  my_router.jpg  nagios.gd2    nagios.gd2   router40.gd2  sunlogo.gd2
beos.gif  hpux.gd2         my_cache.gif  my_router.png  nagios.gif    router40.gd2  sunlogo.gif  win40.gd2
beos.jpg  hpux.gif         my_cache.jpg  my_routerswitch.gd2  nagiosvrm1.png  router40.gif  sunlogo.jpg  win40.gif
beos.png  hpux.jpg        my_cache.png  my_routerswitch.gif  next.gd2        router40.jpg  sunlogo.png  win40.png
caldera.gd2  hpux.png       my_hub.gd2    my_routerswitch.jpg  next.gif        router40.png  switch40.gd2  yellowdog.gd2
caldera.gif  irix.gd2       my_hub.gif    my_routerswitch.png  next.jpg        slackware.gd2  switch40.gif  yellowdog.gif
caldera.jpg  irix.gif       my_hub.jpg    my_server.gd2    next.png       slackware.gif  switch40.jpg  yellowdog.jpg
caldera.png  irix.jpg       my_hub.png    my_server.gif    ng-switch40.gd2  slackware.jpg  switch40.png  yellowdog.png
cat1900.gd2  irix.png      my_lanswitch.gd2  my_server.jpg  ng-switch40.gif  slackware.png  turbolinux.gd2
cat2900.gd2  linux40.gd2   my_lanswitch.gif  my_server.png  ng-switch40.jpg  stamped.gd2   turbolinux.gif
```

```
[ahk@u0085542-udp519310uds logos]# pwd
/usr/local/nagios/share/images/logos
[ahk@u0085542-udp519310uds logos]#
```

The files that need to be modified on the Nagios server are the following:

1. cgi.cfg
2. hostextinfo.cfg
3. hosts.cfg

The cgi.cfg file uses a command line to reference the hostextinfo file for information concerning the status map icons. The syntax is as follows:

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

The hostextinfo.cfg file provides the filenames of the images used in the status map screens. Like other Nagios configuration files, a template is used to organize the information for each host. One example entry is shown below:

```
# Galaxy definition
define hostextinfo{
    host_name          galaxy
    icon_image         win40.jpg
    icon_image_alt    Galaxy
    vrm1_image         win40.jpg
    statusmap_image   win40.jpg
    gd2_image          win40.jpg
    register           1
}
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

The hosts.cfg file is used to provide a hierarchical organization to the hosts on the map pages. The members of the chain of command on which Nagios reports are divided into two categories: parent and child. A parent is a node that has other nodes subordinate to it, while a child node is one that is subsidiary to another node. A node can be a parent, child, or both. The existence or absence of a single line in the template of host determines whether the host is a child or parent. The first example that follows shows the template setting for a parent node.

```
# 'founders_5' host definition
define host{
    use                generic-host        ; Name of host template to use

    host_name          founders_5
    alias              founders_5
    address            127.0.0.1
    check_command      check-host-alive
    max_check_attempts 10
    notification_interval 480
    notification_period workhours
}
```

The absence of the “parents” line indicates this host to be a parent. The following example includes the setting in question, and thereby designates the host as a child node.

```
# 'linux1' host definition
define host{
    use                generic-host        ; Name of host template to use

    host_name          Oneofxeon
    alias              Oneofxeon
    address            132.183.12.21
    parents            stargate
    check_command      check-host-alive
    max_check_attempts 10
    notification_interval 480
    notification_period 24x7
    notification_options d,u,r
}
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## Using virtual, or dummy, nodes for organizing network assets:

When organizing the layout of the network infrastructure, the use of dummy nodes as a means of compartmentalizing and classifying nodes into their respective subdivisions and departments can help provide order and viewability to the status maps.

Dummy, or virtual, nodes are those that do not actually exist, but are configured as hosts via the hosts, hostextinfo, and services.cfg files. An example of a virtual node would be the floor of a building. The building floor is not an actual host, i.e.: computer, printer, etc, but is used as a means of classifying those network components which are located there. This virtual node would be the parent of all those hosts on which they are located. The following example provides an explanation of this concept.

There are several servers located in the Founders Building on the fifth floor. To better distinguish them from other locations, a virtual host is created using the three files mentioned earlier. The hosts file listing has the following information:

```
# 'founders_5' host definition
define host{
    use                generic-host           ; Name of host template to use

    host_name          founders_5
    alias               founders_5
    address             127.0.0.1
    check_command       check-host-alive
    max_check_attempts 10
    notification_interval 480
    notification_period workhours
}
```

The host definition shown above uses the 127.0.0.1 or loopback address. The IP stack that is being monitored is in actuality the Nagios server itself. This is what makes the host virtual as opposed to tangible.

The hostextinfo file for the above host entry is shown below.

```
# Founders_5_definition
define hostextinfo{
    host_name          founders_5
    icon_image         my_lanswitch.jpg
    icon_image_alt     Organizational Unit
    vrml_image         my_lanswitch.jpg
    statusmap_image    my_lanswitch.jpg
    gd2_image          my_lanswitch.jpg
    register           1
}
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

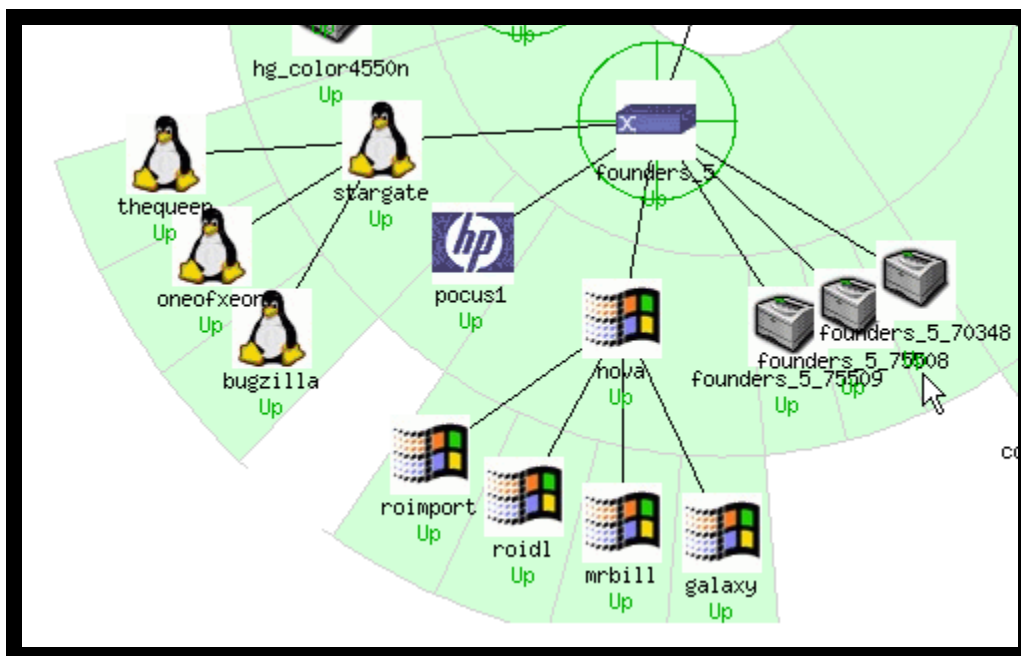
The icon image that can be used is left to the discretion of the administrator. Similarly, the icon\_image\_alt entry is also open to interpretation. The above example uses the term Organizational Unit to define the host as the area on the network where the child nodes associated with the virtual node are located.

The services.cfg file has the following entries configured for the virtual node.

```
# Service definition
define service{
    use                generic-service      ; Name of service template to use

    host_name          Founders_5
    service_description Organizational Unit
    is_volatile         0
    check_period        24x7
    max_check_attempts 3
    normal_check_interval 5
    retry_check_interval 1
    contact_groups      linux-admins
    notification_interval 240
    notification_period 24x7
    notification_options c,r
    check_command        check_ping!100.0,20%!500.0,60%
}
```

The check\_command line uses the check\_ping plug-in to test the connectivity to the host. Because the network address of the dummy host was the 127.0.0.1, the Nagios server is actually pinging itself. The end result of the configuring of the above three files is a network map scheme similar to the one shown below:



# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## Section Eight – Remote Host Monitoring via SSH

There are occasions where the monitoring of a remote host must be made via a secure network connection. If insecure port is not available, remote clients can be monitored via the ssh network protocol. When this method is used, the nagios server or the remote system does not use the check\_nrpe and nrpe daemons.

Prior to configuring the secure connection, a third-party program should be installed on the server to better facilitate the management of encrypted passwords and public and private keys. The program that best suits this purpose is the Keychain program developed by Gentoo. The following excerpt is from the freshmeat website.

**keychain helps you to manage RSA and DSA keys in a convenient and secure manner. It acts as a front-end to SSH-agent, but allows you to easily have one long running SSH-agent process per system, rather than the norm of one SSH-agent per login session. This dramatically reduces the number of times you need to enter your passphrase - with keychain, you only need to enter a passphrase once every time your local machine is rebooted. keychain also makes it easy for remote cron jobs to securely "hook in" to a long running SSH-agent process, allowing your scripts to take advantage of RSA and DSA keys.<sup>2</sup>**

The installation procedure for the Keychain application is as follows:

1. Create the ssh key for each user with the following command:  
**ssh-keygen -t dsa**
2. Copy the public key to the remote client.
3. Verify the .cshrc or .bash\_profile environmental variable files have the following lines -  
**keychain /home/nagios/.ssh2/id\_dsa**  
**source /home/nagios/.keychain/nagios server hostname**
4. Configure Nagios to use SSH when monitoring the remote host(s).

Refer to Appendix K for the setup documentation associated with the Keychain application.

The Nagios server and its client(s) must have the SSH daemon installed on their systems. Most Linux distributions come with the server and client software bundled as part of their installation package. If the administrator opts for the most recent version, it can be downloaded from [www.openssh.org](http://www.openssh.org). The procedure for building the protocol from its source is included in the download package available at that site.

After the ssh daemon has been installed on the appropriate machines, it can be used by Nagios to monitor remote hosts. To accomplish this, the

---

<sup>2</sup> by Daniel Robbins - Thursday, August 30th 2001 22:02 PST

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

checkcommands, and services.cfg files need to be modified to have the appropriate configurations. The checkcommands.cfg file needs to have a define command line configured for every instance that ssh will be used to monitor the remote host. Hence, if there are five services that need to be monitored on the remote host, then five command definitions must be entered into the file for that one host. One example is shown below:

```
# 'check_blue_httpd' definition
define command{
    command_name          /usr/bin/ssh blue /usr/local/nagios/check_http
    command_line          /usr/bin/ssh blue /usr/local/nagios/check_http
}
# 'check_blue_check_disk' definition
define command{
    command_name          /usr/bin/ssh blue /usr/local/nagios/check_disk
    command_line          /usr/bin/ssh blue /usr/local/nagios/check_disk
}
```

The command to run ssh by the configuration shown previously is located in the services.cfg file. The syntax of the command is the following:

***/directory path/ssh hostname /directory path/nagios plug-in***

For example, if the ssh daemon was located in /usr/bin on the Nagios server, and the plug-in check\_http was located in the /usr/local/nagios directory on the “blue” webserver, the syntax would be:

***/usr/bin/ssh blue /usr/local/nagios/check\_http***

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## Section Nine – Monitoring HP-UX Workstations and Servers

C. Benseid [benny@bennyvision.com] posted the e-mail shown below on the Nagios forum site.

Hey folks,

I've gone ahead and built native HP-UX depots for NRPE. I have built them for 10.20, 11.00, and 11i. They're very simple, containing only the README from NRPE, as well as nrpe and check\_nrpe binaries. Feel free to snag them from:

<http://www.bennyvision.com/projects/nagios/>

Benny

The author of this document had the opportunity to test the ported version of the nrpe client and plug-ins on HP-UX 10.20, and 11.00 systems. The results of the ported version were that it worked without any difficulty. Congratulations to C. Benseid.

The following procedure was undertaken to install the nrpe client on the HP-UX 10 and 11 systems, and to have it automatically start on system boot.

1. Create a shared (NFS) filesystem location for the depot and plug-in files. Download the files from the Internet and save them to the shared location.
2. Prior to installing the nrpe client onto the HP-UX machines, prepare a generic nrpe.cfg file that can be copied onto and subsequently modified on each workstation. Have it located in the shared filesystem mentioned in the previous step.
3. Copy the appropriate depot, plug-in, and generic nrpe.cfg files to the /tmp directory of the workstation.
4. At the workstation console, log in as root or a sudoer account.
5. Invoke SAM. The pathname to the binary is /usr/sbin/sam
6. Select the **Software Management** option.
7. Select **Install Software to Local Host**
8. Change the **Source Depot Type** to Local Directory.
9. Under **Source Depot Path**, provide the full pathname to the depot file. For

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

example, if the depot file is in the /tmp directory, the pathname would be /tmp/nrpe-1.8-B.*versionnumber*.depot

10. Mark the depot for installation. SAM will first conduct an analysis of the file. Any errors or warnings should be corrected before proceeding. If there are no problems reported, the nrpe client can be installed onto the system. The location of the program will be **/opt/nrpe**.
11. Change to the /opt/nrpe directory and create the libexec directory using the mkdir command.
12. Move the plugin's gzip file from the /tmp directory to libexec.
13. Run the gzip and tar commands to uncompress and extract the plugins.
14. Copy the generic nrpe.cfg file to the /opt/nrpe directory, and modify it to accommodate the configuration and services of the HP workstation.
15. Change to the /sbin/init.d directory.
16. Run the vi, or comparable, text editor utility to create a file called nrpe. The file will have the following text:  
  
**/opt/nrpe/bin/nrpe -c /opt/nrpe/nrpe.cfg -daemon**  
  
The above command will have the nrpe client run as a daemon on the workstation. The -c option has the binary file reference the nrpe.cfg file in the directory shown above.
17. Change the permissions of the newly created file to read and execute for all users via the **chmod 555 nrpe** command syntax.
18. Change the ownership of the /sbin/init.d/nrpe file to user bin, group bin via the **chown bin:bin nrpe** command.
19. Change to the /sbin/rc2.d directory.
20. Create a symbolic link to the /sbin/init.d/nrpe file using the following command:  
**ln -s /sbin/init.d/nrpe Snumbernrpe**  
  
The number that is used can be anyone that is currently available.
21. Manually start the nrpe client program using the script that was created in the /sbin/init.d directory.



# **PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM**

22. Modify the appropriate files on the Nagios server to accommodate the new client, and restart the application as necessary.

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## APPENDIX A – NSCLIENT DOCUMENTATION

February 2002 - Yves Rubin - [rubiyz@yahoo.com](mailto:rubiyz@yahoo.com)  
Official Web Site: <http://www.ifrance.com/rubiyz/>  
FAQ J <http://www.ifrance.com/rubiyz/faq.htm>

### Purpose

NSClient has been developed to get vital information concerning Windows NT servers from Netsaint. Virtually every performance counter of Windows can be retrieved by the plugin.

### License

This tool has been developed in the same spirit as Netsaint, which means that it is free-ware and release under the GNU General Public License.

### Requirement

Windows NT 4 or Windows 2000 ENGLISH or GERMAN version.  
Need the pdh.dll and psapi.dll libraries to work

### For the compilation:

Jedi Code Library (<http://www.delphi-jedi.org>)

Full Win32 Api from Marcel van Brakel (<http://www.delphi-jedi.org>)

### History

10.02.2002 – 1.06.2

#### Beta version

- New psapi.dll, pdh.dll
- new jedi library
- use of jwaPDH library
- new language in Counters.defs
- compiled with Delphi 6

13.09.2001 – 1.06.0

#### Beta version

- Fix the service start up error message
- Improve the event log handling
- Should work on any Windows version. Different platform languages shouldn't be a problem anymore.

09.05.2001 – 1.05.2

#### Beta version

- Fix some issue with the German support
- Build using the Jedi Code release 1.2
- Add the psapi.dll library which should fix the PROCSTATE issue

18.04.2001 - 1.05.0

#### Beta version.

- Add German support to NSClient

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

27.03.2001 - 1.04.5

Release version.

- Some documentation changes.
- Fix the wrong check\_nt.c in UnixSource folder.

06.03.2001 - 1.04.4

- Change the way the description is handled in the custom counter requests to improve the flexibility.

28.02.2001 - 1.04.3

- Fix an issue with process check under Windows 2000 SP1

26.02.2001 - 1.04.2

- Add the "custom counter" query feature
- Fix some errors with pdh.dll calls
- New version of pdh.dll
- Better information in the EventLog

06.12.2000 - 1.03

- Add the memory usage feature

29.11.2000 - 1.02

- First release

Features

Check CPU load on single or multiprocessors system during the last 24 hours.

Check disk space

Check memory use

Check uptime time of the machine

Check services states

Check processes

Can check every available performance counter

Installation

On the Windows machine

1. Copy pNSClient.exe in any directory on the machine you want to monitor. i.e.. (c:\nsclient).
2. Open a dos prompt in the installation directory
3. Run the following command : >pNSClient /install
4. Start the service 'Netsaint NT Agent' in the services applet of the control panel.
5. The installation will create an entry for the service in the registry and create a new key to store parameters. The created key is the following:

HKEY\_LOCAL\_MACHINE\SOFTWARE\NSClient

Please read if you are using another Windows version than English!

A new file "counters.defs" has been added in the NSClient folder. This file is the resource file for counters description for every language. At this time, only English and german are declared. You can easily add another language by

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

defining all counters for this language. The language code can be found in the Windows Event Log in the NSClient startup message. I would be please to update this file if you provide me with the correct information for other languages.

On the Unix machine

1. Copy the file 'check\_nt' under the libexec folder of Netsaint
2. Add the following lines to the commands.cfg file:

```
# NSClient
command[check_nt_disk]=$USER1$/check_nt -H $HOSTADDRESS$ -p 1248 -v USEDDISKSPACE -I $ARG1$ -w $ARG2$ -c $ARG3$
command[check_nt_cpuload]=$USER1$/check_nt -H $HOSTADDRESS$ -p 1248 -v CPULOAD -I $ARG1$
command[check_nt_uptime]=$USER1$/check_nt -H $HOSTADDRESS$ -p 1248 -v UPTIME
command[check_nt_clientversion]=$USER1$/check_nt -H $HOSTADDRESS$ -p 1248 -v CLIENTVERSION
command[check_nt_process]=$USER1$/check_nt -H $HOSTADDRESS$ -p 1248 -v PROCSTATE -I $ARG1$
command[check_nt_service]=$USER1$/check_nt -H $HOSTADDRESS$ -p 1248 -v SERVICESTATE -I $ARG1$
command[check_nt_memuse]=$USER1$/check_nt -H $HOSTADDRESS$ -p 1248 -v MEMUSE -w $ARG1$ -c $ARG2$
command[check_nt_pagingfile]=$USER1$/check_nt -H $HOSTADDRESS$ -p 1248 -v COUNTER -I "\\Paging File(_Total)\% Usage", "Paging File usage is %.2f %" -w $ARG1$ -c $ARG2$
```

## Uninstallation

Go to the installation directory and run the following command:

```
>pNSClient /uninstall
```

All entries in the registry will be removed as well as the definition of the service..

## Configuration

There are two parameters you can change: the port (default: 1248) and the password (default: 'None'). These two settings are store in the registry and can only be changed using 'regedit'. Open the following key and change the values if needed :

```
HKEY_LOCAL_MACHINE\SOFTWARE\NSClient\Parms
```

If you change the password, you will have to use the -s <password> with every request you send to NSClient.

## Plugin syntax

### CPUload

#### Syntax:

```
check_nt -H <hostname> -p <port> -v CPULOAD -I <minutes range>,<warning percent>,<critical percent> <minutes range> between 1 and 1440 (24 hours) <warning percent> and <critical percent> : thresholds between 1 and 100.
```

You can check several intervals in one shot. The following command get the average for the last 10min., 60min. and 24hours.

Example:

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

`./check_nt -H 192.168.1.1 -p 1248 -v CPULOAD -I 10,80,95,60,80,95,1440,80,95`  
Disk usage

Syntax:

`check_nt -H <hostname> -p <port> -v USEDDISKSPACE -I <drive letter> [-w <warning percent> ] [-c <critical percent>]` drive letter should be only one character. <warning percent> and <critical percent> : thresholds between 1 and 100.

Example:

`./check_nt -H 192.168.1.1 -p 1248 -v USEDDISKSPACE -I C -w 80 -c 90`

Uptime

Syntax: .

`/check_nt -H <hostname> -p <port> -v UPTIME`

This plugin doesn't care about warning or critical values. Only the uptime of the machine is received.

Services states

Syntax:

`check_nt -H <hostname> -p <port> -v SERVICESTATE [-d SHOWALL] -I <service 1>[,<service 2>,<service 3>,...]`

<service> should be the real name of the service, not the displayed name. You can find this information when going to the registry for the corresponding service:

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services

or use the free utility: 'Service Manager NT' :

<http://www-rnks.informatik.tucottbus.de/~fsch/english/nttols.htm>

-d SHOWALL can be specified if you want to see all tested services including started ones. You can specify several services in one request. No blank should appear in the list ! If not all services are running, you get the faulty one(s) and a critical state.

Example:

`./check_nt -H 192.168.1.1 -p 1248 -v SERVICESTATE -d SHOWALL -I LanmanServer,Schedule`

Processes states

Syntax:

`check_nt -H <hostname> -p <port> -v PROCSTATE [-d SHOWALL] -I <process 1>[,<process 2>,<process 3>,...]`

<processes> You can find process name in the Windows NT Task Manager.

-d SHOWALL can be specified if you want to see all tested processes including

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

running ones.

You can specify several processes in one request. No blank should appear in the list !

If not all processes are running, you get the faulty one(s) and a critical state.

Example:

```
./check_nt -H 192.168.1.1 -p 1248 -v PROCSTATE -d SHOWALL -I McShield.exe,RASMAN.EXE
```

Client version

Syntax:

```
check_nt -H <hostname> -p <port> -v CLIENTVERSION
```

Return the NSClient version. No warning or critical threshold can be specified.

Memory usage

Syntax:

```
check_nt -H <hostname> -p <port> -v MEMUSE [-w <warning percent> ] [-c <critical percent>]  
<warning percent> and <critical percent> : thresholds between 1 and 100.
```

Example:

```
./check_nt -H 192.168.1.1 -p 1248 -v MEMUSE -w 80 -c 90
```

"Custom counter"

Syntax:

```
check_nt -H <hostname> -p <port> -v COUNTER -I <counter name>[,<counter description>] [-w  
<warning percent> ] [-c <critical percent>]
```

- <counter name> is the exact description of the Windows counter. It must be enclosed in " and the \ should be doubled.
- <counter description> is the description which is displayed in the "Service Information" column. This string is passed to a printf command, which means that you need to give a precise and specific syntax to make it work. Have a look at any printf reference to get the full story about this C command.

Here are some examples:

"Paging file usage is %.2f %%" (the %.2f means that the result will be displayed with two decimal digits. The double % represents a % in the printf syntax. "%f %% of the paging file used" (the %f means that the result will be displayed with no decimal digit).

The results returned by NSClient have five decimal digits.

<warning percent> and <critical percent> : thresholds between 1 and 100. If

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

<warning percent> is higher than <critical percent>, the check is reversed.

Example:

```
./check_nt -H 192.168.1.1 -p 1248 -v COUNTER -I "\\Paging File(_Total)\% Usage","Paging file usage is %.2f %%" -w 80 -c 90
```

```
./check_nt -H 192.168.1.1 -p 1248 -v COUNTER -I "\\Process(_Total)\Thread Count","Thread Count: %.f" -w 600 -c 800
```

```
./check_nt -H 192.168.1.1 -p 1248 -v COUNTER -I "\\Server\Server Sessions","Server Sessions: %.f" -w 20 -c 30
```

## Technical information

### Windows agent

NSClient has been developed using Borland Delphi 5. It is installed as a service. Every five seconds, NSClient query Windows to get the CPU load and store this information in a circular buffer which keeps the measures for the last 24 hours. That's the only task of the agent if no request is sent to him. Have a look at the source code in case you want to know more about it. It's well documented so you shouldn't have too much pain to figure out how it works. Some functions and pieces of code where found on the internet, so I thank very much the programmers who release them as open source. When possible, credits are left in the code.

### Unix plug in

The Unix plugin has been written in C, using a template by Ethan Galstad. It mostly uses common functions and should be compiled in the same directory as all other plugins. I hope that it will be included in the common distribution.

### Problems

In case of problems, have a look at the EventLog of the machine. You should find messages from Netsaint with an error code in hexadecimal. The corresponding english description can be found in NTSource\pdhmsg.pas.

### Reported issues

Wrong memory information reported by NSClient on some machine having lots of memory (bigger than 1GB). This is a Windows NT 4 bug and, according to Microsoft, should be corrected in SP7.

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## APPENDIX B: INIT SCRIPT FOR NAGIOS - LOCATED IN THE /ETC/RC.D/INIT.D DIRECTORY

```
#!/bin/sh
#
# chkconfig: 345 99 01
# description: Nagios network monitor
#
# File : nagios
#
# Author : Jorge Sanchez Aymar (jsanchez@lanchile.cl)
#
# Changelog :
#
# 1999-07-09 Karl DeBisschop <kdebisschop@infoplease.com>
# - setup for autoconf
# - add reload function
# 1999-08-06 Ethan Galstad <nagios@nagios.org>
# - Added configuration info for use with RedHat's chkconfig tool
# per Fran Boon's suggestion
# 1999-08-13 Jim Popovitch <jimpop@rocketship.com>
# - added variable for nagios/var directory
# - cd into nagios/var directory before creating tmp files on startup
# 1999-08-16 Ethan Galstad <nagios@nagios.org>
# - Added test for rc.d directory as suggested by Karl DeBisschop
# 2000-07-23 Karl DeBisschop <kdebisschop@users.sourceforge.net>
# - Clean out redhat macros and other dependencies
#
# Description: Starts and stops the Nagios monitor
# used to provide network services status.
#
status ()
{
if test ! -f $NagiosRun; then
echo "No lock file found in $NagiosRun"
return 1
fi
NagiosPID=`head -n 1 $NagiosRun`
if test -x $NagiosCGI/daemonchk.cgi; then
if $NagiosCGI/daemonchk.cgi -l $NagiosRun; then
return 0
else
return 1
fi
else
if ps -p $NagiosPID; then
return 0
else
return 1
fi
fi
return 1
}
killproc_nagios ()
{
if test ! -f $NagiosRun; then
echo "No lock file found in $NagiosRun"
return 1
fi
NagiosPID=`head -n 1 $NagiosRun`
```



# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
kill $2 $NagiosPID
}
# Source function library
# Solaris doesn't have an rc.d directory, so do a test first
if [ -f /etc/rc.d/init.d/functions ]; then
./etc/rc.d/init.d/functions
elif [ -f /etc/init.d/functions ]; then
./etc/init.d/functions
fi
prefix=/usr/local/nagios
exec_prefix=${prefix}
NagiosBin=${exec_prefix}/bin/nagios
NagiosCfg=${prefix}/etc/nagios.cfg
NagiosLog=${prefix}/var/status.log
NagiosTmp=${prefix}/var/nagios.tmp
NagiosSav=${prefix}/var/status.sav
NagiosCmd=${prefix}/var/rw/nagios.cmd
NagiosVar=${prefix}/var
NagiosRun=${prefix}/var/nagios.lock
NagiosLckDir=/var/lock/subsys
NagiosLckFile=nagios
NagiosCGI=${exec_prefix}/sbin
Nagios=nagios
# Check that nagios exists.
test -f $NagiosBin || exit 0
# Check that nagios.cfg exists.
test -f $NagiosCfg || exit 0
# See how we were called.
case "$1" in
start)
echo "Starting network monitor: nagios"
su -l $Nagios -c "touch $NagiosVar/nagios.log $NagiosSav"
rm -f $NagiosCmd
$NagiosBin -d $NagiosCfg
if [ -d $NagiosLckDir ]; then touch $NagiosLckDir/$NagiosLckFile; fi
sleep 1
status nagios
;;
stop)
echo "Stopping network monitor: nagios"
killproc_nagios nagios
rm -f $NagiosLog $NagiosTmp $NagiosRun $NagiosLckDir/$NagiosLckFile $NagiosCmd
;;
status)
status nagios
;;
restart)
printf "Running configuration check..."
$NagiosBin -v $NagiosCfg > /dev/null 2>&1;
if [ $? -eq 0 ]; then
echo "done"
$0 stop
$0 start
else
$NagiosBin -v $NagiosCfg
echo "failed - aborting restart."
exit 1
fi
;;
reload|force-reload)
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
printf "Running configuration check..."
$NagiosBin -v $NagiosCfg > /dev/null 2>&1;
if [ $? -eq 0 ]; then
echo "done"
if test ! -f $NagiosRun; then
$0 start
else
NagiosPID=`head -n 1 $NagiosRun`
if status nagios > /dev/null; then
printf "Reloading nagios configuration..."
killproc_nagios nagios -HUP
echo "done"
else
$0 stop
$0 start
fi
fi
else
$NagiosBin -v $NagiosCfg
echo "failed - aborting reload."
exit 1
fi
;;
*)
echo "Usage: nagios {start|stop|restart|reload|force-reload|status}"
exit 1
;;
esac
# End of this script
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## APPENDIX C: DOCUMENTATION FOR THE NAGIOS PLUG-INS

### Nagios Plugins Quick-and-Dirty Installation Instructions

---

0) If using the CVS tree, you need m4, automake, and autoconf. To start out, run:

```
# autoconf
# autoheader
# automake
```

1) Run the configure script to initialize variables and create a Makefile, etc.

```
./configure --prefix=BASEDIRECTORY --with-nagios-user=SOMEUSER --withnagios-group=SOMEGRP --with-cgiurl=SOMEURL
```

a) Replace BASEDIRECTORY with the path of the directory under which Nagios is installed (default is '/usr/local/nagios')

b) Replace SOMEUSER with the name of a user on your system that will be assigned permissions to the installed plugins (default is 'nagios')

c) Replace SOMEGRP with the name of a group on your system that will be assigned permissions to the installed plugins (default is 'nagios')

d) Replace CGIURL with the path used to access the Nagios CGIs with a web browser (default is '/cgi-bin/nagios')

2) Compile the plugins with the following command:

```
make all
```

3) Install the compiled plugins and plugin scripts with the following command:

```
make install
```

#### NOTE:

The installation procedure will attempt to place the plugins in a 'libexec/' subdirectory in the base directory you specified with the --prefix argument to the configure script.

4) Verify that your host configuration file (hosts.cfg) for Nagios contains the correct paths to the new plugins.

5) That's it. If you have any problems or questions, feel free send mail to

[nagios-users@lists.sourceforge.net](mailto:nagios-users@lists.sourceforge.net)

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## APPENDIX D: HTTPD.CONF FILE MODIFIED FOR NAGIOS

```
#
# Based upon the NCSA server configuration files originally by Rob McCool.
#
# This is the main Apache server configuration file. It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs-2.0/> for detailed information about
# the directives.
#
# Do NOT simply read the instructions in here without understanding
# what they do. They're here only as hints or reminders. If you are unsure
# consult the online docs. You have been warned.
#
# The configuration directives are grouped into three basic sections:
# 1. Directives that control the operation of the Apache server process as a
# whole (the 'global environment').
# 2. Directives that define the parameters of the 'main' or 'default' server,
# which responds to requests that aren't handled by a virtual host.
# These directives also provide default values for the settings
# of all virtual hosts.
# 3. Settings for virtual hosts, which allow Web requests to be sent to
# different IP addresses or hostnames and have them handled by the
# same Apache server process.
#
# Configuration and logfile names: If the filenames you specify for many
# of the server's control files begin with "/" (or "drive:/" for Win32), the
# server will use that explicit path. If the filenames do *not* begin
# with "/", the value of ServerRoot is prepended -- so "logs/foo.log"
# with ServerRoot set to "/usr/local/apache2" will be interpreted by the
# server as "/usr/local/apache2/logs/foo.log".
#
#### Section 1: Global Environment
#
# The directives in this section affect the overall operation of Apache,
# such as the number of concurrent requests it can handle or where it
# can find its configuration files.
#
#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# NOTE! If you intend to place this on an NFS (or otherwise network)
# mounted filesystem then please read the LockFile documentation
# (available at <URL:http://httpd.apache.org/docs-2.0/mod/core.html#lockfile>);
# you will save yourself a lot of trouble.
#
# Do NOT add a slash at the end of the directory path.
#
ServerRoot "/usr/local/apache2"
#
# The accept serialization lock file MUST BE STORED ON A LOCAL DISK.
#
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
<IfModule !mpm_winnt.c>
<IfModule !mpm_netware.c>
#LockFile logs/accept.lock
</IfModule>
</IfModule>
#
# ScoreBoardFile: File used to store internal server process information.
# If unspecified (the default), the scoreboard will be stored in an
# anonymous shared memory segment, and will be unavailable to third-party
# applications.
# If specified, ensure that no two invocations of Apache share the same
# scoreboard file. The scoreboard file MUST BE STORED ON A LOCAL DISK.
#
<IfModule !mpm_netware.c>
<IfModule !perchild.c>
#ScoreBoardFile logs/apache_runtime_status
</IfModule>
</IfModule>
#
# PidFile: The file in which the server should record its process
# identification number when it starts.
#
<IfModule !mpm_netware.c>
PidFile logs/httpd.pid
</IfModule>
#
# Timeout: The number of seconds before receives and sends time out.
#
Timeout 300
#
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
#
KeepAlive On
#
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
# We recommend you leave this number high, for maximum performance.
#
MaxKeepAliveRequests 100
#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
#
KeepAliveTimeout 15
###
### Server-Pool Size Regulation (MPM specific)
###
# prefork MPM
# StartServers: number of server processes to start
# MinSpareServers: minimum number of server processes which are kept spare
# MaxSpareServers: maximum number of server processes which are kept spare
# MaxClients: maximum number of server processes allowed to start
# MaxRequestsPerChild: maximum number of requests a server process serves
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
<IfModule prefork.c>
StartServers 5
MinSpareServers 5
MaxSpareServers 10
MaxClients 150
MaxRequestsPerChild 0
</IfModule>
# worker MPM
# StartServers: initial number of server processes to start
# MaxClients: maximum number of simultaneous client connections
# MinSpareThreads: minimum number of worker threads which are kept spare
# MaxSpareThreads: maximum number of worker threads which are kept spare
# ThreadsPerChild: constant number of worker threads in each server process
# MaxRequestsPerChild: maximum number of requests a server process serves
<IfModule worker.c>
StartServers 2
MaxClients 150
MinSpareThreads 25
MaxSpareThreads 75
ThreadsPerChild 25
MaxRequestsPerChild 0
</IfModule>
# perchild MPM
# NumServers: constant number of server processes
# StartThreads: initial number of worker threads in each server process
# MinSpareThreads: minimum number of worker threads which are kept spare
# MaxSpareThreads: maximum number of worker threads which are kept spare
# MaxThreadsPerChild: maximum number of worker threads in each server process
# MaxRequestsPerChild: maximum number of connections per server process
<IfModule perchild.c>
NumServers 5
StartThreads 5
MinSpareThreads 5
MaxSpareThreads 10
MaxThreadsPerChild 20
MaxRequestsPerChild 0
</IfModule>
# WinNT MPM
# ThreadsPerChild: constant number of worker threads in the server process
# MaxRequestsPerChild: maximum number of requests a server process serves
<IfModule mpm_winnt.c>
ThreadsPerChild 250
MaxRequestsPerChild 0
</IfModule>
# BeOS MPM
# StartThreads: how many threads do we initially spawn?
# MaxClients: max number of threads we can have (1 thread == 1 client)
# MaxRequestsPerThread: maximum number of requests each thread will process
<IfModule beos.c>
StartThreads 10
MaxClients 50
MaxRequestsPerThread 10000
</IfModule>
# NetWare MPM
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
# ThreadStackSize ..... Stack size allocated for each worker thread
# StartThreads ..... Number of worker threads launched at server startup
# MinSpareThreads ..... Minimum number of idle threads, to handle request spikes
# MaxSpareThreads ..... Maximum number of idle threads
# MaxThreads ..... Maximum number of worker threads alive at the same time
# MaxRequestsPerChild .. Maximum number of requests a thread serves. It is
# recommended that the default value of 0 be set for this
# directive on NetWare. This will allow the thread to
# continue to service requests indefinitely.
<IfModule mpm_netware.c>
ThreadStackSize 65536
StartThreads 250
MinSpareThreads 25
MaxSpareThreads 250
MaxThreads 1000
MaxRequestsPerChild 0
</IfModule>
# OS/2 MPM
# StartServers ..... Number of server processes to maintain
# MinSpareThreads ..... Minimum number of idle threads per process,
# to handle request spikes
# MaxSpareThreads ..... Maximum number of idle threads per process
# MaxRequestsPerChild .. Maximum number of connections per server process
<IfModule mpmt_os2.c>
StartServers 2
MinSpareThreads 5
MaxSpareThreads 10
MaxRequestsPerChild 0
</IfModule>
#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, in addition to the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses (0.0.0.0)
#
#Listen 12.34.56.78:80
Listen 80
#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding 'LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by `httpd -l') do not need
# to be loaded here.
#
# Example:
# LoadModule foo_module modules/mod_foo.so
#
#
# ExtendedStatus controls whether Apache will generate "full" status
# information (ExtendedStatus On) or just basic information (ExtendedStatus
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
# Off) when the "server-status" handler is called. The default is Off.
#
#ExtendedStatus On
#### Section 2: 'Main' server configuration
#
# The directives in this section set up the values used by the 'main'
# server, which responds to any requests that aren't handled by a
# <VirtualHost> definition. These values also provide defaults for
# any <VirtualHost> containers you may define later in the file.
#
# All of these directives may appear inside <VirtualHost> containers,
# in which case these default settings will be overridden for the
# virtual host being defined.
#
<IfModule !mpm_winnt.c>
<IfModule !mpm_netware.c>
#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.
# . On SCO (ODT 3) use "User nouser" and "Group nogroup".
# . On HPUX you may not be able to use shared memory as nobody, and the
# suggested workaround is to create a user www and use that user.
# NOTE that some kernels refuse to setgid(Group) or semctl(IPC_SET)
# when the value of (unsigned)Group is above 60000;
# don't use Group #-1 on these systems!
#
User nobody
Group #-1
</IfModule>
</IfModule>
#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed. This address appears on some server-generated pages, such
# as error documents. e.g. admin@your-domain.com
#
ServerAdmin ahkaplan@partners.org
#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If this is not set to valid DNS name for your host, server-generated
# redirections will not work. See also the UseCanonicalName directive.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
# You will have to access it by its address anyway, and this will make
# redirections work in a sensible way.
#
ServerName ahk:80
#
# UseCanonicalName: Determines how Apache constructs self-referencing
# URLs and the SERVER_NAME and SERVER_PORT variables.
```



# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
# When set "Off", Apache will use the Hostname and Port supplied
# by the client. When set "On", Apache will use the value of the
# ServerName directive.
#
UseCanonicalName Off
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/usr/local/apache2/htdocs"
#
# Each directory to which Apache has access can be configured with respect
# to which services and features are allowed and/or disabled in that
# directory (and its subdirectories).
#
# First, we configure the "default" to be a very restrictive set of
# features.
#
<Directory />
Options FollowSymLinks
AllowOverride None
</Directory>
#
# Note that from this point forward you must specifically allow
# particular features to be enabled - so if something's not working as
# you might expect, make sure that you have specifically enabled it
# below.
#
# This should be changed to whatever you set DocumentRoot to.
#
<Directory "/usr/local/apache2/htdocs">
#
# Possible values for the Options directive are "None", "All",
# or any combination of:
# Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI Multiviews
#
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
# The Options directive is both complicated and important. Please see
# http://httpd.apache.org/docs-2.0/mod/core.html#options
# for more information.
#
Options Indexes FollowSymLinks
#
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
# Options FileInfo AuthConfig Limit
#
AllowOverride None
#
# Controls who can get stuff from this server.
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
#
Order allow,deny
Allow from all
</Directory>
#
# UserDir: The name of the directory that is appended onto a user's home
# directory if a ~user request is received.
#
UserDir public_html
#
# Control access to UserDir directories. The following is an example
# for a site where these directories are restricted to read-only.
#
#<Directory /home/*/public_html>
# AllowOverride FileInfo AuthConfig Limit
# Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
# <Limit GET POST OPTIONS PROPFIND>
# Order allow,deny
# Allow from all
# </Limit>
# <LimitExcept GET POST OPTIONS PROPFIND>
# Order deny,allow
# Deny from all
# </LimitExcept>
#</Directory>
#
# DirectoryIndex: sets the file that Apache will serve if a directory
# is requested.
#
# The index.html.var file (a type-map) is used to deliver content-
# negotiated documents. The MultiViews Option can be used for the
# same purpose, but it is much slower.
#
DirectoryIndex index.html index.html.var
#
# AccessFileName: The name of the file to look for in each directory
# for access control information. See also the AllowOverride directive.
#
AccessFileName .htaccess
#
# The following lines prevent .htaccess and .htpasswd files from being
# viewed by Web clients.
#
<Files ~ "\.ht">
Order allow,deny
Deny from all
</Files>
#
# TypesConfig describes where the mime.types file (or equivalent) is
# to be found.
#
TypesConfig conf/mime.types
#
# DefaultType is the default MIME type the server will use for a document
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
# if it cannot otherwise determine one, such as from filename extensions.
# If your server contains mostly text or HTML documents, "text/plain" is
# a good value. If most of your content is binary, such as applications
# or images, you may want to use "application/octet-stream" instead to
# keep browsers from trying to display binary files as though they are
# text.
#
DefaultType text/plain
#
# The mod_mime_magic module allows the server to use various hints from the
# contents of the file itself to determine its type. The MIMEMagicFile
# directive tells the module where the hint definitions are located.
#
<IfModule mod_mime_magic.c>
MIMEMagicFile conf/magic
</IfModule>
#
# HostnameLookups: Log the names of clients or just their IP addresses
# e.g., www.apache.org (on) or 204.62.129.132 (off).
# The default is off because it'd be overall better for the net if people
# had to knowingly turn this feature on, since enabling it means that
# each client request will result in AT LEAST one lookup request to the
# nameserver.
#
HostnameLookups Off
#
# ErrorLog: The location of the error log file.
# If you do not specify an ErrorLog directive within a <VirtualHost>
# container, error messages relating to that virtual host will be
# logged here. If you *do* define an error logfile for a <VirtualHost>
# container, that host's errors will be logged there and not here.
#
ErrorLog logs/error_log
#
# LogLevel: Control the number of messages logged to the error_log.
# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
#
LogLevel warn
#
# The following directives define some format nicknames for use with
# a CustomLog directive (see below).
#
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
#
# The location and format of the access logfile (Common Logfile Format).
# If you do not define any access logfiles within a <VirtualHost>
# container, they will be logged here. Contrariwise, if you *do*
# define per-<VirtualHost> access logfiles, transactions will be
# logged therein and *not* in this file.
#
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
CustomLog logs/access_log common
#
# If you would like to have agent and referer logfiles, uncomment the
# following directives.
#
#CustomLog logs/referer_log referer
#CustomLog logs/agent_log agent
#
# If you prefer a single logfile with access, agent, and referer information
# (Combined Logfile Format) you can use the following directive.
#
#CustomLog logs/access_log combined
#
# Optionally add a line containing the server version and virtual host
# name to server-generated pages (error documents, FTP directory listings,
# mod_status and mod_info output etc., but not CGI generated documents).
# Set to "EMail" to also include a mailto: link to the ServerAdmin.
# Set to one of: On | Off | EMail
#
ServerSignature On
#
# Aliases: Add here as many aliases as you need (with no limit). The format is
# Alias fakename realname
#
# Note that if you include a trailing / on fakename then the server will
# require it to be present in the URL. So "/icons" isn't aliased in this
# example, only "/icons/". If the fakename is slash-terminated, then the
# realname must also be slash terminated, and if the fakename omits the
# trailing slash, the realname must also omit it.
#
# We include the /icons/ alias for FancyIndexed directory listings. If you
# do not use FancyIndexing, you may comment this out.
#
# Alias /icons/ "/usr/local/apache2/icons/"
# <Directory "/usr/local/apache2/icons">
# Options Indexes MultiViews
# AllowOverride None
# Order allow,deny
# Allow from all
# </Directory>
#
# This should be changed to the ServerRoot/manual/. The alias provides
# the manual, even if you choose to move your DocumentRoot. You may comment
# this out if you do not care for the documentation.
#
Alias /manual "/usr/local/apache2/manual"
<Directory "/usr/local/apache2/manual">
Options Indexes FollowSymLinks MultiViews
AllowOverride None
Order allow,deny
Allow from all
</Directory>
#
# ScriptAlias: This controls which directories contain server scripts.
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
# ScriptAliases are essentially the same as Aliases, except that
# documents in the realname directory are treated as applications and
# run by the server when requested rather than as documents sent to the client.
# The same rules about trailing "/" apply to ScriptAlias directives as to
# Alias.
#
ScriptAlias /cgi-bin/nagios/ "/usr/local/nagios/sbin/"
<IfModule mod_cgid.c>
#
# Additional to mod_cgid.c settings, mod_cgid has Scriptsock <path>
# for setting UNIX socket for communicating with cgid.
#
#Scriptsock logs/cgisock
</IfModule>
#
# "/usr/local/apache2/cgi-bin" should be changed to whatever your ScriptAliased
# CGI directory exists, if you have that configured.
#
# <Directory "/usr/local/apache2/cgi-bin">
# AllowOverride None
# Options None
# Order allow,deny
# Allow from all
# </Directory>
ScriptAlias /nagios/cgi-bin/ /usr/local/nagios/sbin/
<Directory "/usr/local/nagios/sbin">
AllowOverride AuthConfig
Options ExecCGI
Order allow,deny
Allow from all
</Directory>
Alias /nagios /usr/local/nagios/share/
<Directory "/usr/local/nagios/share">
Options None
AllowOverride AuthConfig
Order allow,deny
Allow from all
</Directory>
<Directory /usr/local/nagios/sbin>
AllowOverride AuthConfig
order allow,deny
allow from all
Options ExecCGI
</Directory>
<Directory /usr/local/nagios/share>
AllowOverride AuthConfig
order allow,deny
allow from all
</Directory>
#
# Redirect allows you to tell clients about documents which used to exist in
# your server's namespace, but do not anymore. This allows you to tell the
# clients where to look for the relocated document.
# Example:
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
# Redirect permanent /foo http://www.example.com/bar
#
# Directives controlling the display of server-generated directory listings.
#
#
# FancyIndexing is whether you want fancy directory indexing or standard.
# VersionSort is whether files containing version numbers should be
# compared in the natural way, so that `apache-1.3.9.tar' is placed before
# `apache-1.3.12.tar'.
#
IndexOptions FancyIndexing VersionSort
#
# AddIcon* directives tell the server which icon to show for different
# files or filename extensions. These are only displayed for
# FancyIndexed directories.
#
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip
AddIconByType (TXT,/icons/text.gif) text/*
AddIconByType (IMG,/icons/image2.gif) image/*
AddIconByType (SND,/icons/sound2.gif) audio/*
AddIconByType (VID,/icons/movie.gif) video/*
AddIcon /icons/binary.gif .bin .exe
AddIcon /icons/binhex.gif .hqx
AddIcon /icons/tar.gif .tar
AddIcon /icons/world2.gif .wrl .wrl.gz .vrm .vrm .iv
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
AddIcon /icons/a.gif .ps .ai .eps
AddIcon /icons/layout.gif .html .shtml .htm .pdf
AddIcon /icons/text.gif .txt
AddIcon /icons/c.gif .c
AddIcon /icons/p.gif .pl .py
AddIcon /icons/f.gif .for
AddIcon /icons/dvi.gif .dvi
AddIcon /icons/uuencoded.gif .uu
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
AddIcon /icons/tex.gif .tex
AddIcon /icons/bomb.gif core
AddIcon /icons/back.gif ..
AddIcon /icons/hand.right.gif README
AddIcon /icons/folder.gif ^^DIRECTORY^^
AddIcon /icons/blank.gif ^^BLANKICON^^
#
# DefaultIcon is which icon to show for files which do not have an icon
# explicitly set.
#
DefaultIcon /icons/unknown.gif
#
# AddDescription allows you to place a short description after a file in
# server-generated indexes. These are only displayed for FancyIndexed
# directories.
# Format: AddDescription "description" filename
#
#AddDescription "GZIP compressed document" .gz
#AddDescription "tar archive" .tar
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
#AddDescription "GZIP compressed tar archive" .tgz
#
# ReadmeName is the name of the README file the server will look for by
# default, and append to directory listings.
#
# HeaderName is the name of a file which should be prepended to
# directory indexes.
ReadmeName README.html
HeaderName HEADER.html
#
# IndexIgnore is a set of filenames which directory indexing should ignore
# and not include in the listing. Shell-style wildcarding is permitted.
#
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t
#
# AddEncoding allows you to have certain browsers (Mosaic/X 2.1+) uncompress
# information on the fly. Note: Not all browsers support this.
# Despite the name similarity, the following Add* directives have nothing
# to do with the FancyIndexing customization directives above.
#
AddEncoding x-compress Z
AddEncoding x-gzip gz tgz
#
# DefaultLanguage and AddLanguage allows you to specify the language of
# a document. You can then use content negotiation to give a browser a
# file in a language the user can understand.
#
# Specify a default language. This means that all data
# going out without a specific language tag (see below) will
# be marked with this one. You probably do NOT want to set
# this unless you are sure it is correct for all cases.
#
# * It is generally better to not mark a page as
# * being a certain language than marking it with the wrong
# * language!
#
# DefaultLanguage nl
#
# Note 1: The suffix does not have to be the same as the language
# keyword --- those with documents in Polish (whose net-standard
# language code is pl) may wish to use "AddLanguage pl .po" to
# avoid the ambiguity with the common suffix for perl scripts.
#
# Note 2: The example entries below illustrate that in some cases
# the two character 'Language' abbreviation is not identical to
# the two character 'Country' code for its country,
# E.g. 'Danmark/dk' versus 'Danish/da'.
#
# Note 3: In the case of 'ltz' we violate the RFC by using a three char
# specifier. There is 'work in progress' to fix this and get
# the reference data for rfc1766 cleaned up.
#
# Danish (da) - Dutch (nl) - English (en) - Estonian (et)
# French (fr) - German (de) - Greek-Modern (el)
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
# Italian (it) - Norwegian (no) - Norwegian Nynorsk (nn) - Korean (kr)
# Portugese (pt) - Luxembourggeois* (ltz)
# Spanish (es) - Swedish (sv) - Catalan (ca) - Czech(cz)
# Polish (pl) - Brazilian Portuguese (pt-br) - Japanese (ja)
# Russian (ru) - Croatian (hr)
#
AddLanguage da .dk
AddLanguage nl .nl
AddLanguage en .en
AddLanguage et .et
AddLanguage fr .fr
AddLanguage de .de
AddLanguage he .he
AddLanguage el .el
AddLanguage it .it
AddLanguage ja .ja
AddLanguage pl .po
AddLanguage kr .kr
AddLanguage pt .pt
AddLanguage nn .nn
AddLanguage no .no
AddLanguage pt-br .pt-br
AddLanguage ltz .ltz
AddLanguage ca .ca
AddLanguage es .es
AddLanguage sv .se
AddLanguage cz .cz
AddLanguage ru .ru
AddLanguage tw .tw
AddLanguage zh-tw .tw
AddLanguage hr .hr
#
# LanguagePriority allows you to give precedence to some languages
# in case of a tie during content negotiation.
#
# Just list the languages in decreasing order of preference. We have
# more or less alphabetized them here. You probably want to change this.
#
LanguagePriority en da nl et fr de el it ja kr no pl pt pt-br ltz ca es sv tw
#
# ForceLanguagePriority allows you to serve a result page rather than
# MULTIPLE CHOICES (Prefer) [in case of a tie] or NOT ACCEPTABLE (Fallback)
# [in case no accepted languages matched the available variants]
#
ForceLanguagePriority Prefer Fallback
#
# Specify a default charset for all pages sent out. This is
# always a good idea and opens the door for future internationalisation
# of your web site, should you ever want it. Specifying it as
# a default does little harm; as the standard dictates that a page
# is in iso-8859-1 (latin1) unless specified otherwise i.e. you
# are merely stating the obvious. There are also some security
# reasons in browsers, related to javascript and URL parsing
# which encourage you to always set a default char set.
```



# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
#
AddDefaultCharset ISO-8859-1
#
# Commonly used filename extensions to character sets. You probably
# want to avoid clashes with the language extensions, unless you
# are good at carefully testing your setup after each change.
# See ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets for
# the official list of charset names and their respective RFCs
#
AddCharset ISO-8859-1 .iso8859-1 .latin1
AddCharset ISO-8859-2 .iso8859-2 .latin2 .cen
AddCharset ISO-8859-3 .iso8859-3 .latin3
AddCharset ISO-8859-4 .iso8859-4 .latin4
AddCharset ISO-8859-5 .iso8859-5 .latin5 .cyr .iso-ru
AddCharset ISO-8859-6 .iso8859-6 .latin6 .arb
AddCharset ISO-8859-7 .iso8859-7 .latin7 .grk
AddCharset ISO-8859-8 .iso8859-8 .latin8 .heb
AddCharset ISO-8859-9 .iso8859-9 .latin9 .trk
AddCharset ISO-2022-JP .iso2022-jp .jis
AddCharset ISO-2022-KR .iso2022-kr .kis
AddCharset ISO-2022-CN .iso2022-cn .cis
AddCharset Big5 .Big5 .big5
# For russian, more than one charset is used (depends on client, mostly):
AddCharset WINDOWS-1251 .cp-1251 .win-1251
AddCharset CP866 .cp866
AddCharset KOI8-r .koi8-r .koi8-ru
AddCharset KOI8-ru .koi8-uk .ua
AddCharset ISO-10646-UCS-2 .ucs2
AddCharset ISO-10646-UCS-4 .ucs4
AddCharset UTF-8 .utf8
# The set below does not map to a specific (iso) standard
# but works on a fairly wide range of browsers. Note that
# capitalization actually matters (it should not, but it
# does for some browsers).
#
# See ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets
# for a list of sorts. But browsers support few.
#
AddCharset GB2312 .gb2312 .gb
AddCharset utf-7 .utf7
AddCharset utf-8 .utf8
AddCharset big5 .big5 .b5
AddCharset EUC-TW .euc-tw
AddCharset EUC-JP .euc-jp
AddCharset EUC-KR .euc-kr
AddCharset shift_jis .sjis
#
# AddType allows you to add to or override the MIME configuration
# file mime.types for specific file types.
#
AddType application/x-tar .tgz
#
# AddHandler allows you to map certain file extensions to "handlers":
# actions unrelated to filetype. These can be either built into the server
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
# or added with the Action directive (see below)
#
# To use CGI scripts outside of ScriptAliased directories:
# (You will also need to add "ExecCGI" to the "Options" directive.)
#
#AddHandler cgi-script .cgi
#
# For files that include their own HTTP headers:
#
#AddHandler send-as-is asis
#
# For server-parsed imagemap files:
#
#AddHandler imap-file map
#
# For type maps (negotiated resources):
# (This is enabled by default to allow the Apache "It Worked" page
# to be distributed in multiple languages.)
#
AddHandler type-map var
# Filters allow you to process content before it is sent to the client.
#
# To parse .shtml files for server-side includes (SSI):
# (You will also need to add "Includes" to the "Options" directive.)
#
#AddOutputFilter INCLUDES .shtml
#
# Action lets you define media types that will execute a script whenever
# a matching file is called. This eliminates the need for repeated URL
# pathnames for oft-used CGI file processors.
# Format: Action media/type /cgi-script/location
# Format: Action handler-name /cgi-script/location
#
#
# Customizable error responses come in three flavors:
# 1) plain text 2) local redirects 3) external redirects
#
# Some examples:
#ErrorDocument 500 "The server made a boo boo."
#ErrorDocument 404 /missing.html
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"
#ErrorDocument 402 http://www.example.com/subscription_info.html
#
#
# Putting this all together, we can Internationalize error responses.
#
# We use Alias to redirect any /error/HTTP_<error>.html.var response to
# our collection of by-error message multi-language collections. We use
# includes to substitute the appropriate text.
#
# You can modify the messages' appearance without changing any of the
# default HTTP_<error>.html.var files by adding the line;
#
# Alias /error/include/ "/your/include/path/"
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
#
# which allows you to create your own set of files by starting with the
# /usr/local/apache2/error/include/ files and
# copying them to /your/include/path/, even on a per-VirtualHost basis.
#
<IfModule mod_negotiation.c>
<IfModule mod_include.c>
Alias /error/ "/usr/local/apache2/error/"
<Directory "/usr/local/apache2/error">
AllowOverride None
Options IncludesNoExec
AddOutputFilter Includes html
AddHandler type-map var
Order allow,deny
Allow from all
LanguagePriority en es de fr
ForceLanguagePriority Prefer Fallback
</Directory>
ErrorDocument 400 /error/HTTP_BAD_REQUEST.html.var
ErrorDocument 401 /error/HTTP_UNAUTHORIZED.html.var
ErrorDocument 403 /error/HTTP_FORBIDDEN.html.var
ErrorDocument 404 /error/HTTP_NOT_FOUND.html.var
ErrorDocument 405 /error/HTTP_METHOD_NOT_ALLOWED.html.var
ErrorDocument 408 /error/HTTP_REQUEST_TIME_OUT.html.var
ErrorDocument 410 /error/HTTP_GONE.html.var
ErrorDocument 411 /error/HTTP_LENGTH_REQUIRED.html.var
ErrorDocument 412 /error/HTTP_PRECONDITION_FAILED.html.var
ErrorDocument 413 /error/HTTP_REQUEST_ENTITY_TOO_LARGE.html.var
ErrorDocument 414 /error/HTTP_REQUEST_URI_TOO_LARGE.html.var
ErrorDocument 415 /error/HTTP_SERVICE_UNAVAILABLE.html.var
ErrorDocument 500 /error/HTTP_INTERNAL_SERVER_ERROR.html.var
ErrorDocument 501 /error/HTTP_NOT_IMPLEMENTED.html.var
ErrorDocument 502 /error/HTTP_BAD_GATEWAY.html.var
ErrorDocument 503 /error/HTTP_SERVICE_UNAVAILABLE.html.var
ErrorDocument 506 /error/HTTP_VARIANT_ALSO_VARIES.html.var
</IfModule>
</IfModule>
#
# The following directives modify normal HTTP response behavior to
# handle known problems with browser implementations.
#
BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4.0b2;" nokeepalive downgrade-1.0 force-response-1.0
BrowserMatch "RealPlayer 4.0" force-response-1.0
BrowserMatch "Java/1.0" force-response-1.0
BrowserMatch "JDK/1.0" force-response-1.0
#
# The following directive disables redirects on non-GET requests for
# a directory that does not include the trailing slash. This fixes a
# problem with Microsoft WebFolders which does not appropriately handle
# redirects for folders with DAV methods.
#
BrowserMatch "Microsoft Data Access Internet Publishing Provider" redirect-carefully
BrowserMatch "^WebDrive" redirect-carefully
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
#
# Allow server status reports, with the URL of http://servername/server-status
# Change the ".your-domain.com" to match your domain to enable.
#
#<Location /server-status>
# SetHandler server-status
# Order deny,allow
# Deny from all
# Allow from .your-domain.com
#</Location>
#
# Allow remote server configuration reports, with the URL of
# http://servername/server-info (requires that mod_info.c be loaded).
# Change the ".your-domain.com" to match your domain to enable.
#
#<Location /server-info>
# SetHandler server-info
# Order deny,allow
# Deny from all
# Allow from .your-domain.com
#</Location>
#
# Proxy Server directives. Uncomment the following lines to
# enable the proxy server:
#
#<IfModule mod_proxy.c>
#ProxyRequests On
#
#<Proxy *>
# Order deny,allow
# Deny from all
# Allow from .your-domain.com
#</Proxy>
#
# Enable/disable the handling of HTTP/1.1 "Via:" headers.
# ("Full" adds the server version; "Block" removes all outgoing Via: headers)
# Set to one of: Off | On | Full | Block
#
#ProxyVia On
#
# To enable the cache as well, edit and uncomment the following lines:
# (no cacheing without CacheRoot)
#
#CacheRoot "/usr/local/apache2/proxy"
#CacheSize 5
#CacheGcInterval 4
#CacheMaxExpire 24
#CacheLastModifiedFactor 0.1
#CacheDefaultExpire 1
#NoCache a-domain.com another-domain.edu joes.garage-sale.com
#</IfModule>
# End of proxy directives.
#
# Bring in additional module-specific configurations
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
#
<IfModule mod_ssl.c>
Include conf/ssl.conf
</IfModule>
### Section 3: Virtual Hosts
#
# VirtualHost: If you want to maintain multiple domains/hostnames on your
# machine you can setup VirtualHost containers for them. Most configurations
# use only name-based virtual hosts so the server doesn't need to worry about
# IP addresses. This is indicated by the asterisks in the directives below.
#
# Please see the documentation at
# <URL:http://httpd.apache.org/docs-2.0/vhosts/>
# for further details before you try to setup virtual hosts.
#
# You may use the command line option '-S' to verify your virtual host
# configuration.
#
# Use name-based virtual hosting.
#
#NameVirtualHost *
#
# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost container.
# The first VirtualHost section is used for requests without a known
# server name.
#
#<VirtualHost *>
# ServerAdmin webmaster@dummy-host.example.com
# DocumentRoot /www/docs/dummy-host.example.com
# ServerName dummy-host.example.com
# ErrorLog logs/dummy-host.example.com-error_log
# CustomLog logs/dummy-host.example.com-access_log common
#</VirtualHost>
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## APPENDIX E: HTTPD.CONF FILE ORIGINAL FORMAT

```
#
# Based upon the NCSA server configuration files originally by Rob McCool.
#
# This is the main Apache server configuration file. It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs-2.0/> for detailed information about
# the directives.
#
# Do NOT simply read the instructions in here without understanding
# what they do. They're here only as hints or reminders. If you are unsure
# consult the online docs. You have been warned.
#
# The configuration directives are grouped into three basic sections:
# 1. Directives that control the operation of the Apache server process as a
# whole (the 'global environment').
# 2. Directives that define the parameters of the 'main' or 'default' server,
# which responds to requests that aren't handled by a virtual host.
# These directives also provide default values for the settings
# of all virtual hosts.
# 3. Settings for virtual hosts, which allow Web requests to be sent to
# different IP addresses or hostnames and have them handled by the
# same Apache server process.
#
# Configuration and logfile names: If the filenames you specify for many
# of the server's control files begin with "/" (or "drive:/" for Win32), the
# server will use that explicit path. If the filenames do *not* begin
# with "/", the value of ServerRoot is prepended -- so "logs/foo.log"
# with ServerRoot set to "/usr/local/apache2" will be interpreted by the
# server as "/usr/local/apache2/logs/foo.log".
#
#### Section 1: Global Environment
#
# The directives in this section affect the overall operation of Apache,
# such as the number of concurrent requests it can handle or where it
# can find its configuration files.
#
#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# NOTE! If you intend to place this on an NFS (or otherwise network)
# mounted filesystem then please read the LockFile documentation
# (available at <URL:http://httpd.apache.org/docs-2.0/mod/core.html#lockfile>);
# you will save yourself a lot of trouble.
#
# Do NOT add a slash at the end of the directory path.
#
ServerRoot "/usr/local/apache2"
#
# The accept serialization lock file MUST BE STORED ON A LOCAL DISK.
#
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
<IfModule !mpm_winnt.c>
<IfModule !mpm_netware.c>
#LockFile logs/accept.lock
</IfModule>
</IfModule>
#
# ScoreBoardFile: File used to store internal server process information.
# If unspecified (the default), the scoreboard will be stored in an
# anonymous shared memory segment, and will be unavailable to third-party
# applications.
# If specified, ensure that no two invocations of Apache share the same
# scoreboard file. The scoreboard file MUST BE STORED ON A LOCAL DISK.
#
<IfModule !mpm_netware.c>
<IfModule !perchild.c>
#ScoreBoardFile logs/apache_runtime_status
</IfModule>
</IfModule>
#
# PidFile: The file in which the server should record its process
# identification number when it starts.
#
<IfModule !mpm_netware.c>
PidFile logs/httpd.pid
</IfModule>
#
# Timeout: The number of seconds before receives and sends time out.
#
Timeout 300
#
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
#
KeepAlive On
#
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
# We recommend you leave this number high, for maximum performance.
#
MaxKeepAliveRequests 100
#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
#
KeepAliveTimeout 15
###
### Server-Pool Size Regulation (MPM specific)
###
# prefork MPM
# StartServers: number of server processes to start
# MinSpareServers: minimum number of server processes which are kept spare
# MaxSpareServers: maximum number of server processes which are kept spare
# MaxClients: maximum number of server processes allowed to start
# MaxRequestsPerChild: maximum number of requests a server process serves
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
<IfModule prefork.c>
StartServers 5
MinSpareServers 5
MaxSpareServers 10
MaxClients 150
MaxRequestsPerChild 0
</IfModule>
# worker MPM
# StartServers: initial number of server processes to start
# MaxClients: maximum number of simultaneous client connections
# MinSpareThreads: minimum number of worker threads which are kept spare
# MaxSpareThreads: maximum number of worker threads which are kept spare
# ThreadsPerChild: constant number of worker threads in each server process
# MaxRequestsPerChild: maximum number of requests a server process serves
<IfModule worker.c>
StartServers 2
MaxClients 150
MinSpareThreads 25
MaxSpareThreads 75
ThreadsPerChild 25
MaxRequestsPerChild 0
</IfModule>
# perchild MPM
# NumServers: constant number of server processes
# StartThreads: initial number of worker threads in each server process
# MinSpareThreads: minimum number of worker threads which are kept spare
# MaxSpareThreads: maximum number of worker threads which are kept spare
# MaxThreadsPerChild: maximum number of worker threads in each server process
# MaxRequestsPerChild: maximum number of connections per server process
<IfModule perchild.c>
NumServers 5
StartThreads 5
MinSpareThreads 5
MaxSpareThreads 10
MaxThreadsPerChild 20
MaxRequestsPerChild 0
</IfModule>
# WinNT MPM
# ThreadsPerChild: constant number of worker threads in the server process
# MaxRequestsPerChild: maximum number of requests a server process serves
<IfModule mpm_winnt.c>
ThreadsPerChild 250
MaxRequestsPerChild 0
</IfModule>
# BeOS MPM
# StartThreads: how many threads do we initially spawn?
# MaxClients: max number of threads we can have (1 thread == 1 client)
# MaxRequestsPerThread: maximum number of requests each thread will process
<IfModule beos.c>
StartThreads 10
MaxClients 50
MaxRequestsPerThread 10000
</IfModule>
# NetWare MPM
```



# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
# ThreadStackSize ..... Stack size allocated for each worker thread
# StartThreads ..... Number of worker threads launched at server startup
# MinSpareThreads ..... Minimum number of idle threads, to handle request spikes
# MaxSpareThreads ..... Maximum number of idle threads
# MaxThreads ..... Maximum number of worker threads alive at the same time
# MaxRequestsPerChild .. Maximum number of requests a thread serves. It is
# recommended that the default value of 0 be set for this
# directive on NetWare. This will allow the thread to
# continue to service requests indefinitely.
<IfModule mpm_netware.c>
ThreadStackSize 65536
StartThreads 250
MinSpareThreads 25
MaxSpareThreads 250
MaxThreads 1000
MaxRequestsPerChild 0
</IfModule>
# OS/2 MPM
# StartServers ..... Number of server processes to maintain
# MinSpareThreads ..... Minimum number of idle threads per process,
# to handle request spikes
# MaxSpareThreads ..... Maximum number of idle threads per process
# MaxRequestsPerChild .. Maximum number of connections per server process
<IfModule mpmt_os2.c>
StartServers 2
MinSpareThreads 5
MaxSpareThreads 10
MaxRequestsPerChild 0
</IfModule>
#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, in addition to the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses (0.0.0.0)
#
#Listen 12.34.56.78:80
Listen 80
#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding `LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by `httpd -l') do not need
# to be loaded here.
#
# Example:
# LoadModule foo_module modules/mod_foo.so
#
#
# ExtendedStatus controls whether Apache will generate "full" status
# information (ExtendedStatus On) or just basic information (ExtendedStatus
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
# Off) when the "server-status" handler is called. The default is Off.
#
#ExtendedStatus On
#### Section 2: 'Main' server configuration
#
# The directives in this section set up the values used by the 'main'
# server, which responds to any requests that aren't handled by a
# <VirtualHost> definition. These values also provide defaults for
# any <VirtualHost> containers you may define later in the file.
#
# All of these directives may appear inside <VirtualHost> containers,
# in which case these default settings will be overridden for the
# virtual host being defined.
#
<IfModule !mpm_winnt.c>
<IfModule !mpm_netware.c>
#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.
# . On SCO (ODT 3) use "User nouser" and "Group nogroup".
# . On HPUX you may not be able to use shared memory as nobody, and the
# suggested workaround is to create a user www and use that user.
# NOTE that some kernels refuse to setgid(Group) or semctl(IPC_SET)
# when the value of (unsigned)Group is above 60000;
# don't use Group #-1 on these systems!
#
User nobody
Group #-1
</IfModule>
</IfModule>
#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed. This address appears on some server-generated pages, such
# as error documents. e.g. admin@your-domain.com
#
ServerAdmin you@your.address
#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If this is not set to valid DNS name for your host, server-generated
# redirections will not work. See also the UseCanonicalName directive.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
# You will have to access it by its address anyway, and this will make
# redirections work in a sensible way.
#
#ServerName new.host.name:80
#
# UseCanonicalName: Determines how Apache constructs self-referencing
# URLs and the SERVER_NAME and SERVER_PORT variables.
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
# When set "Off", Apache will use the Hostname and Port supplied
# by the client. When set "On", Apache will use the value of the
# ServerName directive.
#
UseCanonicalName Off
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/usr/local/apache2/htdocs"
#
# Each directory to which Apache has access can be configured with respect
# to which services and features are allowed and/or disabled in that
# directory (and its subdirectories).
#
# First, we configure the "default" to be a very restrictive set of
# features.
#
<Directory />
Options FollowSymLinks
AllowOverride None
</Directory>
#
# Note that from this point forward you must specifically allow
# particular features to be enabled - so if something's not working as
# you might expect, make sure that you have specifically enabled it
# below.
#
# This should be changed to whatever you set DocumentRoot to.
#
<Directory "/usr/local/apache2/htdocs">
#
# Possible values for the Options directive are "None", "All",
# or any combination of:
# Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI Multiviews
#
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
# The Options directive is both complicated and important. Please see
# http://httpd.apache.org/docs-2.0/mod/core.html#options
# for more information.
#
Options Indexes FollowSymLinks
#
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
# Options FileInfo AuthConfig Limit
#
AllowOverride None
#
# Controls who can get stuff from this server.
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
#
Order allow,deny
Allow from all
</Directory>
#
# UserDir: The name of the directory that is appended onto a user's home
# directory if a ~user request is received.
#
UserDir public_html
#
# Control access to UserDir directories. The following is an example
# for a site where these directories are restricted to read-only.
#
#<Directory /home/*/public_html>
# AllowOverride FileInfo AuthConfig Limit
# Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
# <Limit GET POST OPTIONS PROPFIND>
# Order allow,deny
# Allow from all
# </Limit>
# <LimitExcept GET POST OPTIONS PROPFIND>
# Order deny,allow
# Deny from all
# </LimitExcept>
#</Directory>
#
# DirectoryIndex: sets the file that Apache will serve if a directory
# is requested.
#
# The index.html.var file (a type-map) is used to deliver content-
# negotiated documents. The MultiViews Option can be used for the
# same purpose, but it is much slower.
#
DirectoryIndex index.html index.html.var
#
# AccessFileName: The name of the file to look for in each directory
# for access control information. See also the AllowOverride directive.
#
AccessFileName .htaccess
#
# The following lines prevent .htaccess and .htpasswd files from being
# viewed by Web clients.
#
<Files ~ "\.ht">
Order allow,deny
Deny from all
</Files>
#
# TypesConfig describes where the mime.types file (or equivalent) is
# to be found.
#
TypesConfig conf/mime.types
#
# DefaultType is the default MIME type the server will use for a document
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
# if it cannot otherwise determine one, such as from filename extensions.
# If your server contains mostly text or HTML documents, "text/plain" is
# a good value. If most of your content is binary, such as applications
# or images, you may want to use "application/octet-stream" instead to
# keep browsers from trying to display binary files as though they are
# text.
#
DefaultType text/plain
#
# The mod_mime_magic module allows the server to use various hints from the
# contents of the file itself to determine its type. The MIMEMagicFile
# directive tells the module where the hint definitions are located.
#
<IfModule mod_mime_magic.c>
MIMEMagicFile conf/magic
</IfModule>
#
# HostnameLookups: Log the names of clients or just their IP addresses
# e.g., www.apache.org (on) or 204.62.129.132 (off).
# The default is off because it'd be overall better for the net if people
# had to knowingly turn this feature on, since enabling it means that
# each client request will result in AT LEAST one lookup request to the
# nameserver.
#
HostnameLookups Off
#
# ErrorLog: The location of the error log file.
# If you do not specify an ErrorLog directive within a <VirtualHost>
# container, error messages relating to that virtual host will be
# logged here. If you *do* define an error logfile for a <VirtualHost>
# container, that host's errors will be logged there and not here.
#
ErrorLog logs/error_log
#
# LogLevel: Control the number of messages logged to the error_log.
# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
#
LogLevel warn
#
# The following directives define some format nicknames for use with
# a CustomLog directive (see below).
#
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
#
# The location and format of the access logfile (Common Logfile Format).
# If you do not define any access logfiles within a <VirtualHost>
# container, they will be logged here. Contrariwise, if you *do*
# define per-<VirtualHost> access logfiles, transactions will be
# logged therein and *not* in this file.
#
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
CustomLog logs/access_log common
#
# If you would like to have agent and referer logfiles, uncomment the
# following directives.
#
#CustomLog logs/referer_log referer
#CustomLog logs/agent_log agent
#
# If you prefer a single logfile with access, agent, and referer information
# (Combined Logfile Format) you can use the following directive.
#
#CustomLog logs/access_log combined
#
# Optionally add a line containing the server version and virtual host
# name to server-generated pages (error documents, FTP directory listings,
# mod_status and mod_info output etc., but not CGI generated documents).
# Set to "EMail" to also include a mailto: link to the ServerAdmin.
# Set to one of: On | Off | EMail
#
ServerSignature On
#
# Aliases: Add here as many aliases as you need (with no limit). The format is
# Alias fakename realname
#
# Note that if you include a trailing / on fakename then the server will
# require it to be present in the URL. So "/icons" isn't aliased in this
# example, only "/icons/". If the fakename is slash-terminated, then the
# realname must also be slash terminated, and if the fakename omits the
# trailing slash, the realname must also omit it.
#
# We include the /icons/ alias for FancyIndexed directory listings. If you
# do not use FancyIndexing, you may comment this out.
#
Alias /icons/ "/usr/local/apache2/icons/"
<Directory "/usr/local/apache2/icons">
Options Indexes MultiViews
AllowOverride None
Order allow,deny
Allow from all
</Directory>
#
# This should be changed to the ServerRoot/manual/. The alias provides
# the manual, even if you choose to move your DocumentRoot. You may comment
# this out if you do not care for the documentation.
#
Alias /manual "/usr/local/apache2/manual"
<Directory "/usr/local/apache2/manual">
Options Indexes FollowSymLinks MultiViews
AllowOverride None
Order allow,deny
Allow from all
</Directory>
#
# ScriptAlias: This controls which directories contain server scripts.
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
# ScriptAliases are essentially the same as Aliases, except that
# documents in the realname directory are treated as applications and
# run by the server when requested rather than as documents sent to the client.
# The same rules about trailing "/" apply to ScriptAlias directives as to
# Alias.
#
ScriptAlias /cgi-bin/ "/usr/local/apache2/cgi-bin/"
<IfModule mod_cgid.c>
#
# Additional to mod_cgid.c settings, mod_cgid has Scriptsock <path>
# for setting UNIX socket for communicating with cgid.
#
#Scriptsock logs/cgisock
</IfModule>
#
# "/usr/local/apache2/cgi-bin" should be changed to whatever your ScriptAliased
# CGI directory exists, if you have that configured.
#
<Directory "/usr/local/apache2/cgi-bin">
AllowOverride None
Options None
Order allow,deny
Allow from all
</Directory>
#
# Redirect allows you to tell clients about documents which used to exist in
# your server's namespace, but do not anymore. This allows you to tell the
# clients where to look for the relocated document.
# Example:
# Redirect permanent /foo http://www.example.com/bar
#
# Directives controlling the display of server-generated directory listings.
#
#
# FancyIndexing is whether you want fancy directory indexing or standard.
# VersionSort is whether files containing version numbers should be
# compared in the natural way, so that `apache-1.3.9.tar' is placed before
# `apache-1.3.12.tar'.
#
IndexOptions FancyIndexing VersionSort
#
# AddIcon* directives tell the server which icon to show for different
# files or filename extensions. These are only displayed for
# FancyIndexed directories.
#
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip
AddIconByType (TXT,/icons/text.gif) text/*
AddIconByType (IMG,/icons/image2.gif) image/*
AddIconByType (SND,/icons/sound2.gif) audio/*
AddIconByType (VID,/icons/movie.gif) video/*
AddIcon /icons/binary.gif .bin .exe
AddIcon /icons/binhex.gif .hqx
AddIcon /icons/tar.gif .tar
AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
AddIcon /icons/a.gif .ps .ai .eps
AddIcon /icons/layout.gif .html .shtml .htm .pdf
AddIcon /icons/text.gif .txt
AddIcon /icons/c.gif .c
AddIcon /icons/p.gif .pl .py
AddIcon /icons/f.gif .for
AddIcon /icons/dvi.gif .dvi
AddIcon /icons/uuencoded.gif .uu
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
AddIcon /icons/tex.gif .tex
AddIcon /icons/bomb.gif core
AddIcon /icons/back.gif ..
AddIcon /icons/hand.right.gif README
AddIcon /icons/folder.gif ^^DIRECTORY^^
AddIcon /icons/blank.gif ^^BLANKICON^^
#
# DefaultIcon is which icon to show for files which do not have an icon
# explicitly set.
#
DefaultIcon /icons/unknown.gif
#
# AddDescription allows you to place a short description after a file in
# server-generated indexes. These are only displayed for FancyIndexed
# directories.
# Format: AddDescription "description" filename
#
#AddDescription "GZIP compressed document" .gz
#AddDescription "tar archive" .tar
#AddDescription "GZIP compressed tar archive" .tgz
#
# ReadmeName is the name of the README file the server will look for by
# default, and append to directory listings.
#
# HeaderName is the name of a file which should be prepended to
# directory indexes.
ReadmeName README.html
HeaderName HEADER.html
#
# IndexIgnore is a set of filenames which directory indexing should ignore
# and not include in the listing. Shell-style wildcarding is permitted.
#
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t
#
# AddEncoding allows you to have certain browsers (Mosaic/X 2.1+) uncompress
# information on the fly. Note: Not all browsers support this.
# Despite the name similarity, the following Add* directives have nothing
# to do with the FancyIndexing customization directives above.
#
AddEncoding x-compress Z
AddEncoding x-gzip gz tgz
#
# DefaultLanguage and AddLanguage allows you to specify the language of
# a document. You can then use content negotiation to give a browser a
```



# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
# file in a language the user can understand.
#
# Specify a default language. This means that all data
# going out without a specific language tag (see below) will
# be marked with this one. You probably do NOT want to set
# this unless you are sure it is correct for all cases.
#
# * It is generally better to not mark a page as
# * being a certain language than marking it with the wrong
# * language!
#
# DefaultLanguage nl
#
# Note 1: The suffix does not have to be the same as the language
# keyword --- those with documents in Polish (whose net-standard
# language code is pl) may wish to use "AddLanguage pl .po" to
# avoid the ambiguity with the common suffix for perl scripts.
#
# Note 2: The example entries below illustrate that in some cases
# the two character 'Language' abbreviation is not identical to
# the two character 'Country' code for its country,
# E.g. 'Danmark/dk' versus 'Danish/da'.
#
# Note 3: In the case of 'ltz' we violate the RFC by using a three char
# specifier. There is 'work in progress' to fix this and get
# the reference data for rfc1766 cleaned up.
#
# Danish (da) - Dutch (nl) - English (en) - Estonian (et)
# French (fr) - German (de) - Greek-Modern (el)
# Italian (it) - Norwegian (no) - Norwegian Nynorsk (nn) - Korean (kr)
# Portugese (pt) - Luxembourgish* (ltz)
# Spanish (es) - Swedish (sv) - Catalan (ca) - Czech(cz)
# Polish (pl) - Brazilian Portuguese (pt-br) - Japanese (ja)
# Russian (ru) - Croatian (hr)
#
AddLanguage da .dk
AddLanguage nl .nl
AddLanguage en .en
AddLanguage et .et
AddLanguage fr .fr
AddLanguage de .de
AddLanguage he .he
AddLanguage el .el
AddLanguage it .it
AddLanguage ja .ja
AddLanguage pl .po
AddLanguage kr .kr
AddLanguage pt .pt
AddLanguage nn .nn
AddLanguage no .no
AddLanguage pt-br .pt-br
AddLanguage ltz .ltz
AddLanguage ca .ca
AddLanguage es .es
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
AddLanguage sv .se
AddLanguage cz .cz
AddLanguage ru .ru
AddLanguage tw .tw
AddLanguage zh-tw .tw
AddLanguage hr .hr
#
# LanguagePriority allows you to give precedence to some languages
# in case of a tie during content negotiation.
#
# Just list the languages in decreasing order of preference. We have
# more or less alphabetized them here. You probably want to change this.
#
LanguagePriority en da nl et fr de el it ja kr no pl pt pt-br ltz ca es sv tw
#
# ForceLanguagePriority allows you to serve a result page rather than
# MULTIPLE CHOICES (Prefer) [in case of a tie] or NOT ACCEPTABLE (Fallback)
# [in case no accepted languages matched the available variants]
#
ForceLanguagePriority Prefer Fallback
#
# Specify a default charset for all pages sent out. This is
# always a good idea and opens the door for future internationalisation
# of your web site, should you ever want it. Specifying it as
# a default does little harm; as the standard dictates that a page
# is in iso-8859-1 (latin1) unless specified otherwise i.e. you
# are merely stating the obvious. There are also some security
# reasons in browsers, related to javascript and URL parsing
# which encourage you to always set a default char set.
#
AddDefaultCharset ISO-8859-1
#
# Commonly used filename extensions to character sets. You probably
# want to avoid clashes with the language extensions, unless you
# are good at carefully testing your setup after each change.
# See ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets for
# the official list of charset names and their respective RFCs
#
AddCharset ISO-8859-1 .iso8859-1 .latin1
AddCharset ISO-8859-2 .iso8859-2 .latin2 .cen
AddCharset ISO-8859-3 .iso8859-3 .latin3
AddCharset ISO-8859-4 .iso8859-4 .latin4
AddCharset ISO-8859-5 .iso8859-5 .latin5 .cyr .iso-ru
AddCharset ISO-8859-6 .iso8859-6 .latin6 .arb
AddCharset ISO-8859-7 .iso8859-7 .latin7 .grk
AddCharset ISO-8859-8 .iso8859-8 .latin8 .heb
AddCharset ISO-8859-9 .iso8859-9 .latin9 .trk
AddCharset ISO-2022-JP .iso2022-jp .jis
AddCharset ISO-2022-KR .iso2022-kr .kis
AddCharset ISO-2022-CN .iso2022-cn .cis
AddCharset Big5 .Big5 .big5
# For russian, more than one charset is used (depends on client, mostly):
AddCharset WINDOWS-1251 .cp-1251 .win-1251
AddCharset CP866 .cp866
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
AddCharset KOI8-r .koi8-r .koi8-ru
AddCharset KOI8-ru .koi8-uk .ua
AddCharset ISO-10646-UCS-2 .ucs2
AddCharset ISO-10646-UCS-4 .ucs4
AddCharset UTF-8 .utf8
# The set below does not map to a specific (iso) standard
# but works on a fairly wide range of browsers. Note that
# capitalization actually matters (it should not, but it
# does for some browsers).
#
# See ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets
# for a list of sorts. But browsers support few.
#
AddCharset GB2312 .gb2312 .gb
AddCharset utf-7 .utf7
AddCharset utf-8 .utf8
AddCharset big5 .big5 .b5
AddCharset EUC-TW .euc-tw
AddCharset EUC-JP .euc-jp
AddCharset EUC-KR .euc-kr
AddCharset shift_jis .sjis
#
# AddType allows you to add to or override the MIME configuration
# file mime.types for specific file types.
#
AddType application/x-tar .tgz
#
# AddHandler allows you to map certain file extensions to "handlers":
# actions unrelated to filetype. These can be either built into the server
# or added with the Action directive (see below)
#
# To use CGI scripts outside of ScriptAliased directories:
# (You will also need to add "ExecCGI" to the "Options" directive.)
#
#AddHandler cgi-script .cgi
#
# For files that include their own HTTP headers:
#
#AddHandler send-as-is asis
#
# For server-parsed imagemap files:
#
#AddHandler imap-file map
#
# For type maps (negotiated resources):
# (This is enabled by default to allow the Apache "It Worked" page
# to be distributed in multiple languages.)
#
AddHandler type-map var
# Filters allow you to process content before it is sent to the client.
#
# To parse .shtml files for server-side includes (SSI):
# (You will also need to add "Includes" to the "Options" directive.)
#
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
#AddOutputFilter INCLUDES .shtml
#
# Action lets you define media types that will execute a script whenever
# a matching file is called. This eliminates the need for repeated URL
# pathnames for oft-used CGI file processors.
# Format: Action media/type /cgi-script/location
# Format: Action handler-name /cgi-script/location
#
#
# Customizable error responses come in three flavors:
# 1) plain text 2) local redirects 3) external redirects
#
# Some examples:
#ErrorDocument 500 "The server made a boo boo."
#ErrorDocument 404 /missing.html
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"
#ErrorDocument 402 http://www.example.com/subscription_info.html
#
#
# Putting this all together, we can Internationalize error responses.
#
# We use Alias to redirect any /error/HTTP_<error>.html.var response to
# our collection of by-error message multi-language collections. We use
# includes to substitute the appropriate text.
#
# You can modify the messages' appearance without changing any of the
# default HTTP_<error>.html.var files by adding the line;
#
# Alias /error/include/ "/your/include/path/"
#
# which allows you to create your own set of files by starting with the
# /usr/local/apache2/error/include/ files and
# copying them to /your/include/path/, even on a per-VirtualHost basis.
#
<IfModule mod_negotiation.c>
<IfModule mod_include.c>
Alias /error/ "/usr/local/apache2/error/"
<Directory "/usr/local/apache2/error">
AllowOverride None
Options IncludesNoExec
AddOutputFilter Includes html
AddHandler type-map var
Order allow,deny
Allow from all
LanguagePriority en es de fr
ForceLanguagePriority Prefer Fallback
</Directory>
ErrorDocument 400 /error/HTTP_BAD_REQUEST.html.var
ErrorDocument 401 /error/HTTP_UNAUTHORIZED.html.var
ErrorDocument 403 /error/HTTP_FORBIDDEN.html.var
ErrorDocument 404 /error/HTTP_NOT_FOUND.html.var
ErrorDocument 405 /error/HTTP_METHOD_NOT_ALLOWED.html.var
ErrorDocument 408 /error/HTTP_REQUEST_TIME_OUT.html.var
ErrorDocument 410 /error/HTTP_GONE.html.var
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
ErrorDocument 411 /error/HTTP_LENGTH_REQUIRED.html.var
ErrorDocument 412 /error/HTTP_PRECONDITION_FAILED.html.var
ErrorDocument 413 /error/HTTP_REQUEST_ENTITY_TOO_LARGE.html.var
ErrorDocument 414 /error/HTTP_REQUEST_URI_TOO_LARGE.html.var
ErrorDocument 415 /error/HTTP_SERVICE_UNAVAILABLE.html.var
ErrorDocument 500 /error/HTTP_INTERNAL_SERVER_ERROR.html.var
ErrorDocument 501 /error/HTTP_NOT_IMPLEMENTED.html.var
ErrorDocument 502 /error/HTTP_BAD_GATEWAY.html.var
ErrorDocument 503 /error/HTTP_SERVICE_UNAVAILABLE.html.var
ErrorDocument 506 /error/HTTP_VARIANT_ALSO_VARIES.html.var
</IfModule>
</IfModule>
#
# The following directives modify normal HTTP response behavior to
# handle known problems with browser implementations.
#
BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4\..0b2;" nokeepalive downgrade-1.0 force-response-1.0
BrowserMatch "RealPlayer 4\..0" force-response-1.0
BrowserMatch "Java/1\..0" force-response-1.0
BrowserMatch "JDK/1\..0" force-response-1.0
#
# The following directive disables redirects on non-GET requests for
# a directory that does not include the trailing slash. This fixes a
# problem with Microsoft WebFolders which does not appropriately handle
# redirects for folders with DAV methods.
#
BrowserMatch "Microsoft Data Access Internet Publishing Provider" redirect-carefully
BrowserMatch "^WebDrive" redirect-carefully
#
# Allow server status reports, with the URL of http://servername/server-status
# Change the ".your-domain.com" to match your domain to enable.
#
#<Location /server-status>
# SetHandler server-status
# Order deny,allow
# Deny from all
# Allow from .your-domain.com
#</Location>
#
# Allow remote server configuration reports, with the URL of
# http://servername/server-info (requires that mod_info.c be loaded).
# Change the ".your-domain.com" to match your domain to enable.
#
#<Location /server-info>
# SetHandler server-info
# Order deny,allow
# Deny from all
# Allow from .your-domain.com
#</Location>
#
# Proxy Server directives. Uncomment the following lines to
# enable the proxy server:
#
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
#<IfModule mod_proxy.c>
#ProxyRequests On
#
#<Proxy *>
# Order deny,allow
# Deny from all
# Allow from .your-domain.com
#</Proxy>
#
# Enable/disable the handling of HTTP/1.1 "Via:" headers.
# ("Full" adds the server version; "Block" removes all outgoing Via: headers)
# Set to one of: Off | On | Full | Block
#
#ProxyVia On
#
# To enable the cache as well, edit and uncomment the following lines:
# (no cacheing without CacheRoot)
#
#CacheRoot "/usr/local/apache2/proxy"
#CacheSize 5
#CacheGcInterval 4
#CacheMaxExpire 24
#CacheLastModifiedFactor 0.1
#CacheDefaultExpire 1
#NoCache a-domain.com another-domain.edu joes.garage-sale.com
#</IfModule>
# End of proxy directives.
#
# Bring in additional module-specific configurations
#
<IfModule mod_ssl.c>
Include conf/ssl.conf
</IfModule>
#### Section 3: Virtual Hosts
#
# VirtualHost: If you want to maintain multiple domains/hostnames on your
# machine you can setup VirtualHost containers for them. Most configurations
# use only name-based virtual hosts so the server doesn't need to worry about
# IP addresses. This is indicated by the asterisks in the directives below.
#
# Please see the documentation at
# <URL:http://httpd.apache.org/docs-2.0/vhosts/>
# for further details before you try to setup virtual hosts.
#
# You may use the command line option '-S' to verify your virtual host
# configuration.
#
# Use name-based virtual hosting.
#
#NameVirtualHost *
#
# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost container.
# The first VirtualHost section is used for requests without a known
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
# server name.  
#  
#<VirtualHost *>  
# ServerAdmin webmaster@dummy-host.example.com  
# DocumentRoot /www/docs/dummy-host.example.com  
# ServerName dummy-host.example.com  
# ErrorLog logs/dummy-host.example.com-error_log  
# CustomLog logs/dummy-host.example.com-access_log common  
#</VirtualHost>
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## APPENDIX F: .HTACCESS FILE USED WITH NAGIOS

```
AuthName "Nagios Access"  
AuthType Basic  
AuthUserFile /usr/local/nagios/etc/htpasswd.users  
require valid-user
```



# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## APPENDIX G: MODIFIED AND ORIGINAL CGI.CFG FILES USED WITH NAGIOS

### Modified CGI.CFG File:

```
#####  
#  
# CGI.CFG - Sample CGI Configuration File for Nagios 1.0b5  
#  
# Last Modified: 05-30-2002  
#  
#####  
# MAIN CONFIGURATION FILE  
# This tells the CGIs where to find your main configuration file.  
# The CGIs will read the main and host config files for any other  
# data they might need.  
  
main_config_file=/usr/local/nagios/etc/nagios.cfg  
  
# PHYSICAL HTML PATH  
# This is the path where the HTML files for Nagios reside. This  
# value is used to locate the logo images needed by the statusmap  
# and statuswrl CGIs.  
  
physical_html_path=/usr/local/nagios/share  
  
# URL HTML PATH  
# This is the path portion of the URL that corresponds to the  
# physical location of the Nagios HTML files (as defined above).  
# This value is used by the CGIs to locate the online documentation  
# and graphics. If you access the Nagios pages with an URL like  
# http://www.myhost.com/nagios, this value should be '/nagios'  
# (without the quotes).  
  
url_html_path=/nagios/  
  
# CONTEXT-SENSITIVE HELP  
# This option determines whether or not a context-sensitive  
# help icon will be displayed for most of the CGIs.  
# Values: 0 = disables context-sensitive help  
# 1 = enables context-sensitive help  
  
show_context_help=0  
  
# NAGIOS PROCESS CHECK COMMAND  
# This is the full path and filename of the program used to check  
# the status of the Nagios process. It is used only by the CGIs  
# and is completely optional. However, if you don't use it, you'll  
# see warning messages in the CGIs about the Nagios process  
# not running and you won't be able to execute any commands from  
# the web interface. The program should follow the same rules  
# as plugins; the return codes are the same as for the plugins,  
# it should have timeout protection, it should output something  
# to STDIO, etc.  
#
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
# Note: If you are using the check_nagios plugin here, the first
# argument should be the physical path to the status log, the
# second argument is the number of minutes that the status log
# contents should be "fresher" than, and the third argument is the
# string that should be matched from the output of the 'ps'
# command in order to locate the running Nagios process. That
# process string is going to vary depending on how you start
# Nagios. Run the 'ps' command manually to see what the command
# line entry for the Nagios process looks like.
#nagios_check_command=/usr/local/nagios/libexec/check_nagios /usr/local/nagios/var/status.log
5 /usr/local/nagios/bin/nagios'
# AUTHENTICATION USAGE
# This option controls whether or not the CGIs will use any
# authentication when displaying host and service information, as
# well as committing commands to Nagios for processing.
#
# Read the HTML documentation to learn how the authorization works!
#
# NOTE: It is a really *bad* idea to disable authorization, unless
# you plan on removing the command CGI (cmd.cgi)! Failure to do
# so will leave you wide open to kiddies messing with Nagios and
# possibly hitting you with a denial of service attack by filling up
# your drive by continuously writing to your command file!
#
# Setting this value to 0 will cause the CGIs to *not* use
# authentication (bad idea), while any other value will make them
# use the authentication functions (the default).

use_authentication=1

# DEFAULT USER
# Setting this variable will define a default user name that can
# access pages without authentication. This allows people within a
# secure domain (i.e., behind a firewall) to see the current status
# without authenticating. You may want to use this to avoid basic
# authentication if you are not using a secure server since basic
# authentication transmits passwords in the clear.
#
# Important: Do not define a default username unless you are
# running a secure web server and are sure that everyone who has
# access to the CGIs has been authenticated in some manner! If you
# define this variable, anyone who has not authenticated to the web
# server will inherit all rights you assign to this user!
#default_user_name=guest
# SYSTEM/PROCESS INFORMATION ACCESS
# This option is a comma-delimited list of all usernames that
# have access to viewing the Nagios process information as
# provided by the Extended Information CGI (extinfo.cgi). By
# default, *no one* has access to this unless you choose to
# not use authorization. You may use an asterisk (*) to
# authorize any user who has authenticated to the web server.

authorized_for_system_information=nagiosadmin,ahk
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## # CONFIGURATION INFORMATION ACCESS

# This option is a comma-delimited list of all usernames that  
# can view ALL configuration information (hosts, commands, etc).  
# By default, users can only view configuration information  
# for the hosts and services they are contacts for. You may use  
# an asterisk (\*) to authorize any user who has authenticated  
# to the web server.

authorized\_for\_configuration\_information=nagiosadmin,ahk,jdoe

## # SYSTEM/PROCESS COMMAND ACCESS

# This option is a comma-delimited list of all usernames that  
# can issue shutdown and restart commands to Nagios via the  
# command CGI (cmd.cgi). Users in this list can also change  
# the program mode to active or standby. By default, \*no one\*  
# has access to this unless you choose to not use authorization.  
# You may use an asterisk (\*) to authorize any user who has  
# authenticated to the web server.

authorized\_for\_system\_commands=nagiosadmin,ahk

## # GLOBAL HOST/SERVICE VIEW ACCESS

# These two options are comma-delimited lists of all usernames that  
# can view information for all hosts and services that are being  
# monitored. By default, users can only view information  
# for hosts or services that they are contacts for (unless you  
# you choose to not use authorization). You may use an asterisk (\*)  
# to authorize any user who has authenticated to the web server.

authorized\_for\_all\_services=nagiosadmin,ahk,guest

authorized\_for\_all\_hosts=nagiosadmin,ahk,guest

## # GLOBAL HOST/SERVICE COMMAND ACCESS

# These two options are comma-delimited lists of all usernames that  
# can issue host or service related commands via the command  
# CGI (cmd.cgi) for all hosts and services that are being monitored.  
# By default, users can only issue commands for hosts or services  
# that they are contacts for (unless you you choose to not use  
# authorization). You may use an asterisk (\*) to authorize any  
# user who has authenticated to the web server.

authorized\_for\_all\_service\_commands=nagiosadmin,ahk

authorized\_for\_all\_host\_commands=nagiosadmin,ahk

## # EXTENDED HOST INFORMATION

# This is all entirely optional. If you don't enter any extended  
# information, nothing bad will happen - I promise... Its basically  
# just used to have pretty icons and such associated with your hosts.  
# This is especially nice when you're using the statusmap and  
# statuswrl CGIs. You can also specify an URL that links to a document  
# containing more information about the host (location details, contact  
# information, etc).  
#

# hostextinfo[<host\_name>]=<notes\_url>;<icon\_image>;<vrml\_image>;<gd2\_image>;\

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
# <image_alt>;<x_2d>,<y_2d>;<x_3d>,<y_3d>,<z_3d>;  
#  
# <notes_url> = Optional URL that points to a document of  
# some type containing information on the host.  
# The information (and the document type) can  
# be anything you want. Examples include details  
# on the physical location of the server, info  
# on how to contact the admins in case of an  
# emergency, etc. Relative URLs start in the  
# same path that is used to access the CGIs.  
# The link that is created for the host's notes  
# notes is found in the extinfo CGI.  
# Note: You may use the $HOSTNAME$ and  
# $HOSTADDRESS$ macros in this URL.  
# <icon_image> = A GIF, PNG, or JPG image to associate with  
# the host. This is used in the status and  
# extinfo CGIs.  
# <vrml_image> = An image to use in the statuswrl CGI in the  
# VRML generation. Transparent images don't  
# work so great..  
# <gd2_image> = A GD2 format image used by the statusmap CGI  
# to represent the host. PNG images can be  
# converted to GD2 format by using the 'pngtogd2'  
# utility supplied with Boutell's gd library.  
# <image_alt> = ALT tag used with PNG, GIF, and GD2 images  
# in the status, statusmap, and extinfo CGIs  
# <x_2d>,<y_2d> = X and Y coordinates used when drawing the  
# host in the statusmap CGI. (0,0) is located  
# in the upper left corner of the screen and is  
# considered to be the origin. The coordinates  
# you supply here are used as the coords of the  
# upper left hand corner of host icon. Both  
# numbers should be positive integers.  
# <x_3d>,<y_3d>,<z_3d> = X, Y, and Z coordinates used when drawing  
# the host in the statuswrl (VRML) CGI. All  
# numbers can be positive or negative (anywhere  
# in 3-D space). The coordinates are used to  
# determine the center of the host "cube" that  
# is drawn. Host "cubes" are drawn with a  
# height, width, and depth of 0.5 (meters).  
#  
# Note: All images must be placed in the /logos subdirectory under  
# the HTML images path (i.e. /usr/local/nagios/share/images/logos/).  
# This path is automatically determined by appending "/images/logos"  
# to the path specified by the 'physical_html_path' directive.  
#hostextinfo[es-ed]=/serverinfo/es-ed.html;novell40.gif;novell40.jpg;novell40.gd2;IntranetWare  
4.11;100,50;3.5,0.0,-1.5;  
#hostextinfo[rosie]=/serverinfo/rosie.html;win40.gif;win40.jpg;win40.gd2;NT Server 4.0;;  
# EXTENDED SERVICE INFORMATION  
# This is all entirely optional. If you don't enter any extended  
# information, nothing bad will happen - I promise... Its basically  
# just used to have pretty icons and such associated with your services.  
# You can also specify an URL that links to a document containing more  
# information about the service (location details, contact information,
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
# etc).
#
# serviceextinfo[<host_name>;<svc_description>]=<notes_url>;<icon_image>;<image_alt>
#
# <notes_url> = Optional URL that points to a document of
# some type containing information on the service.
# The information (and the document type) can
# be anything you want. Examples include details
# on the physical location of the server, info
# on how to contact the admins in case of an
# emergency, etc. Relative URLs start in the
# same path that is used to access the CGIs.
# The link that is created for the service's
# notes URL is found in the extinfo CGI.
# Note: You may use the $HOSTNAME$, $HOSTADDRESS$,
# and $SERVICEDESC$ macros in this URL.
# <icon_image> = A GIF, PNG, or JPG image to associate with
# the service. This is used in the status and
# extinfo CGIs.
# <image_alt> = ALT tag used with image
#
# Note: All images must be placed in the /logos subdirectory under
# the HTML images path (i.e. /usr/local/nagios/share/images/logos/).
# This path is automatically determined by appending "/images/logos"
# to the path specified by the 'physical_html_path' directive.
#serviceextinfo[eseds;
PING]=http://www.somewhere.com?tracerouteto=$HOSTADDRESS$;;PING rate
#serviceextinfo[rosie;Security Alerts]=;security.gif;Security alerts
# STATUSMAP BACKGROUND IMAGE
# This option allows you to specify an image to be used as a
# background in the statusmap CGI. It is assumed that the image
# resides in the HTML images path (i.e. /usr/local/nagios/share/images).
# This path is automatically determined by appending "/images"
# to the path specified by the 'physical_html_path' directive.
# Note: The image file must be in GD2 format!
#statusmap_background_image=smbbackground.gd2
# DEFAULT STATUSMAP LAYOUT METHOD
# This option allows you to specify the default layout method
# the statusmap CGI should use for drawing hosts. If you do
# not use this option, the default is to use user-defined
# coordinates. Valid options are as follows:
# 0 = User-defined coordinates
# 1 = Depth layers
# 2 = Collapsed tree
# 3 = Balanced tree
# 4 = Circular
# 5 = Circular (Marked Up)

default_statusmap_layout=5

# DEFAULT STATUSWRL LAYOUT METHOD
# This option allows you to specify the default layout method
# the statuswrl (VRML) CGI should use for drawing hosts. If you
# do not use this option, the default is to use user-defined
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

# coordinates. Valid options are as follows:

# 0 = User-defined coordinates

# 2 = Collapsed tree

# 3 = Balanced tree

# 4 = Circular

default\_statuswrl\_layout=4

# STATUSWRL INCLUDE

# This option allows you to include your own objects in the

# generated VRML world. It is assumed that the file

# resides in the HTML path (i.e. /usr/local/nagios/share).

#statuswrl\_include=myworld.wrl

# PING SYNTAX

# This option determines what syntax should be used when

# attempting to ping a host from the WAP interface (using

# the statuswml CGI. You must include the full path to

# the ping binary, along with all required options. The

# \$HOSTADDRESS\$ macro is substituted with the address of

# the host before the command is executed.

ping\_syntax=/bin/ping -n -U -c 5 \$HOSTADDRESS\$

# REFRESH RATE

# This option allows you to specify the refresh rate in seconds

# of various CGIs (status, statusmap, extinfo, and outages).

refresh\_rate=90

# SOUND OPTIONS

# These options allow you to specify an optional audio file

# that should be played in your browser window when there are

# problems on the network. The audio files are used only in

# the status CGI. Only the sound for the most critical problem

# will be played. Order of importance (higher to lower) is as

# follows: unreachable hosts, down hosts, critical services,

# warning services, and unknown services. If there are no

# visible problems, the sound file optionally specified by

# 'normal\_sound' variable will be played.

#

#

# <varname>=<sound\_file>

#

# Note: All audio files must be placed in the /media subdirectory

# under the HTML path (i.e. /usr/local/nagios/share/media/).

#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

# DG EXTENDED DATA

# Note: These config directives are only used if you compiled

# in database support for extended data!

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
# The user you specify here only needs SELECT privileges on the
# 'hostextinfo' table in the database.
#xeddb_host=somehost
#xeddb_port=someport
#xeddb_database=somedatabase
#xeddb_username=someuser
#xeddb_password=somepassword
# DB STATUS DATA (Read-Only For CGIs)
# Note: These config directives are only used if you compiled
# in database support for status data!
# The user you specify here only needs SELECT privileges on the
# 'programstatus', 'hoststatus', and 'servicestatus' tables
# in the database, as these values are only used by the CGIs.
# The core program will read the directives you specify in
# in a resource file.
#xsddb_host=somehost
#xsddb_port=someport
#xsddb_database=somedatabase
#xsddb_username=someuser
#xsddb_password=somepassword
# DB COMMENT DATA (Read-Only For CGIs)
# Note: These config directives are only used if you compiled
# in database support for comment data!
# The user you specify here only needs SELECT privileges on the
# 'hostcomments', and 'servicecomments' tables in the database,
# as these values are only used by the CGIs. The core program
# will read the directives you specify in a resource file.
#xcddb_host=somehost
#xcddb_port=someport
#xcddb_database=somedatabase
#xcddb_username=someuser
#xcddb_password=somepassword
# DB DOWNTIME DATA (Read-Only For CGIs)
# Note: These config directives are only used if you compiled
# in database support for downtime data!
# The user you specify here only needs SELECT privileges on the
# 'hostdowntime', and 'servicedowntime' tables in the database,
# as these values are only used by the CGIs. The core program
# will read the directives you specify in a resource file.
#xdddb_host=somehost
#xdddb_port=someport
#xdddb_database=somedatabase
#xdddb_username=someuser
#xdddb_password=somepassword
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## Original CGI.CFG File:

```
#####  
#  
# CGI.CFG - Sample CGI Configuration File for Nagios 1.0b5  
#  
# Last Modified: 05-30-2002  
#  
#####  
# MAIN CONFIGURATION FILE  
# This tells the CGIs where to find your main configuration file.  
# The CGIs will read the main and host config files for any other  
# data they might need.  
  
main_config_file=/usr/local/nagios/etc/nagios.cfg  
  
# PHYSICAL HTML PATH  
# This is the path where the HTML files for Nagios reside. This  
# value is used to locate the logo images needed by the statusmap  
# and statuswrl CGIs.  
  
physical_html_path=/usr/local/nagios/share  
  
# URL HTML PATH  
# This is the path portion of the URL that corresponds to the  
# physical location of the Nagios HTML files (as defined above).  
# This value is used by the CGIs to locate the online documentation  
# and graphics. If you access the Nagios pages with an URL like  
# http://www.myhost.com/nagios, this value should be '/nagios'  
# (without the quotes).  
  
url_html_path=/nagios/  
  
# CONTEXT-SENSITIVE HELP  
# This option determines whether or not a context-sensitive  
# help icon will be displayed for most of the CGIs.  
# Values: 0 = disables context-sensitive help  
# 1 = enables context-sensitive help  
  
show_context_help=0  
  
# NAGIOS PROCESS CHECK COMMAND  
# This is the full path and filename of the program used to check  
# the status of the Nagios process. It is used only by the CGIs  
# and is completely optional. However, if you don't use it, you'll  
# see warning messages in the CGIs about the Nagios process  
# not running and you won't be able to execute any commands from  
# the web interface. The program should follow the same rules  
# as plugins; the return codes are the same as for the plugins,  
# it should have timeout protection, it should output something  
# to STDIO, etc.  
#  
# Note: If you are using the check_nagios plugin here, the first  
# argument should be the physical path to the status log, the  
# second argument is the number of minutes that the status log
```



# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
# contents should be "fresher" than, and the third argument is the
# string that should be matched from the output of the 'ps'
# command in order to locate the running Nagios process. That
# process string is going to vary depending on how you start
# Nagios. Run the 'ps' command manually to see what the command
# line entry for the Nagios process looks like.
#nagios_check_command=/usr/local/nagios/libexec/check_nagios
```

```
/usr/local/nagios/var/status.log 5 '/usr/local/nagios/bin/nagios'
```

## # AUTHENTICATION USAGE

```
# This option controls whether or not the CGIs will use any
# authentication when displaying host and service information, as
# well as committing commands to Nagios for processing.
#
# Read the HTML documentation to learn how the authorization works!
#
# NOTE: It is a really *bad* idea to disable authorization, unless
# you plan on removing the command CGI (cmd.cgi)! Failure to do
# so will leave you wide open to kiddies messing with Nagios and
# possibly hitting you with a denial of service attack by filling up
# your drive by continuously writing to your command file!
#
# Setting this value to 0 will cause the CGIs to *not* use
# authentication (bad idea), while any other value will make them
# use the authentication functions (the default).
```

```
use_authentication=1
```

## # DEFAULT USER

```
# Setting this variable will define a default user name that can
# access pages without authentication. This allows people within a
# secure domain (i.e., behind a firewall) to see the current status
# without authenticating. You may want to use this to avoid basic
# authentication if you are not using a secure server since basic
# authentication transmits passwords in the clear.
#
# Important: Do not define a default username unless you are
# running a secure web server and are sure that everyone who has
# access to the CGIs has been authenticated in some manner! If you
# define this variable, anyone who has not authenticated to the web
# server will inherit all rights you assign to this user!
```

```
#default_user_name=guest
```

## # SYSTEM/PROCESS INFORMATION ACCESS

```
# This option is a comma-delimited list of all usernames that
# have access to viewing the Nagios process information as
# provided by the Extended Information CGI (extinfo.cgi). By
# default, *no one* has access to this unless you choose to
# not use authorization. You may use an asterisk (*) to
# authorize any user who has authenticated to the web server.
#authorized_for_system_information=nagiosadmin,theboss,jdoe
```

## # CONFIGURATION INFORMATION ACCESS

```
# This option is a comma-delimited list of all usernames that
# can view ALL configuration information (hosts, commands, etc).
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
# By default, users can only view configuration information
# for the hosts and services they are contacts for. You may use
# an asterisk (*) to authorize any user who has authenticated
# to the web server.
#authorized_for_configuration_information=nagiosadmin,jdoe
# SYSTEM/PROCESS COMMAND ACCESS
# This option is a comma-delimited list of all usernames that
# can issue shutdown and restart commands to Nagios via the
# command CGI (cmd.cgi). Users in this list can also change
# the program mode to active or standby. By default, *no one*
# has access to this unless you choose to not use authorization.
# You may use an asterisk (*) to authorize any user who has
# authenticated to the web server.
#authorized_for_system_commands=nagiosadmin
# GLOBAL HOST/SERVICE VIEW ACCESS
# These two options are comma-delimited lists of all usernames that
# can view information for all hosts and services that are being
# monitored. By default, users can only view information
# for hosts or services that they are contacts for (unless you
# you choose to not use authorization). You may use an asterisk (*)
# to authorize any user who has authenticated to the web server.
#authorized_for_all_services=nagiosadmin,guest
#authorized_for_all_hosts=nagiosadmin,guest
# GLOBAL HOST/SERVICE COMMAND ACCESS
# These two options are comma-delimited lists of all usernames that
# can issue host or service related commands via the command
# CGI (cmd.cgi) for all hosts and services that are being monitored.
# By default, users can only issue commands for hosts or services
# that they are contacts for (unless you you choose to not use
# authorization). You may use an asterisk (*) to authorize any
# user who has authenticated to the web server.
#authorized_for_all_service_commands=nagiosadmin
#authorized_for_all_host_commands=nagiosadmin
# EXTENDED HOST INFORMATION
# This is all entirely optional. If you don't enter any extended
# information, nothing bad will happen - I promise... Its basically
# just used to have pretty icons and such associated with your hosts.
# This is especially nice when you're using the statusmap and
# statuswrl CGIs. You can also specify an URL that links to a document
# containing more information about the host (location details, contact
# information, etc).
#
# hostextinfo[<host_name>]=<notes_url>;<icon_image>;<vml_image>;<gd2_image>;\
# <image_alt>;<x_2d>;<y_2d>;<x_3d>;<y_3d>;<z_3d>;
#
# <notes_url> = Optional URL that points to a document of
# some type containing information on the host.
# The information (and the document type) can
# be anything you want. Examples include details
# on the physical location of the server, info
# on how to contact the admins in case of an
# emergency, etc. Relative URLs start in the
# same path that is used to access the CGIs.
# The link that is created for the host's notes
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
# notes is found in the extinfo CGI.
# Note: You may use the $HOSTNAME$ and
# $HOSTADDRESS$ macros in this URL.
# <icon_image> = A GIF, PNG, or JPG image to associate with
# the host. This is used in the status and
# extinfo CGIs.
# <vrml_image> = An image to use in the statuswrl CGI in the
# VRML generation. Transparent images don't
# work so great..
# <gd2_image> = A GD2 format image used by the statusmap CGI
# to represent the host. PNG images can be
# converted to GD2 format by using the 'pngtogd2'
# utility supplied with Boutell's gd library.
# <image_alt> = ALT tag used with PNG, GIF, and GD2 images
# in the status, statusmap, and extinfo CGIs
# <x_2d>, <y_2d> = X and Y coordinates used when drawing the
# host in the statusmap CGI. (0,0) is located
# in the upper left corner of the screen and is
# considered to be the origin. The coordinates
# you supply here are used as the coords of the
# upper left hand corner of host icon. Both
# numbers should be positive integers.
# <x_3d>, <y_3d>, <z_3d> = X, Y, and Z coordinates used when drawing
# the host in the statuswrl (VRML) CGI. All
# numbers can be positive or negative (anywhere
# in 3-D space). The coordinates are used to
# determine the center of the host "cube" that
# is drawn. Host "cubes" are drawn with a
# height, width, and depth of 0.5 (meters).
#
# Note: All images must be placed in the /logos subdirectory under
# the HTML images path (i.e. /usr/local/nagios/share/images/logos/).
# This path is automatically determined by appending "/images/logos"
# to the path specified by the 'physical_html_path' directive.
#hostextinfo[es-eds]=/serverinfo/eseds.
html;novell40.gif;novell40.jpg;novell40.gd2;IntranetWare 4.11;100,50;3.5,0.0,-1.5;
#hostextinfo[rosie]=/serverinfo/rosie.html;win40.gif;win40.jpg;win40.gd2;NT Server 4.0;;;
# EXTENDED SERVICE INFORMATION
# This is all entirely optional. If you don't enter any extended
# information, nothing bad will happen - I promise... Its basically
# just used to have pretty icons and such associated with your services.
# You can also specify an URL that links to a document containing more
# information about the service (location details, contact information,
# etc).
#
#
serviceextinfo[<host_name>;<svc_description>]=<notes_url>;<icon_image>;<image_alt
>
#
# <notes_url> = Optional URL that points to a document of
# some type containing information on the service.
# The information (and the document type) can
# be anything you want. Examples include details
# on the physical location of the server, info
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
# on how to contact the admins in case of an
# emergency, etc. Relative URLs start in the
# same path that is used to access the CGIs.
# The link that is created for the service's
# notes URL is found in the extinfo CGI.
# Note: You may use the $HOSTNAME$, $HOSTADDRESS$,
# and $SERVICEDESC$ macros in this URL.
# <icon_image> = A GIF, PNG, or JPG image to associate with
# the service. This is used in the status and
# extinfo CGIs.
# <image_alt> = ALT tag used with image
#
# Note: All images must be placed in the /logos subdirectory under
# the HTML images path (i.e. /usr/local/nagios/share/images/logos/).
# This path is automatically determined by appending "/images/logos"
# to the path specified by the 'physical_html_path' directive.
#serviceextinfo[eseds;
PING]=http://www.somewhere.com?tracerouteto=$HOSTADDRESS$;;PING rate
#serviceextinfo[rosie;Security Alerts]=;security.gif;Security alerts
# STATUSMAP BACKGROUND IMAGE
# This option allows you to specify an image to be used as a
# background in the statusmap CGI. It is assumed that the image
# resides in the HTML images path (i.e. /usr/local/nagios/share/images).
# This path is automatically determined by appending "/images"
# to the path specified by the 'physical_html_path' directive.
# Note: The image file must be in GD2 format!
#statusmap_background_image=smbbackground.gd2
# DEFAULT STATUSMAP LAYOUT METHOD
# This option allows you to specify the default layout method
# the statusmap CGI should use for drawing hosts. If you do
# not use this option, the default is to use user-defined
# coordinates. Valid options are as follows:
# 0 = User-defined coordinates
# 1 = Depth layers
# 2 = Collapsed tree
# 3 = Balanced tree
# 4 = Circular
# 5 = Circular (Marked Up)

default_statusmap_layout=5

# DEFAULT STATUSWRL LAYOUT METHOD
# This option allows you to specify the default layout method
# the statuswrl (VRML) CGI should use for drawing hosts. If you
# do not use this option, the default is to use user-defined
# coordinates. Valid options are as follows:
# 0 = User-defined coordinates
# 2 = Collapsed tree
# 3 = Balanced tree
# 4 = Circular

default_statuswrl_layout=4
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
# STATUSWRL INCLUDE
# This option allows you to include your own objects in the
# generated VRML world. It is assumed that the file
# resides in the HTML path (i.e. /usr/local/nagios/share).
#statuswrl_include=myworld.wrl
# PING SYNTAX
# This option determines what syntax should be used when
# attempting to ping a host from the WAP interface (using
# the statuswml CGI. You must include the full path to
# the ping binary, along with all required options. The
# $HOSTADDRESS$ macro is substituted with the address of
# the host before the command is executed.

ping_syntax=/bin/ping -n -U -c 5 $HOSTADDRESS$

# REFRESH RATE
# This option allows you to specify the refresh rate in seconds
# of various CGIs (status, statusmap, extinfo, and outages).

refresh_rate=90

# SOUND OPTIONS
# These options allow you to specify an optional audio file
# that should be played in your browser window when there are
# problems on the network. The audio files are used only in
# the status CGI. Only the sound for the most critical problem
# will be played. Order of importance (higher to lower) is as
# follows: unreachable hosts, down hosts, critical services,
# warning services, and unknown services. If there are no
# visible problems, the sound file optionally specified by
# 'normal_sound' variable will be played.
#
#
# <varname>=<sound_file>
#
# Note: All audio files must be placed in the /media subdirectory
# under the HTML path (i.e. /usr/local/nagios/share/media/).
#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
# DG EXTENDED DATA
# Note: These config directives are only used if you compiled
# in database support for extended data!
# The user you specify here only needs SELECT privileges on the
# 'hostextinfo' table in the database.
#xeddb_host=somehost
#xeddb_port=someport
#xeddb_database=somedatabase
#xeddb_username=someuser
#xeddb_password=somepassword
# DB STATUS DATA (Read-Only For CGIs)
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
# Note: These config directives are only used if you compiled
# in database support for status data!
# The user you specify here only needs SELECT privileges on the
# 'programstatus', 'hoststatus', and 'servicestatus' tables
# in the database, as these values are only used by the CGIs.
# The core program will read the directives you specify in
# in a resource file.
#xsddb_host=somehost
#xsddb_port=someport
#xsddb_database=somedatabase
#xsddb_username=someuser
#xsddb_password=somepassword
# DB COMMENT DATA (Read-Only For CGIs)
# Note: These config directives are only used if you compiled
# in database support for comment data!
# The user you specify here only needs SELECT privileges on the
# 'hostcomments', and 'servicecomments' tables in the database,
# as these values are only used by the CGIs. The core program
# will read the directives you specify in a resource file.
#xcddb_host=somehost
#xcddb_port=someport
#xcddb_database=somedatabase
#xcddb_username=someuser
#xcddb_password=somepassword
# DB DOWNTIME DATA (Read-Only For CGIs)
# Note: These config directives are only used if you compiled
# in database support for downtime data!
# The user you specify here only needs SELECT privileges on the
# 'hostdowntime', and 'servicedowntime' tables in the database,
# as these values are only used by the CGIs. The core program
# will read the directives you specify in a resource file.
#xdddb_host=somehost
#xdddb_port=someport
#xdddb_database=somedatabase
#xdddb_username=someuser
#xdddb_password=somepassword
```

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## APPENDIX H – NRPE README

\*\*\*\*\*

### NRPE README

\*\*\*\*\*

#### Purpose

-----

The purpose of this addon is to allow you to execute Nagios plugins on a remote host in as transparent a manner as possible.

#### Contents

-----

There are two pieces to this addon:

1) NRPE -This program runs as a background process on the remote host and processes command execution requests from the check\_nrpe plugin on the Nagios host. Upon receiving a plugin request from an authorized host, it will execute the command line associated with the command name it received and send the program output and return code back to the check\_nrpe plugin.

2) check\_nrpe - This is a plugin that is run on the Nagios host and is used to contact the NRPE process on remote hosts. The plugin requests that a plugin be executed on the remote host and wait for the NRPE process to execute the plugin and return the result. The plugin then uses the output and return code from the plugin execution on the remote host for its own output and return code.

#### Compiling

-----

The code is very basic and may not work on your particular system without some tweaking. I just haven't put a lot of effort into this addond. Most Linux users should be able to compile NRPE and the check\_nrpe plugin with the following commands...

```
./configure  
make all
```

The binaries will be located in the src/ directory after you run 'make all' and will have to be installed manually somewhere on your system.

NOTE: Since the check\_nrpe plugin and nrpe daemon run on different machines (the plugin runs on the Nagios host and the daemon runs on the remote host), you will have to compile the nrpe daemon on the target machine.

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## Installing

-----  
The check\_nrpe plugin should be placed on the Nagios host along with your other plugins. In most cases, this will be in the /usr/local/nagios/libexec directory. The nrpe program and the configuration file (nrpe.cfg) should be placed somewhere on the remote host. Note that you will also have to install some plugins on the remote host if you want to make much use of this addon.

## Running Under INETD or XINETD

-----  
If you plan on running nrpe under inetd or xinetd and making use of TCP wrappers, you need to do the following things:

1) Add a line to your /etc/services file as follows (modify the port number as you see fit) nrpe 5666/tcp # NRPE

2) Add entries for the NRPE daemon to either your inetd or xinetd configuration files. Which one your use will depend on which superserver is installed on your system. Both methods are described below. NOTE: If you run nrpe under inetd or xinetd, the server\_port and allowed\_hosts variables in the nrpe configuration file are ignored.

### \*\*\*\*\* INETD \*\*\*\*\*

If your system uses the inetd superserver, add an entry to /etc/inetd.conf as follows:

```
nrpe stream tcp nowait <user> /usr/sbin/tcpd <nrpebin> -i <nrpecfg>
```

- Replace <user> with the name of the user that the nrpe server should run as.

Example: nagios

- Replace <nrpebin> with the path to the nrpe binary on your system.

Example: /usr/local/nagios/nrpe

- Replace <nrpecfg> with the path to the nrpe config file on your system.

Example: /usr/local/nagios/nrpe.cfg



# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

\*\*\*\*\* XINETD \*\*\*\*\*

If your system uses xinetd instead of inetd, you'll probably want to create a file called 'nrpe' in your /etc/xinetd.d directory that contains the following entries:

```
# default: on
# description: NRPE
service nrpe
{
    flags = REUSE
    socket_type = stream
    wait = no
    user = <user>
    server = <nrpebin>
    server_args = -i <nrpecfg>
    log_on_failure += USERID
    disable = no
    only_from = <ipaddress1> <ipaddress2> ...
}
```

- Replace <user> with the name of the user that the nrpe server should run as.
- Replace <nrpebin> with the path to the nrpe binary on your system.
- Replace <nrpecfg> with the path to the nrpe config file on your system.
- Replace the <ipaddress> fields with the IP addresses of hosts which are allowed to connect to the NRPE daemon. This only works if xinetd was compiled with support for wrappers.

3) Restart inetd or xinetd with the following command (pick the one that is appropriate for your system):

```
/etc/rc.d/init.d/inet restart
/etc/rc.d/init.d/xinetd restart
```

4) Add entries to your /etc/hosts.allow and /etc/hosts.deny file to enable TCP wrapper protection for the nrpe service. This is optional, although highly recommended.

## Configuring Things On The Nagios Host

---

Examples for configuring the nrpe daemon are found in the sample nrpe.cfg file included in this distribution. That config file resides on the remote host(s) along with the nrpe daemon. The check\_nrpe plugin gets installed on the Nagios host. In order to use the check\_nrpe plugin from within Nagios, you'll have to define a few things in the host config file. An example command definition for the check\_nrpe plugin would look like this:

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

```
define command{
command_name check_nrpe
command_line /usr/local/nagios/libexec/check_nrpe $HOSTADDRESS$ -c $ARG1$
}
```

In any service definitions that use the nrpe plugin/daemon to get their results, you would set the service check command portion of the definition to something like this (sample service definition is simplified for this example):

```
define service{
host_name someremotehost
service_description someremoteservice
check_command check_nrpe!yourcommand
... etc ...
}
```

where "yourcommand" is a name of a command that you define in your nrpe.cfg file on the remote host (see the docs in the sample nrpe.cfg file for more information).

Questions?

-----

If you have questions about this addon, or problems getting things working, first try searching the nagios-users mailing list archives. Details on searching the list archives can be found at <http://www.nagios.org> If all else fails, you can email me and I'll try and respond as soon as I get a chance.

- Ethan Galstad ([nagios@nagios.org](mailto:nagios@nagios.org))

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## APPENDIX I – Installation Procedure for Net-SNMP

### TABLE OF CONTENTS

=====

Table Of Contents  
Quick Instructions  
\* Ucd-Snmp Specific Information  
Long (but you should read these) Instructions  
Installing the Perl/SNMP Module  
\* Compilers and Options  
Compiling For Multiple Architectures  
Installation Names  
Optional Features  
Sharing Defaults  
Operation Controls  
\* = required reading

### QUICK INSTRUCTIONS

=====

- 1) Run ./configure  
(type "./configure --help" for a quick usage summary.)  
(--prefix=PATH will change the default /usr/local installation path.) (see "Compilers and Options" on changing the compiler to use)
- 2) Optionally edit config.h  
(due to prompting done by the configure script, this is rarely necessary.)
- 3) make
- 4) Run the next two commands as root:
- 5) umask 022 # <-- zero-two-two
- 6) make install

Note: By default, everything will be installed in /usr/local. (see below for more instructions)

### UCD-SNMP Specific Information

=====

As of V3.3.1 the configuration files are now looked for in \$(prefix)/share/snmp, where (\$prefix) is defined as the value passed to the --prefix argument of the configure script, or /usr/local if undefined. In version 3.0.3 till 3.3, the files were kept in \$(prefix)/lib/snmp Optional features to pass to configure for ucd-snmp can be obtained by running configure --help.

### LONG (but you should read these) INSTRUCTIONS

=====

The `configure' shell script attempts to guess correct values for various system-dependent variables used during compilation. It uses those values to create a `Makefile' in each directory of the package. It may also

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

create one or more `.h` files containing system-dependent definitions. Finally, it creates a shell script `config.status` that you can run in the future to recreate the current configuration, a file `config.cache` that saves the results of its tests to speed up reconfiguring, and a file `config.log` containing compiler output (useful mainly for debugging `configure`). If you need to do unusual things to compile the package, please try to figure out how `configure` could check whether to do them, and mail diffs or instructions to the address given in the `README` so they can be considered for the next release. If at some point `config.cache` contains results you don't want to keep, you may remove or edit it. The file `configure.in` is used to create `configure` by a program called `autoconf`. You only need `configure.in` if you want to change it or regenerate `configure` using a newer version of `autoconf`. The simplest way to compile this package is:

1. `cd` to the directory containing the package's source code and type `./configure` to configure the package for your system. If you're using `csh` on an old version of System V, you might need to type `sh ./configure` instead to prevent `csh` from trying to execute `configure` itself. Running `configure` takes awhile. While running, it prints some messages telling which features it is checking for.
2. Edit `config.h`. IMPORTANT information is in here. Like the location of the log file and the location of the configuration files. Again, you should really edit `config.h.in` instead and then run `config.status`, which will generate `config.h` from `config.h.in`.
3. Type `make` to compile the package.
4. Type `make install` to install the programs and any data files and documentation.
5. You can remove the program binaries and object files from the source code directory by typing `make clean`. To also remove the files that `configure` created (so you can compile the package for a different kind of computer), type `make distclean`.

## Installing the Perl/SNMP Module

=====

The Perl/SNMP Module is now bundled with the `ucd-snmp` package and is located in the `ucd-snmp/perl/SNMP` directory. This module is an interface to the `ucd-snmp` SNMP client side management API. The Perl package provides a high level abstract interface to these functions for rapid application development. Following are the steps required to build and install this perl module.

- 1.) Be sure to read the `ucd-snmp/perl/SNMP/README`
- 2.) Be sure that `ucd-snmp` has been successfully tested and installed – the headers and libraries will be required to build, the applications and mibs will be required to test the perl module

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

3.) cd to ucd-snmp/perl/SNMP and run  
perl Makefile.PL  
make  
make test  
make install (as root)

## Compilers and Options

=====

Some systems require unusual options for compilation or linking that the `configure` script does not know about. You can give `configure` initial values for variables by setting them in the environment. Using a Bournecompatible shell, you can do that on the command line like this:

```
CC=c89 CFLAGS=-O2 LIBS=-lposix ./configure
```

Or on systems that have the `env` program, you can do it like this: `env CPPFLAGS=-I/usr/local/include LDFLAGS=-s ./configure`

## Compiling For Multiple Architectures

=====

You can compile the package for more than one kind of computer at the same time, by placing the object files for each architecture in their own directory. To do this, you must use a version of `make` that supports the `VPATH` variable, such as GNU `make`. `cd` to the directory where you want the object files and executables to go and run the `configure` script. `configure` automatically checks for the source code in the directory that `configure` is in and in `..`. If you have to use a `make` that does not supports the `VPATH` variable, there is a `maketarget` script that will generate a symlink'ed shadow-directory for the object files. Do a `sh maketarget`, then `cd` into `targets/config.guess` and do the configuration and installation.`

## Installation Names

=====

By default, `make install` will install the package's files in `/usr/local/bin`, `/usr/local/man`, etc. You can specify an installation prefix other than `/usr/local` by giving `configure` the option `--prefix=PATH`. You can specify separate installation prefixes for architecture-specific files and architecture-independent files. If you give `configure` the option `--exec-prefix=PATH`, the package will use PATH as the prefix for installing programs and libraries. Documentation and other data files will still use the regular prefix. If the package supports it, you can cause programs to be installed with an extra prefix or suffix on their names by giving `configure` the option `--program-prefix=PREFIX` or `--program-suffix=SUFFIX`.

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## Optional Features

=====

Some packages pay attention to '--enable-FEATURE' options to 'configure', where FEATURE indicates an optional part of the package. They may also pay attention to '--with-PACKAGE' options, where PACKAGE is something like 'gnuas' or 'x' (for the X Window System). The 'README' should mention any '-enable-' and '--with-' options that the package recognizes. For packages that use the X Window System, 'configure' can usually find the X include and library files automatically, but if it doesn't, you can use the 'configure' options '--x-includes=DIR' and '--x-libraries=DIR' to specify their locations.

## Specifying the System Type

=====

There may be some features 'configure' can not figure out automatically, but needs to determine by the type of host the package will run on. Usually 'configure' can figure that out, but if it prints a message saying it can not guess the host type, give it the '--host=TYPE' option. TYPE can either be a short name for the system type, such as 'sun4', or a canonical name with three fields: CPU-COMPANY-SYSTEM See the file 'config.sub' for the possible values of each field. If 'config.sub' isn't included in this package, then this package doesn't need to know the host type. If you are building compiler tools for cross-compiling, you can also use the '--target=TYPE' option to select the type of system they will produce code for and the '--build=TYPE' option to select the type of system on which you are compiling the package.

## Sharing Defaults

=====

If you want to set default values for 'configure' scripts to share, you can create a site shell script called 'config.site' that gives default values for variables like 'CC', 'cache\_file', and 'prefix'. 'configure' looks for 'PREFIX/share/config.site' if it exists, then 'PREFIX/etc/config.site' if it exists. Or, you can set the 'CONFIG\_SITE' environment variable to the location of the site script. A warning: not all 'configure' scripts look for a site script.

## Operation Controls

=====

'configure' recognizes the following options to control how it operates.

'--cache-file=FILE'

Use and save the results of the tests in FILE instead of './config.cache'. Set FILE to '/dev/null' to disable caching, for debugging 'configure'.

'--help'

Print a summary of the options to 'configure', and exit.

'--quiet'

'--silent'

'-q'

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

Do not print messages saying which checks are being made.

`--srcdir=DIR`

Look for the package's source code in directory DIR. Usually `configure` can determine that directory automatically.

`--version`

Print the version of Autoconf used to generate the `configure` script, and exit.

`configure` also accepts some other, not widely useful, options.

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## APPENDIX J – NET::SNMP DOCUMENTATION

Net::SNMP, version 4.0.3

### NAME

Net::SNMP - Object oriented interface to SNMP

### DESCRIPTION

The Net::SNMP module implements an object oriented interface to the Simple Network Management Protocol. Perl applications can use the module to retrieve or update information on a remote host using the SNMP protocol. The module supports SNMP version-1, SNMP version-2c (Community-Based SNMPv2), and SNMP version-3. The Net::SNMP module assumes that the user has a basic understanding of the Simple Network Management Protocol and related network management concepts.

### INSTALLATION

To install the Net::SNMP module and all of its dependencies directly from the Comprehensive Perl Archive Network (CPAN) execute the command:

```
perl -MCPAN -e "install Net::SNMP"
```

The Net::SNMP module can also be installed using the distribution file downloaded from CPAN. After unpacking the file, perform one of the following steps while in the top level directory of the distribution:

- a. Create a makefile by running Perl against Makefile.PL and then run make:

```
perl Makefile.PL  
make test  
make install
```

- b. Copy or move the entire directory structure (including files) located under the lib/Net directory in the distribution into a directory named Net in a Perl library directory.

### REQUIREMENTS

Net::SNMP uses syntax that is not supported in versions of Perl earlier than v5.6.0. The non-core modules Crypt::DES, Digest::MD5, Digest::SHA1, and Digest::HMAC are needed to support SNMPv3.

### DOCUMENTATION

Documentation is included as part of the Net::SNMP module in Plain Old Documentation (POD) format.



# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## AUTHOR

David M. Town [dtown@cpan.org](mailto:dtown@cpan.org)

## COPYRIGHT

Copyright (c) 1998-2002 David M. Town. All rights reserved. This program is free software; you may redistribute it and/or modify it under the same terms as Perl itself.

RCS \$Id: README,v 4.5 2002/09/09 12:46:09 dtown Exp \$

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

## APPENDIX K – KEYCHAIN APPLICATION SETUP DOCUMENTATION

Usage: `./keychain [ options ] sshkey ...`

Description:

Keychain is an OpenSSH key manager, typically run from `.bash_profile`. When run, it will make sure `ssh-agent` is running; if not, it will start `ssh-agent`. It will redirect `ssh-agent`'s output to `~/.keychain/[hostname]-sh`, so that cron jobs that need to use `ssh-agent` keys can simply source this file and make the necessary passwordless `ssh` connections. In addition, when `keychain` runs, it will check with `ssh-agent` and make sure that the `ssh` RSA/DSA keys that you specified on the `keychain` command line have actually been added to `ssh-agent`. If not, you are prompted for the appropriate passphrases so that they can be added by `keychain`.

Typically, one uses `keychain` by adding the following to the top of their `.bash_profile` (or `~/.zlogin`, in case of `zsh`):

```
keychain ~/.ssh/id_rsa ~.ssh/id_dsa.keychain/${HOSTNAME}-sh
```

```
# alt. syntax: . ~/.keychain/`uname -n`-sh  
# note the use of back-quotes (`) rather than single quotes (') above.  
# We now include the hostname (`uname -n`) in the keychain filename  
# for NFS-compatibility.
```

You can make `keychain` work with your `csh`-compatible shell by adding the following to your `.cshrc`:

```
keychain ~/.ssh/id_rsa ~/.ssh/id_dsa  
source ~/.keychain/${HOSTNAME}-csh
```

`Keychain` allows all your apps and cron jobs to use a single `ssh-agent` process as an authentication agent. By default, the `ssh-agent` started by `keychain` is long-running and will continue to run, even after you have logged out from the system. If you'd like to tighten up security a bit, take a look at the `--clear` option, described below.

Options:

`--clear`

Tells `keychain` to delete all of `ssh-agent`'s host keys. Typically, This is used in the `~/.bash_profile`. The theory behind this is that `keychain` should assume that

# PROCEDURE FOR THE INSTALLATION OF THE NAGIOS NETWORK MONITORING PROGRAM

you are an intruder until proven otherwise. However, while this option increases security, it still allows your cron jobs to use your ssh keys when you're logged out.

--dir [directoryname]

Keychain will look in [directoryname] for the .keychain file, rather than your home directory.

--noask

This option tells keychain do everything it normally does (ensure ssh-agent is running, set up the .keychain/[hostname]-{c}sh files) except that it will not prompt you to add any of the keys you specified if they haven't yet been added to ssh-agent.

--nocolor

This option disables color highlighting for non vt-100 compatible terms.

--stop | -k

This option tells keychain to stop all running ssh-agent processes, and then exit.

--quiet | -q

This option tells keychain to turn off verbose mode and only print error messages and interactive messages. This is useful for login scripts etc.