

# NMIS 8 Installation Guide (up to Version 8.5.4G)

**Attention:** This document covers installations up to and including NMIS Version 8.5.4G and **will not be updated in the future**.

For NMIS versions 8.5.6G and newer the installation process was simplified substantially, and there is an [updated version of the installation guide for 8.5.6G](#).

- [Introduction](#)
  - [Caveats](#)
  - [Assumptions](#)
- [Installation](#)
  - [Linux Distribution](#)
    - [Linux Security Settings](#)
  - [Additional Packages](#)
    - [CentOS Additional Packages:](#)
    - [Debian/Ubuntu Additional Packages:](#)
  - [RRDTool Installation](#)
    - [Debian/Ubuntu:](#)
    - [CentOS/Red Hat:](#)
    - [Compiling your own RRDtool:](#)
  - [NMIS8 Installation](#)
    - [Automatic Installation](#)
    - [Manual Installation](#)
  - [Perl CPAN Module Installation](#)
    - [Unix File Permissions](#)
    - [Initial NMIS Test](#)
  - [Initial NMIS Troubleshooting](#)
    - [RRDTool Libraries not Found](#)
    - [NMIS Config Test](#)
  - [NMIS Web Server Setup](#)
    - [CentOS and other Red Hat derivatives:](#)
    - [Debian/Ubuntu:](#)
    - [Auto Start HTTPD Server](#)
    - [Debian/Ubuntu Users](#)
    - [Start HTTPD Server](#)
    - [CentOS/Red Hat:](#)
    - [Debian/Ubuntu:](#)
  - [Ensure Security settings will allow you to access the web server](#)
    - [IPTables](#)
    - [SELinux](#)
  - [Quick Testing](#)
  - [NMIS Crontab Configuration](#)
- [SNMPD, Net-SNMP and Testing your Configuration](#)
  - [Debian/Ubuntu](#)
  - [Auto Start SNMP Daemons](#)
  - [Start SNMPD Manually](#)
  - [Run a Test Update](#)
  - [Run a Test Collection](#)

## Introduction

This document will describe how to install NMIS8 on GNU/Linux distros. This document is based on Terrell Prude Jr's document for the installation of NMIS 4.3.x, v1.1 April 1, 2010.

## Caveats

NMIS8 has been widely deployed and tested on many Linux distros, with Red Hat/CentOS being the most popular. This document contains variations specific to several Linux distros, but not all of them have been tested equally heavily. Please provide feedback and submit modifications or changes to [support@opmantek.com](mailto:support@opmantek.com).

## Assumptions

The reader is assumed to have a working knowledge of UNIX or GNU/Linux systems, how to compile software, and what Perl is. While NMIS is written in Perl, you do not need to be a "Perl hacker" to install this program and make it work.

Also, this document will use the <nmisdir> tag to refer to NMIS's installation directory. The default location is /usr/local/nmis8.

# Installation

## Linux Distribution

First, install your GNU/Linux distribution. With both CentOS 5 and Debian, I do a basic installation with no extra stuff. That means no GNOME, KDE, "Server" packages, or anything else. Just a base installation is all that's necessary. This guide has been checked on a CentOS minimal install which is indeed quite minimal, because of this you may see some packages being installed that you might normally assume to be installed.

## Linux Security Settings

If you're running CentOS or any other Red Hat-derived distribution, be sure that SELinux is turned off, Permissive mode was tested and it worked well, just very nagging. So disabling SELinux is a lot easier that way.

Also, I like to turn off the default Red Hat firewall. It is assumed that the reader knows how to do both of these; if not, there are plenty of resources on the Web describing how.

## Additional Packages

Now that you have your system set up, there are some things that we need to install to make NMIS work. Some of these are required for the compilation of certain Perl modules and of RRDtool. These packages will come from your distribution's repositories.

### CentOS Additional Packages:

```
su - root # or sudo sh
yum install autoconf automake gcc cvs
yum install cairo cairo-devel \
  pango pango-devel \
  glib glib-devel \
  libxml2 libxml2-devel \
  gd gd-devel libXpm-devel libXpm
yum install openssl openssl-devel
yum install net-snmp net-snmp-lib net-snmp-utils net-snmp-perl perl-Net-SSLeay
yum install httpd fping
yum install make groff cpan crontabs
yum -y install dejavu* perl-Authen-SASL
```

### Debian/Ubuntu Additional Packages:

```
su root # or sudo sh
apt-get install autoconf automake gcc make \
  libcairo2 libcairo2-dev libglib2.0-dev libpango1.0-dev libxml2 libxml2-dev libgd-gd2-perl \
  libnet-ssleay-perl libcrypt-ssleay-perl apache2 fping \
  snmp snmpd libnet-snmp-perl libcrypt-passwdmd5-perl \
  libjson-xs-perl libnet-dns-perl libio-socket-ssl-perl libwww-perl libnet-smtp-ssl-perl \
  libcrypt-unixcrypt-perl libdata-uuid-perl \
  libproc-processtable-perl libnet-ldap-perl libnet-snpp-perl libdbi-perl libtime-modules-perl libauthen-sasl-perl
```

## RRDTool Installation

We now need to get RRDtool, by Tobi Oetiker. The NMIS development team suggests that you download the latest stable RRDtool and compile it yourself - however, for normal NMIS deployments the distribution-provided RRDtool is generally sufficient.

If you want to skip the build step and use packaged versions, simply install the following packages and then skip to the next step.

### Debian/Ubuntu:

```
apt-get install rrdtool librrds-perl
```

## CentOS/Red Hat:

There are suitably recent RRDTool packages for CentOS, but they are not part of the 'default' repositories (these don't contain anything or very outdated versions). To work around this you need to enable the [RepoForge repository](#) first to gain access.

```
# adding repoforge for centos 6:
yum install http://pkgs.repoforge.org/rpmforge-release/rpmforge-release-0.5.3-1.el6.rf.x86_64.rpm
# centos 6 ships an ancient rrdtool, repoforge-extras has what we're after
yum -y --enablerepo=rpmforge-extras install rrdtool rrdtool-perl
```

## Compiling your own RRDtool:

The latest version of RRD is available at <http://oss.oetiker.ch/rrdtool/>.

Unpack the RRDtool tarball. NMIS8 has been testing using v1.4.7 and for this example. To build and install a "GNU" make based system, you run . /configure, then make, then make install, for RRDTool for NMIS we need a configure option setup.

```
tar -xvf rrdtool-1.4.8.tar.gz
[local]# cd rrdtool-1.4.8
[rrdtool-1.4.8]#
./configure --enable-shared --prefix=/usr/local/rrdtool
```

When this step is finished, you should see something like this:

```
ordering CD from http://tobi.oetiker.ch/wish .... just kidding ;-)
```

```
-----  
Config is DONE!
```

```
With MMAP IO: yes  
Build rrd_getopt: no  
Static programs: no  
Perl Modules: perl_piped perl_shared  
Perl Binary: /usr/bin/perl  
Perl Version: 5.8.8  
Perl Options: PREFIX=/usr/local/rrdtool LIB=/usr/local/rrdtool/lib/perl/5.8.8  
Ruby Modules:  
Ruby Binary: no  
Ruby Options: sitedir=/usr/local/rrdtool/lib/ruby  
Build Lua Bindings: no  
Build Tcl Bindings: no  
Build Python Bindings: no  
Build rrdcgi: yes  
Build librrd MT: yes  
Use gettext: yes  
With libDBI: no  
  
Libraries: -lxml2 -lcairo -lcairo -lcairo -lm -lcairo -lpng12 -lglib-2.0 -lpangocairo-1.0 -lpango-1.0 -lcairo -  
lgobject-2.0 -lgmodule-2.0 -ldl -lglib-2.0  
  
Type 'make' to compile the software and use 'make install' to  
install everything to: /usr/local/rrdtool.  
  
... that wishlist is NO JOKE. If you find RRDtool useful  
make me happy. Go to http://tobi.oetiker.ch/wish and  
place an order.  
  
-- Tobo Oetiker <tobi@oetiker.ch>  
-----
```

So far, so good. You have all the things RRDtool needs to compile. Let's keep going.

```
make && make install
```

Now, rrdtool should be installed the way NMIS likes it.

## NMIS8 Installation

And speaking of NMIS, if you haven't already downloaded it, the latest version (v8.5G at time of this writing) is available at <http://opmantek.com>

```
cd ~  
tar -xvzf nmis8.5g.tar.gz
```

The unpacking of the tarball in the previous step created a directory called `nmis8.5g`, which can be installed manually or semi-automatically.

## Automatic Installation

The easiest method for installing NMIS 8.5g (or upgrading to this version) is to use the semi-automatic installation and upgrade script, `install.pl`. Simply run the commands shown below and the remaining steps of the installation procedure will be taken care for you:

```
cd nmis8.5g
./install.pl -h
...shows the list of options for the install script
./install.pl listdeps=true
...shows what perl modules are missing, and how to get them
/install.pl
...simply follow the on-screen prompts to perform the actual installation.
```

The installer will likely tell you that a number of required CPAN modules are missing and have to be installed, but the on-screen prompts will guide you through that process.

That's all there is to it, NMIS should now be installed and working.

## Manual Installation

If you don't want to use the install script for any reason then please continue with the remaining instructions in this document to perform a manual installation.

**Please note that these instructions will not be updated in the future** (except for gross problems) as we're focusing on improving the automatic installer to simplify all sorts of deployments and upgrades; under all normal circumstances the automatic installer does a more comprehensive job already. The instructions below are therefore mainly of historic interest - You should not need to perform any of these manual steps!

At this point, I like to copy this entire directory to the /usr/local directory tree and make some symlinks, like so.

```
mkdir /usr/local/nmis8/
cp -r nmis8.4.8g/* /usr/local/nmis8/
rm -rf ~/nmis8.4.8g
cd /usr/local
```

My <nmisdir> is, in this case, /usr/local/nmis8. For those who have been using NMIS for a while, make a /usr/local/nmis.

```
ln -s nmis8 nmis
```

The <nmisdir> tree needs to be owned by the user "nmis", which doesn't exist yet. Furthermore, the userID under which Apache runs ("apache" on Red Hat and "www-data" on Debian) needs to have access to the <nmisdir> tree. So, we need to add to make these things happen. We are still in the /usr/local directory at this point.

```
useradd nmis
groupadd nmis
#this may return "groupadd: group 'nmis' already exists" this can safely be ignored but if you want to verify
try this:
groups nmis
```

Now we add the Apache userID to the nmis group.

```
usermod -G nmis apache # Red Hat/CentOS
adduser nmis www-data # Debian/Ubuntu
```

NOTE: the stuff that used to be sitting in <nmisdir>/conf in previous NMIS tarballs is now located in <nmisdir>/install. Since we're doing a new installation here, we'll just copy everything in <nmisdir>/install to <nmisdir>/conf.

```
cd /usr/local/nmis/  
cp -rPv install/* conf/  
# also copy models-install to models  
cp models-install/* models/
```

## Perl CPAN Module Installation

Now, let's get busy with the Perl modules. NMIS, being written in Perl, uses numerous "Perl modules". These are available from the Comprehensive Perl Archive Network, known affectionately as "CPAN". No, not "C-SPAN", I mean "CPAN". :-)

Here's how we get to CPAN.

```
perl -MCPAN -e "shell"  
# alternatively, with more modern cpan installations:  
cpan
```

You will be asked a bunch of questions. It is safe to select all the defaults. Indeed, if you don't know what the questions are asking you, selecting the defaults is strongly recommended! HOWEVER, with some distros (Debian is a notable exception) you must select some CPAN mirrors, because that's where you're going to download your Perl modules from. Choose those which are closest to you. For me, that's North America; for Peruvians and Brazilians, that's South America; and so on.

On some distros like Red Hat/CentOS, you should start with this command. Debian doesn't seem to require it, which is good, because the command doesn't work on Debian's CPAN client anyway. :-)

```
$cpan> o conf set LC_ALL=C #(Red Hat/CentOS)
```

Additionally, I like to ensure my CPAN client program, which is what we're running ("CPAN.pm"), is the latest, up-to-date version. This will take a while and ask for several other "dependency" modules. Just tell it "yes" each time. If you would like to have cpan automatically choose yes use this code:

```
$cpan> o conf prerequisites_policy follow  
$cpan> o conf commit #use this if you want to save this setting for next time
```

And run this code to do the update:

```
$cpan>  
$cpan> reload cpan
```

OK, we've got CPAN prepped, locked, and loaded. :-) Now, install all the Perl modules listed in the official NMIS docs, like so.

```
$cpan> (assume the following lines are at this prompt)
```

```

install JSON
install JSON::XS
install Proc::ProcessTable
install Image::Resize
install Net::SNMP
install Time::HiRes
install Time::ParseDate
install Statistics::Lite
install Net::Syslog
install Net::SNPP
install Net::LDAP
install Net::SMTP
install Net::SMTP::SSL
install Net::DNS # Only required if using DNS lookups for geoplots
install IPC::Shareable # Only required if multithreading on Unix/Linux.
install BSD::Resource # Only required if multithreading on Unix/Linux.
install GD::Graph # Only required for Capacity Planning tool.
install Net::EasyTCP # Only required for master poller operation.
install Data::Dumper # Required for cgi metrics stats caching
install Cache::Mmap # required for caching cgi metric stat
install Proc::Queue
install Date::Parse
install Date::Calc
install Math::Round
install DBI
install CGI::Pretty
install Crypt::DES
#Now, you're ready to exit CPAN.
exit

```

With recent versions of cpan it's also possible to give the modules to install on the command line:

```

cpan Math::Round DBI CGI::Pretty # ...and so on

```

## Unix File Permissions

Make nmis user and group own all the files.

```

chown -Rv nmis:nmis <nmisdir>
chmod -Rv g+w <nmisdir>

```

If you decide on a different user and group name, update the NMIS Configuration Config.nmis accordingly.

One of the joys of Unix is granular file permissions, one of the frustrations of Unix is granular file permissions. To assist we have added a handy script in the admin directory <nmisdir>/admin/fixperms.pl which will read the NMIS Configuration and fix the permissions accordingly.

```

admin/fixperms.pl

# This script will fix the permissions for NMIS based on the configuration /usr/local/nmis8/admin/./conf
/Config.nmis
# The directory to be processed is: /usr/local/nmis8
# The user will be set to: nmis
# The group will be set to: nmis

```

## Initial NMIS Test

You should now be set. As an initial check, I like to go into my <nmisdir>/cgi-bin directory and run

```
cd /usr/local/nmis8/cgi-bin
./nmiscgi.pl
```

and see what happens. If all is working, you should get a bunch of HTML code. If you don't, it'll probably be NMISCGI complaining that it cannot find a given Perl module. Review your steps above; you likely missed something. The important thing to do is not to panic.

## Initial NMIS Troubleshooting

### RRDTool Libraries not Found

*(This should no longer be a problem on NMIS8, but if you did something different with RRDTool, this is an easy way to fix it.)*

For some reason, on some distros, NMIS will not be able to find the file RRDs.pm, which is a Perl module that comes with RRDtool, and that NMIS needs to be able to find and use. I ran into this problem. Fortunately, the NMIS development team included a small hack to take care of this. The line explicitly "uses" the RRDs.pm file in the /usr/local/rrdtool directory tree, but you have to uncomment that line in the NMIS program files themselves.

One way to quickly accomplish that is to run the following script within /usr/local/nmis8/bin and /usr/local/nmis8/cgi-bin. You can just cut 'n' paste it into your terminal.

```
#!/bin/bash
for file in *.pl
do
    sed -i 's|#use lib "/usr/local/rrdtool|use lib "/usr/local/rrdtool|g' $file
done
```

## NMIS Config Test

Once this is working, you're ready to head into the <nmisdir>/bin directory and do the "acid" test.

```
cd /usr/local/nmis8
./bin/nmis.pl type=config debug=true
```

You should get what looks like a long checklist with "OK" at the end of every check. If you do this, then NMIS itself is configured correctly. The output will look something like this:

```
./bin/nmis.pl type=config debug=true
NMIS version 8.1.1

Root directory of NMIS is /usr/local/nmis8

05:20:05 checkConfig, Config Checking - Checking database directories
05:20:05 loadTable, ERROR file does not exist dir=var name=nmis-event
05:20:05 writeHashtoFile, write data to /usr/local/nmis8/var/nmis-event.nmis
05:20:05 loadTable, ERROR file does not exist dir=var name=nmis-system
05:20:05 writeHashtoFile, write data to /usr/local/nmis8/var/nmis-system.nmis
Continue with bin/nmis.pl type=apache for configuration rules of the Apache web server
```

The errors above are OK, NMIS will fix them itself.

## NMIS Web Server Setup

Now, we need to set up the Web server so that folks can actually \*get to\* NMIS and look at the data. :-) That means generating an Apache config file pointing to the NMIS directory tree. NMIS itself is smart enough to be able to generate this for us. Thanks, NMIS team!

All you have to do is this.



```
./bin/nmis.pl type=apache > 00nmis.conf
```

We name the file prepending 00 so that it is the first config file read. The main part of the Apache config is as follows.

```
# NMIS Aliases

Alias /nmis8/ "/usr/local/nmis8/htdocs/"
<Directory ">
  Options Indexes FollowSymLinks MultiViews
  AllowOverride None
  Order allow,deny
  Allow from all
</Directory>

Alias /menu8/ "/usr/local/nmis8/menu/"
<Directory "/menu8">
  Options Indexes FollowSymLinks MultiViews
  AllowOverride None
  Order allow,deny
  Allow from all
</Directory>

ScriptAlias /cgi-nmis8/ "/usr/local/nmis8/cgi-bin/"
<Directory "/usr/local/nmis8/cgi-bin">
  Options +ExecCGI
  Order allow,deny
  Allow from all
</Directory>
```

You can use whatever filename you want instead of "00nmis.conf". This is very distro-dependent.

## CentOS and other Red Hat derivatives:

Copy 00nmis.conf to /etc/httpd/conf.d/

```
cp ./00nmis.conf /etc/httpd/conf.d/
```

## Debian/Ubuntu:

Copy 00nmis.conf /etc/apache2/sites-available/, and enable that site:

```
cp ./00nmis.conf /etc/apache/sites-available/nmis
a2ensite nmis
```

You could also integrate the 00nmis.conf into your existing web server configuration under its own virtual host, or into an existing set of hosts. This means you can run other version of NMIS on the same server, or for different customers for example, this is intended to be flexible for alternate integration.

Now that your NMIS config file for Apache is present--and if necessary, tweaked for your distro--start Apache itself. If it's already started, then restart it. How to do this is also very distro-dependent. Unlike me, BSD and Solaris users don't need no steenkin' startup scripts. :-)

## Auto Start HTTPD Server

Ensure that you tell Linux to start HTTPD automatically

```
/sbin/chkconfig httpd on
```

## Debian/Ubuntu Users

Nothing should be necessary as the apache2 is set to automatically start during the initial package installation, but if this should have failed:

```
update-rc.d apache2 defaults
```

## Start HTTPD Server

You will need to start or restart Apache after configuration.

## CentOS/Red Hat:

```
/etc/init.d/httpd start  
# or  
/etc/init.d/httpd restart
```

## Debian/Ubuntu:

```
service apache2 start  
# or  
service apache2 restart
```

If you get an error like this, it just means the hostname and FQDN on the Linux box hasn't been setup right, verify you have this setup correctly and update the httpd.conf for Apache accordingly.

```
Starting httpd: httpd: apr_sockaddr_info_get() failed for nmis-dev  
httpd(apache2) : Could not reliably determine the server's fully qualified domain name, using 127.0.0.1 for  
ServerName
```

## Ensure Security settings will allow you to access the web server

For the purpose of this guide we will just disable security settings, this may not be the correct answer for you depending on your situation

## IPTables

We will just disable ip tables, on Redhat/CentOS

```
service iptables stop #make sure the service isn't running  
chkconfig iptables off
```

## SELinux

Again, just turn it off. Edit /etc/selinux/config

```
#change this line:  
SELINUX=enforcing  
#to this:  
SELINUX=disabled
```

You can reboot now, or you can step SELINUX with the following command.

```
echo 0 >/selinux/enforce
```

## Quick Testing

Now, fire up your favourite Web browser (mine's Konqueror) and surf to your server. The URL will be "<http://yourhostname.domain.tld/cgi-nmis8/nmiscgi.pl>". Of course, if you have no DNS or hosts lookup for this box yet, just replace "yourhostname.domain.tld" with your new NMIS server's IP address.

You should get the NMIS Dashboard. If you don't, then check to make sure you included the "/cgi-nmis8/nmiscgi.pl" in the URL. Yep, I've made that mistake too many times to count. It's easy to forget, so make sure you include it.

Congratulations! It works, and you're almost done. We're in the home stretch.

## NMIS Crontab Configuration

We ensured crontab was installed above, now we'll make sure it's running:

```
chkconfig crond on
/etc/init.d/crond start
```

Debian/Ubuntu: nothing needs to be done. Cron is automatically started.

Now we have to tell NMIS to run the data collection every so often. For that, we use a cron job. Edit root's crontab ("crontab -e" at the root shell), like so. Remember to change the first line, the "MAILTO" line, to contain your actual email address.

```
MAILTO=WhoeverYouAre@yourdomain.tld
#####
# NMIS8 Cronfig
#####
# Run Statistics Collection
*/5 * * * * /usr/local/nmis8/bin/nmis.pl type=collect mthread=true maxthreads=10
#####
# Run Summary Update more frequently
*/3 * * * * /usr/local/nmis8/bin/nmis.pl type=summary
#####
# Run the interfaces 4 times an hour with Thresholding on!!!
*/15 * * * * nice /usr/local/nmis8/bin/nmis.pl type=threshold
#####
# Run the update once a day
30 20 * * * nice /usr/local/nmis8/bin/nmis.pl type=update mthread=true maxthreads=10
#####
# Check to rotate the logs 4am every day UTC
5 20 * * * /usr/sbin/logrotate /usr/local/nmis8/conf/logrotate.conf
#####
# save this crontab every day
0 8 * * * crontab -l > /usr/local/nmis8/conf/crontab.root
# These file locations for are Red Hat.
#0 8 * * * cp /etc/httpd/conf/httpd.conf /usr/local/nmis8/conf/httpd.conf.backup
#0 8 * * * cp /etc/httpd/conf.d/00nmis.conf /usr/local/nmis8/conf/00nmis.conf.backup
#0 8 * * * cat /etc/syslog.conf > /usr/local/nmis8/conf/syslog.conf.backup
# These file locations are for Debian/Ubuntu
#0 8 * * * cp /etc/apache2/apache2.conf /usr/local/nmis8/conf/httpd.conf.backup
#0 8 * * * cp /etc/apache2/sites-available/00nmis.conf /usr/local/nmis8/conf/00nmis.conf.backup
#0 8 * * * cat /etc/rsyslog/50-default.conf > /usr/local/nmis8/conf/syslog.conf.backup
```

Again, this file was originally generated for Red Hat / CentOS. The lines to change for other distros (e. g. Debian) are the last three. Everything else should be fine.

After you're done, issuing a crontab -l is a good idea to make sure it's all there. Note that crontab -e on most GNU/Linux distros uses the traditional "vi" editor, whereas on Debian, "nano" is used.

Note that there is a reference in this crontab to a <nmisdir>/conf/logrotate.conf file. That file contains a reference to <nmisdir>/logs/ciscopix.log. This is for those who run Cisco PIX/ASA Firewalls. Unfortunately, this file is not in the tarball for NMIS, so you'll get a non-fatal "Hey, there's no ciscopix.log file for me to logrotate!" every day. Fortunately, that's easy to fix.

```
touch <nmisdir>/logs/ciscopix.log
```

And now, go ahead and add your hosts to NMIS. There's a nice GUI for doing that which is pretty easy to use. Hey, if an MCSE like me can figure this out, anyone can. :-)

Time to celebrate! You're done! Enjoy your shiny new NMIS installation, and be sure to thank the teams that brought it to you. That would be the NMIS development team and Tobias Oetiker.

## SNMPD, Net-SNMP and Testing your Configuration

NMIS8 has been shipped with two nodes configured for collection, to prove it is all working, one is a pingonly node which is ping of the localhost, and the other is localhost using SNMP, and in the <nmisdir>/conf/snmpd.conf is what you need to get it working.

Because you have already installed NET-SNMP earlier, you just need to backup that config and move in the NMIS sample one, you should modify this accordingly to secure the SNMP read access according to your organisations security policy.

```
cd /usr/local/nmis8
mv /etc/snmp/snmpd.conf /etc/snmp/snmpd.conf.orig
mv /etc/snmp/snmptrapd.conf /etc/snmp/snmptrapd.conf.orig
cp install/snmpd.conf /etc/snmp/snmpd.conf
cp install/snmptrapd.conf /etc/snmp
cp install/snmptrapd.options /etc/sysconfig/snmptrapd # for CentOS/Red Hat only
```

### Debian/Ubuntu

The last command in the list above will fail as there is no /etc/sysconfig directory on this platform. Instead the snmptrap service is started by the snmpd service, and is enabled by updating the file /etc/default/snmpd:

```
TRAPDRUN=yes
```

### Auto Start SNMP Daemons

Ensure that you tell Linux to start SNMPD automatically

```
/sbin/chkconfig snmpd on
/sbin/chkconfig snmptrapd on
```

Debian/Ubuntu - this is start both snmp and snmptrap services

```
sysv-rc-conf snmpd on
service snmpd start
```

### Start SNMPD Manually

The first time you can start it manually.

```
/etc/init.d/snmpd start
/etc/init.d/snmptrapd start
```

## Run a Test Update

With everything else done, you can run a test update

```
bin/nmis.pl type=update node=localhost debug=true
```

The output will look something like

```
NMIS version 8.1.1
09:24:29 runThreads, Starting
09:24:31 runThreads, tables loaded
09:24:31 runThreads, Kernel name of NMIS server is linux
09:24:31 runDaemons, Starting
09:24:31 runDaemons, Finished
09:24:31 runThreads, INFO start of update process
09:24:31 doUpdate, =====
09:24:31 doUpdate, Starting, node localhost
09:24:31 init, info of node=localhost loaded
09:24:31 init, info of nmis-system loaded
09:24:31 getNodeCfg, cfg of node=localhost found
09:24:31 init, cfg of node=localhost loaded
09:24:31 loadModel, INFO, model Model loaded
09:24:31 initsnmp, snmp for node=localhost initialized
09:24:31 runPing, Starting localhost (127.0.0.1) with timeout=300 retries=3 packet=56
09:24:34 runPing, localhost is PINGABLE min/avg/max = 0.036/0.070/0.121 ms loss=0%
09:24:34 checkEvent, event localhost-node_down- marked for UP notify and delete
09:24:34 writeHashtoFile, write data to /usr/local/nmis8/var/nmis-event.nmis
09:24:34 getLevelLogEvent, node=localhost, event=Node Up, role=distribution not found in class="event" of model=
09:24:34 runPing, Finished with exit=1
09:24:34 getNodeInfo, Starting
09:24:34 getNodeInfo, sysObjectId=1.3.6.1.4.1.8072.3.2.10, sysObjectName=enterprises.8072.3.2.10
09:24:34 getNodeInfo, sysDescr=Linux nmis-dev 2.6.18-194.el5 #1 SMP Fri Apr 2 14:58:35 EDT 2010 i686
09:24:34 getNodeInfo, oid index 8072, Vendor is net-snmp
09:24:34 selectNodeModel, INFO, Model 'net-snmp' found for Vendor net-snmp
09:24:34 loadModel, INFO, model Model-net-snmp loaded
09:24:34 getDNSloc, Starting
09:24:34 getDNSloc, Finished
09:24:34 checkPIX, Starting
09:24:34 checkPIX, Finished
09:24:34 getNodeInfo, Finished with exit=1
09:24:34 getIntfInfo, Starting
09:24:34 getIntfInfo, Get Interface Info of node localhost, model net-snmp
09:24:34 getIntfInfo, ifIndex=1 ifDescr=lo ifType=softwareLoopback ifAdminStatus=up ifOperStatus=up
ifSpeed=100000000
09:24:34 getIntfInfo, ifIndex=2 ifDescr=eth0 ifType=ethernetCsmacd ifAdminStatus=up ifOperStatus=up
ifSpeed=1000000000
09:24:34 getIntfInfo, ifIndex=3 ifDescr=sit0 ifType=tunnel ifAdminStatus=down ifOperStatus=down ifSpeed=0
09:24:34 getIntfInfo, Getting Device IP Address Table
09:24:34 getIntfInfo, ifIndex=1, addr=127.0.0.1 mask=255.0.0.0
09:24:34 getIntfInfo, ifIndex=2, addr=192.168.1.7 mask=255.255.255.0
09:24:34 getIntfInfo, ifIndex 1, collect=false, found softwareLoopback in ifType
09:24:34 getIntfInfo, ifIndex 2, collect=true
09:24:34 getIntfInfo, ifIndex 3, collect=false, ifAdminStatus eq down|testing|null
09:24:34 getIntfInfo, Finished
09:24:34 doUpdate, node=localhost role=distribution type=server
09:24:34 doUpdate, vendor=net-snmp model=net-snmp interfaces=3
```

```
09:24:34 getEnvInfo, Starting
09:24:34 getEnvInfo, Get Environment Info of node localhost, model net-snmp
09:24:34 getEnvInfo, No class 'environment' declared in Model
09:24:34 getEnvInfo, Finished
09:24:34 getCBQoS, no collecting (none) for node localhost
09:24:34 getCalls, no collecting for node localhost
09:24:34 runReach, Starting node localhost, type=server
09:24:34 runReach, Outage for localhost is
09:24:34 runReach, Getting Interface Utilisation Health
09:24:34 getSummaryStats, Start type=interface, index=2, start=-15 minutes, end=1300973074
09:24:34 getTypeName, ERROR (localhost) type=interface not found in graphtype table
09:24:34 getDBName, ERROR (localhost) database name not found graphtype=interface (section=0), index=2, item=
09:24:34 runReach, Intf Summary in= out= intsumm=200 count=1
09:24:34 runReach, Calculation of health=100
09:24:34 runReach, Reachability and Metric Stats Summary
09:24:34 runReach, collect=true (Node table)
09:24:34 runReach, ping=100 (normalised)
09:24:34 runReach, cpuWeight=100 (normalised)
09:24:34 runReach, memWeight=100 (normalised)
09:24:34 runReach, intWeight=100.00 (100 less the actual total interface utilisation)
09:24:34 runReach, responseWeight=100 (normalised)
09:24:34 runReach, total number of interfaces=2
09:24:34 runReach, total number of interfaces up=0
09:24:34 runReach, total number of interfaces collected=1
09:24:34 runReach, total number of interfaces coll. up=
09:24:34 runReach, availability=100
09:24:34 runReach, cpu=
09:24:34 runReach, health=100
09:24:34 runReach, intfColUp=
09:24:34 runReach, intfCollect=1
09:24:34 runReach, intfTotal=2
09:24:34 runReach, intfUp=0
09:24:34 runReach, loss=0
09:24:34 runReach, mem=100
09:24:34 runReach, operCount=
09:24:34 runReach, operStatus=
09:24:34 runReach, reachability=100
09:24:34 runReach, responsetime=0.070
09:24:34 updateRRD, Starting RRD Update Process, type=health, index=,item=
09:24:34 getFileName, filename of type=health is /usr/local/nmis8/database/health/server/localhost-reach.rrd
09:24:34 updateRRD, file=/usr/local/nmis8/database/health/server/localhost-reach.rrd not found, try nmis4 format
09:24:34 getFileName, filename of type=health is /usr/local/nmis8/database/health/server/127.0.0.1-reach.rrd
09:24:34 createRRD, Starting
09:24:34 createRRD, check and/or create RRD database /usr/local/nmis8/database/health/server/localhost-reach.rrd
09:24:34 optionsRRD, type health
09:24:34 optionsRRD, INFO, using database format 'default'
09:24:34 createRRD, options to create database /usr/local/nmis8/database/health/server/localhost-reach.rrd
09:24:34 createRRD, -b
09:24:34 createRRD, 1300973070
09:24:34 createRRD, -s
09:24:34 createRRD, 300
09:24:34 createRRD, DS:availability:GAUGE:900:0:100
09:24:34 createRRD, DS:health:GAUGE:900:0:100
09:24:34 createRRD, DS:intfColUp:GAUGE:900:0:U
09:24:34 createRRD, DS:intfCollect:GAUGE:900:0:U
09:24:34 createRRD, DS:intfTotal:GAUGE:900:0:U
09:24:34 createRRD, DS:intfUp:GAUGE:900:0:U
09:24:34 createRRD, DS:loss:GAUGE:900:0:100
09:24:34 createRRD, DS:reachability:GAUGE:900:0:100
09:24:34 createRRD, DS:responsetime:GAUGE:900:0:U
09:24:34 createRRD, RRA:AVERAGE:0.5:1:2304
09:24:34 createRRD, RRA:AVERAGE:0.5:6:1536
09:24:34 createRRD, RRA:AVERAGE:0.5:24:2268
09:24:34 createRRD, RRA:AVERAGE:0.5:288:1890
09:24:34 createRRD, RRA:MAX:0.5:1:2304
09:24:34 createRRD, RRA:MAX:0.5:6:1536
09:24:34 createRRD, RRA:MAX:0.5:24:2268
09:24:34 createRRD, RRA:MAX:0.5:288:1890
09:24:34 createRRD, RRA:MIN:0.5:1:2304
09:24:34 createRRD, RRA:MIN:0.5:6:1536
09:24:34 createRRD, RRA:MIN:0.5:24:2268
```

```
09:24:34 createRRD, RRA:MIN:0.5:288:1890
09:24:34 createRRD, INFO (localhost) created RRD /usr/local/nmis8/database/health/server/localhost-reach.rrd
09:24:35 createRRD, Finished
09:24:35 updateRRD, DS intfTotal:intfColUp:intfCollect:health:intfUp:availability:reachability:responsetime:loss
09:24:35 updateRRD, value N:2:0:1:100:0:100:100:0.070:0
09:24:35 runReach, Finished
09:24:35 writeHashtoFile, write data to /usr/local/nmis8/var/localhost-view.nmis
09:24:35 writeHashtoFile, write data to /usr/local/nmis8/var/localhost-node.nmis
09:24:35 doUpdate, Finished
09:24:35 runThreads, ### continue normally ###
09:24:35 runThreads, === debug output suppressed with debug=1 ===

09:24:35 End of bin/nmis.pl Processed 1 nodes ran for 6 seconds.
```

## Run a Test Collection

With everything else done, you can run a test update

```
# bin/nmis.pl type=collect node=localhost debug=true
```

The output will look something like

```
NMIS version 8.1.1
09:27:47 runThreads, Starting
09:27:47 runThreads, tables loaded
09:27:47 runThreads, Kernel name of NMIS server is linux
09:27:47 runDaemons, Starting
09:27:47 runDaemons, Finished
09:27:47 doCollect, =====
09:27:47 doCollect, Starting, node localhost
09:27:47 init, info of node=localhost loaded
09:27:47 init, info of nmis-system loaded
09:27:47 getNodeCfg, cfg of node=localhost found
09:27:47 init, cfg of node=localhost loaded
09:27:47 loadModel, INFO, model Model-net-snmp loaded
09:27:47 initsnmp, snmp for node=localhost initialized
09:27:47 doCollect, node=localhost role=distribution type=server
09:27:47 doCollect, vendor=net-snmp model=net-snmp interfaces=3
09:27:47 runPing, Starting localhost (127.0.0.1) with timeout=300 retries=3 packet=56
09:27:49 runPing, localhost is PINGABLE min/avg/max = 0.026/0.034/0.041 ms loss=0%
09:27:49 runPing, Finished with exit=1
09:27:49 updateNodeInfo, Starting Update Node Info, node localhost
09:27:49 updateNodeInfo, sysUpTime: Old=0:05:49 New=0:06:03
09:27:49 checkPIX, Starting
09:27:49 checkPIX, Finished
09:27:49 updateNodeInfo, Finished with exit=1
09:27:49 getNodeData, Starting Node get data, node localhost
09:27:49 getValues, no oid loaded for section=
09:27:49 getIntfData, Starting Interface get data, node localhost
09:27:49 getIntfData, lo: ifIndex=1, was => OperStatus=up, ifAdminStatus=up, Collect=false
09:27:49 getIntfData, NOT Collected: lo: ifIndex=1, OperStatus=up, ifAdminStatus=up, Interface Collect=false
09:27:49 getIntfData, eth0: ifIndex=2, was => OperStatus=up, ifAdminStatus=up, Collect=true
09:27:49 getIntfData, collect interface index=2
09:27:49 getIntfData, status now admin=up, oper=up was admin=up, oper=up
09:27:49 getIntfData, rrdData section interface, ds ifOperStatus, value=100, option=gauge,0:100
09:27:49 getIntfData, rrdData section interface, ds ifInOctets, value=14431330, option=counter,0:U
09:27:49 getIntfData, rrdData section interface, ds ifOutOctets, value=707043, option=counter,0:U
09:27:49 updateRRD, Starting RRD Update Process, type=interface, index=2,item=
09:27:49 getFileName, filename of type=interface is /usr/local/nmis8/database/interface/server/localhost
/localhost-eth0.rrd
09:27:49 updateRRD, database /usr/local/nmis8/database/interface/server/localhost/localhost-eth0.rrd exists and
is R/W
```

```
09:27:49 updateRRD, DS ifOperStatus:ifInOctets:ifOutOctets
09:27:49 updateRRD, value N:100:14431330:707043
09:27:49 getSummaryStats, Start type=interface, index=2, start=-6 hours, end=1300973269
09:27:49 getDBName, database name=/usr/local/nmis8/database/interface/server/localhost/localhost-eth0.rrd
09:27:49 getSummaryStats, name=availability, index=2, value=NaN
09:27:49 getSummaryStats, name=inputUtil, index=2, value=NaN
09:27:49 getSummaryStats, name=outputUtil, index=2, value=NaN
09:27:49 getSummaryStats, name=totalUtil, index=2, value=NaN
09:27:49 getSummaryStats, name=inputBits, index=2, value=NaN
09:27:49 getSummaryStats, name=outputBits, index=2, value=NaN
09:27:49 getIntfData, sit0: ifIndex=3, was => OperStatus=down, ifAdminStatus=down, Collect=false
09:27:49 getIntfData, NOT Collected: sit0: ifIndex=3, OperStatus=down, ifAdminStatus=down, Interface
Collect=false
09:27:49 getIntfData, : ifIndex=, was => OperStatus=, ifAdminStatus=, Collect=
09:27:49 getIntfData, NOT Collected: : ifIndex=, OperStatus=, ifAdminStatus=, Interface Collect=
09:27:49 getIntfData, : ifIndex=, was => OperStatus=, ifAdminStatus=, Collect=
09:27:49 getIntfData, NOT Collected: : ifIndex=, OperStatus=, ifAdminStatus=, Interface Collect=
09:27:49 getIntfData, : ifIndex=, was => OperStatus=, ifAdminStatus=, Collect=
09:27:49 getIntfData, NOT Collected: : ifIndex=, OperStatus=, ifAdminStatus=, Interface Collect=
09:27:49 getIntfData, : ifIndex=, was => OperStatus=, ifAdminStatus=, Collect=
09:27:49 getIntfData, NOT Collected: : ifIndex=, OperStatus=, ifAdminStatus=, Interface Collect=
09:27:49 getIntfData, Finished
09:27:49 getEnvData, Starting
09:27:49 getEnvData, Get Environment Data of node localhost, model net-snmp
09:27:49 getEnvData, No class 'environment' declared in Model
09:27:49 getEnvData, Finished
09:27:49 getCBQoS, no collecting (none) for node localhost
09:27:49 getCalls, no collecting for node localhost
09:27:49 runServer, Starting server device/storage collection, node localhost
09:27:49 getValues, collect of section device skipped by indexed section
09:27:49 getValues, collect of type cpu skipped by NON indexed section, check this Model
09:27:49 runServer, device Descr=network interface sit0, Type=1.3.6.1.2.1.25.3.1.4
09:27:49 getValues, collect of type cpu skipped by NON indexed section, check this Model
09:27:49 runServer, device Descr=Guessing that there's a floating point co-processor, Type=1.3.6.1.2.1.25.3.1.12
09:27:49 getValues, collect of type cpu skipped by NON indexed section, check this Model
09:27:49 runServer, device Descr=SCSI disk (/dev/sda), Type=1.3.6.1.2.1.25.3.1.6
09:27:49 getValues, collect of type cpu skipped by NON indexed section, check this Model
09:27:49 runServer, device Descr=network interface eth0, Type=1.3.6.1.2.1.25.3.1.4
09:27:49 getValues, collect of type cpu skipped by NON indexed section, check this Model
09:27:49 runServer, device Descr=VMware Virtual IDE CDROM Drive, Type=1.3.6.1.2.1.25.3.1.6
09:27:49 getValues, collect of type cpu skipped by NON indexed section, check this Model
09:27:49 runServer, device Descr=network interface lo, Type=1.3.6.1.2.1.25.3.1.4
09:27:49 getValues, collect of type cpu skipped by NON indexed section, check this Model
09:27:49 runServer, device Descr=Angel___3300_Series@192.168.1.8, Type=1.3.6.1.2.1.25.3.1.5
09:27:49 getValues, collect of type cpu skipped by NON indexed section, check this Model
09:27:49 runServer, device Descr=GenuineIntel: Intel(R) Core(TM)2 Duo CPU T9550 @ 2.66GHz, Type=1.
3.6.1.2.1.25.3.1.3
09:27:49 runServer, cpu Load=0.00, Descr=GenuineIntel: Intel(R) Core(TM)2 Duo CPU T9550 @ 2.66GHz
09:27:49 updateRRD, Starting RRD Update Process, type=hrsmppcpu, index=768,item=
09:27:49 getFileName, filename of type=hrsmppcpu is /usr/local/nmis8/database/health/server/localhost-
hrsmppcpu768.rrd
09:27:49 updateRRD, database /usr/local/nmis8/database/health/server/localhost-hrsmppcpu768.rrd exists and is R/W
09:27:49 updateRRD, DS hrCpuLoad
09:27:49 updateRRD, value N:3
09:27:49 runServer, storage Type=1.3.6.1.2.1.25.2.1.4, Size=5078687, Used=44473, Units=4096
09:27:49 updateRRD, Starting RRD Update Process, type=hrdisk, index=6,item=
09:27:49 getFileName, filename of type=hrdisk is /usr/local/nmis8/database/health/server/localhost-hrdisk6.rrd
09:27:49 updateRRD, database /usr/local/nmis8/database/health/server/localhost-hrdisk6.rrd exists and is R/W
09:27:49 updateRRD, DS hrDiskSize:hrDiskUsed
09:27:49 updateRRD, value N:20802301952:182161408
09:27:49 runServer, storage Type=1.3.6.1.2.1.25.2.1.3, Size=4194296, Used=0, Units=1024
09:27:49 updateRRD, Starting RRD Update Process, type=hrvmem, index=,item=
09:27:49 getFileName, filename of type=hrvmem is /usr/local/nmis8/database/health/server/localhost-hrvmem.rrd
09:27:49 updateRRD, database /usr/local/nmis8/database/health/server/localhost-hrvmem.rrd exists and is R/W
09:27:49 updateRRD, DS hrVMemUsed:hrVMemSize
09:27:49 updateRRD, value N:0:4294959104
09:27:49 runServer, storage Type=1.3.6.1.2.1.25.2.1.4, Size=101086, Used=12273, Units=1024
09:27:49 runServer, storage Type=1.3.6.1.2.1.25.2.1.2, Size=1035108, Used=968328, Units=1024
09:27:49 updateRRD, Starting RRD Update Process, type=hrmem, index=,item=
09:27:49 getFileName, filename of type=hrmem is /usr/local/nmis8/database/health/server/localhost-hrmem.rrd
09:27:49 updateRRD, database /usr/local/nmis8/database/health/server/localhost-hrmem.rrd exists and is R/W
```



```
09:27:49 updateRRD, DS hrMemUsed:hrMemSize
09:27:49 updateRRD, value N:991567872:1059950592
09:27:49 runServer, storage Type=1.3.6.1.2.1.25.2.1.4, Size=0, Used=0, Units=0
09:27:49 runServer, storage Type=1.3.6.1.2.1.25.2.1.4, Size=11649262, Used=876570, Units=4096
09:27:49 updateRRD, Starting RRD Update Process, type=hrdisk, index=4,item=
09:27:49 getFileName, filename of type=hrdisk is /usr/local/nmis8/database/health/server/localhost-hrdisk4.rrd
09:27:49 updateRRD, database /usr/local/nmis8/database/health/server/localhost-hrdisk4.rrd exists and is R/W
09:27:49 updateRRD, DS hrDiskSize:hrDiskUsed
09:27:49 updateRRD, value N:47715377152:3590430720
09:27:49 runServer, storage Type=1.3.6.1.2.1.25.2.1.1, Size=1035108, Used=172032, Units=1024
09:27:49 runServer, storage Type=1.3.6.1.2.1.25.2.1.4, Size=2539342, Used=240465, Units=4096
09:27:49 updateRRD, Starting RRD Update Process, type=hrdisk, index=5,item=
09:27:49 getFileName, filename of type=hrdisk is /usr/local/nmis8/database/health/server/localhost-hrdisk5.rrd
09:27:49 updateRRD, database /usr/local/nmis8/database/health/server/localhost-hrdisk5.rrd exists and is R/W
09:27:49 updateRRD, DS hrDiskSize:hrDiskUsed
09:27:49 updateRRD, value N:10401144832:984944640
09:27:49 runServer, Finished
09:27:49 runServices, Starting Services stats, node=localhost, nodeType=server
```