NagMIN Guide

Author: Christian Bock

Contact: nagmin-documentation@krisbe.de

FOR THE UP TO DATE VERSION HAVE A LOOK AT

http://www.krisbe.de/nagmin

1 History

Version	***Date***	***Changes***	***Annotations***
0.1	03-27-2005	initial version	work in progress
0.2	03-28-2005	explanation of features and bug fixes added	still work in progress
0.3	05-07-2005	Minor corrections, spelling etc.	

2 Table Of Content

3 Abstract

NagMIN claims to be a single point of Nagios configuration and offers quite a large set of features. Unfortunately these features are not documented, that means they are more or less useless for us. But still, after some initial headaches I set up an distributed Nagios environment whose configuration is in the hands of NagMIN .

I try to explain all I found out about NagMIN and especially all about the installation process and “basic” features, but I am only human as you guys are so please tell me about your opinion, subjects that have to be discussed or that are not appropriately documented.

Do not hesitate to contact me at nagmin-documentation@krisbe.de because I would like to keep this documentation as useful as possible.

And still have a look at Frederick's documentation because I will not repeat that document, except the installation process which I hope I can make a bit clearer to you.

Topics you can find in his document you can not find here are: Design Strategy, Inheritance, Notification Aggregator, NagMIN Navigator,NagMIN/Nagios Table Definitions.

The additional benefit of this document is the explanation of how to use the undocumented features and perhaps if there will be some feedback, this can be a knowledge base for unsolved issues in the current version of NagMIN.

There still do exist lots of bugs especially in the Portal / Graph / RRD area I hope some of you have already solved one or another issue. Please let me know so that all users can benefit from it.

And of course after the criticism a lot of thanks to Frederick who invested that much time to bring up NagMIN which has some really useful features. I would be happy if he found some time to support the process again.

4 Requirements

- Nagios 1.*
  - Be aware that the current Version of NagMIN (2.2.0) **only** supports Version 1.* of Nagios. There is currently no support of Nagios2 !
- Nmap (optional)
  - You can run NagMIN perfectly without Nmap, but of course can't use some features of automatic host and service discovery and configuration.
- Webmin
  - NagMIN is a Webmin module, so Webmin is mandatory :-)
- MySQL
  - MySQL is mandatory as well. NagMIN is storing all its information in it.
- OpenSSL
- Perl Modules
  - DBI
  - Bundle::DBD::mysql
  - Crypt::DES
  - Digest::SHA1
  - Digest::HMAC
  - Net::SNMP
  - HTML::Template
- Net-SNMP (optional)
  - Net-SNMP is optional but is needed if you want to use the automated discovery of interfaces and traffic statistics etc.
- RRDTool (optional)
  - RRDTool is optional, but is needed for automated generation of rrd files e.g. for the automated interface discovery traffic statistics
- OpenSSH
  - used for distribution to remote Nagios servers

5 Installation

5.1 Prerequisites

In this document I assume that all software is installed in default places. Of course these may vary from software to software, but I expect it to be under
/usr/local/<name[and version of the software]>
For comments on sources see the next chapter.

5.2 Sources

Personally I prefer to build my systems from source for several reasons. These are:

1) I ALWAYS had problems building a system entirely on rpms for various reasons. Which led me to install a package from source which then was not compatible with an installed rpm so I had to ... but perhaps you know the game :-)
2) I got full control over the compile options, so the software is compiled especially for my needs.
3) It is easy to update just the packages that need to be updated.

Some Unix/Linux freaks my know better, but my default path for all sources is

/usr/local/src

5.3 Nagios

You should have Nagios running and have played around with it before you are going to install NagMIN, but to give you some idea of the installation process, here it is.

5.3.1 Sources

Sources are available at www.nagios.org.
Be aware to take the latest 1.* version as NagMIN currently does not support version 2!

5.3.2 Configure

What I am showing you below is some config for an advanced mysql- installation an some more features using performance data of Nagios. I will not discuss these in this document.
One may argue not to use mysql anymore as it is not supported by Nagios2 anymore. And if you think about it twice, you will recognize that the object definitions of Nagios are never subject of using MySQL or not, they are always stored in text files, in 1.* and 2, too!

```
cd /usr/local/src/nagios-1.2
 ./configure  \
--enable-embedded-perl \
--with-perlcache \
--with-default-perfdata \
--with-mysql-lib=/usr/local/mysql/lib/mysql/ \
--with-mysql-inc=/usr/local/mysql/include/mysql/ \
--with-mysql-comments \
--with-mysql-retention \
--with-mysql-downtime \
--with-mysql-status \
--without-statuswrl
```

The reduced version is sufficient until you have some experience and understand Nagios:

```
./configure
```

Is that easy?

5.3.3 Make (Install)

```
make all
make install
make install-init
make install-commandmode
make install-config
```

Installation process finnished.

5.4 MySQL

5.4.1 Overview

MySQL is the place NagMIN is storing its information, so setup is mandatory. Please refer to MySQL documentation as well.

5.4.2 Sources

Available at: www.mysq.org.

5.4.3 Configure

```
./configure \
--prefix=/usr/local/mysql \
--without-debug
```

5.4.4 Make

```
make
make install
```

5.4.5 Some additional setup

```
groudadd mysql
chown -R mysql.mysql /usr/local/mysql
/usr/local/src/mysql<version>/scripts/mysql_install_db
cp /usr/local/src/mysql<version>/support-files/mysql.server /etc/init.d
```

5.5 Perl

5.5.1 Configure

```
./configure
```

5.5.2 Make

```
make
make install
```

5.6 Perl Modules

5.6.1 Make (install)

Each of the Perl modules in the requirements list has to be install via

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

5.7 Webmin

5.7.1 Sources

www.webmin.com

5.7.2 Installation

Simply type

```
./setup.sh
```

in the source directory.

5.8 NagMIN

5.8.1 Sources

nagmin.sourceforge.net

5.8.2 Installation

Unpack the NagMIN source in /usr/local/src

```
tar zxvf nagmin-n.n.n.tar.gz
```

Set the permissions of /usr/local/nagmin to the needs of your system

```
chmod -R nagios:nagios /usr/local/nagmin
```

The NagMIN module is then installed via Webmin:

```
-> Webmin
-> Webmin Confiuration
-> Webmin Modules
-> From local File, choose the nagmin*.wbm from nagmin-src directoy
```

6 Configuration

6.1 Webmin

Webmin runs out of the box. One good thing is perhaps to activate https. You do this at

```
-> Webmin
-> Webmin Configuration
-> SSL Encryption
-> Redirect non-SSL requests to SSL mode? -> Yes
-> Save
```

6.2 NagMIN

6.2.1 NagMIN Database Setup

In order to be successful with the database setup, you have to have a running, error free Nagios config. See chapter on Nagios config. And of course MySQL has to be running as well.

First you have to create the NagMIN database. By default NagMIN calls this database “nagios” but as I have a distributed system using several databases for several purposes I chose to call it “nagmin”. Nice idea isn't it? The user is, you might guess it: “nagmin”, but thats all up to you.

```
mysqladmin -p create nagmin
mysql -p
mysql> GRANT ALL ON nagmin.* to nagmin@localhost  IDENTIFIED BY 'somepw';
```

Of course as you want to get used to Webmin, you can do this with the Webmin MySQL module as well.

Unfortunately there is no setup script supplied in version 2.2.0, so you have to take the script from 2.1.0. Unfortunately that does not meet the requirements of 2.2.0 that means a fix is needed. I have done some (but not all) fixes.

But anyways, what the script does is to read you nagios.cfg. And your cgi.cfg an parses the information into SQL statements that we put into MySQL later.

In the original documentation it reads “Please Note! You may lose some host and service configuration information since NagMIN puts as many objects in templates as possible”. Luckily I did not experience this so far.

Download the NagMIN setup script from www.krisbe.de/nagmin.

```
cp setup <nagmin-src>/
cd  <nagmin-src>/
```

Now execute the setup script and answer all questions, but:

DO NOT BUILD THE DATABASE

because the script has some errors in sql production. And it is me, too who is to lazy to fix that. Anyone already has a working script? Please tell me!

```
./setup
grep INSERT ./install/nagmin_load.sql > ./install/insert.sql
mysql -p nagmin < ./install/nagmin_init.sql
mysql -p nagmin < ./install/nagmin_services.sql
mysql -p nagmin < ./install/insert.sql
```

In table HostDetail create the field

time_zone varchar(3) default NULL

6.2.2 Initial Configuration Of NagMIN

Go to the

Module Config

and go carefully through ALL of the options, especially do the following (for you it is easy now, some options drove me crazy to find out, honestly):

- change the **database name** to “nagmin” or whatever you called it. If you later decide to add another database for your distributed Nagios system you can have a list of space separated database names here.
- set “**USE DBI**” to YES ,
- set “**Administration Password**” via “Set to”. It is important to set the password here because the initial NagMIN dialog is asking you for the password but for some bug reason does not except any password. You have to enter it here!

6.2.3 NagMIN Generated Nagios Config

6.2.3.1 Database

We already learned that we can configure several databases. In chapter “NagMIN-Database Setup” we already set up a database for the local system called “nagmin”. If you want to setup another database just add the database name in

Module Config

to with a space separator to

NagMIN Databases.

6.2.3.2 System Host

For every database we have to choose a “system” host. For your localhost this will be localhost, or representative and for an remote host, the remote DNS and IP, respectively.
What you have to do is to

create your host in the Host table.
Add the host to the System table.

You can choose whether this is a

Local Host

If your are setting up a localhost you choose “yes”. If you choose “no” NagMIN will try to transfer the config via ssh/scp to the remote host.

6.2.3.3 Templates

Now, you have to slightly fix the configuration read from your text files. This has to do with the template infrastructure of Nagios and NagMIN. Have a look at the Nagios and NagMIN documentation for further information.
For the moment it is enough to go to

Contact Templates

and to create a

- generic-contact

Press “Verify” to see if there are some more inconsistencies in the current configuration and fix them.

To further configure your Nagios system, you simply have to fill out the tables. To do that in an appropriate way you should understand the philosophy of how Nagios configuration files are structured. Especially do read the Nagios documentation about templates. It might help you.

6.3 SSH

NagMIN has a nice feature of distributing Nagios config to distributed Nagios machines. Nagios uses SSH to upload the configs to the distributed machines. Unfortunately there is no option to teach NagMIN about some SSH passwords. Be it a security whole or not, we have to introduce the client to the server in order to skip the password process. The disadvantage is that everybody as access without password from server to client!

- On the ***client*** run the following command:

```
ssh-keygen -t dsa -f $HOME/.ssh/id_dsa -P ''
```

- Copy $HOME/.ssh/id_dsa.pub to the ***server***.

- On the ***server*** run the following commands:

```
cat id_dsa.pub >> $HOME/.ssh/authorized_keys
```

6.4 Apache

6.4.1 httpd.conf

Add these lines to you Apache config. This is the default I used for Nagios, too. As you can see I recycled the Nagios httppasswd as well.

```
#
# NagMIN
#
ScriptAlias /nagmin/cgi-bin/ "/usr/local/nagmin/sbin/"
<Directory "/usr/local/nagmin/sbin">
            Options ExecCGI FollowSymLinks
            AllowOverride AuthConfig
            Order allow,deny
            Allow from all
            AuthName "Nagios Access"
            AuthType Basic
            AuthUserFile /usr/local/nagios/etc/htpasswd.users
            require valid-user
</Directory>

Alias /nagmin/ /usr/local/nagmin/share/
<Directory "/usr/local/nagmin/share/">
            Options ExecCGI FollowSymLinks
            AllowOverride AuthConfig
            Order allow,deny
            Allow from all
            AuthName "Nagios Access"
            AuthType Basic
            AuthUserFile /usr/local/nagios/etc/htpasswd.users
            require valid-user
</Directory>
```

7 NagMIN Features Explained

7.1 Overview

The following chapters are based on the list of features taken from nagmin.sourceforge.net. I introduce the chapters by citing Fredericks feature description and, if necessary give further explanations about the usage.

7.2 Centralized Management

NagMIN can manage customized configurations for remote Nagios/NagMIN systems. NagMIN uses ssh and scp to securely perform remote commands and to transfer configuration files.

The setup of a remote Nagios machine is almost the same as the local machine. Be aware that this machine must be a Nagios 1.* as well.

1) Copy the configuration files of the remote machine to a local directory.
2) See chapter "NagMIN-Database Setup" and simply follow the instructions. To summarize the steps:
   1. Create a database with mysqladmin whose name is representing the remote instance,
   2. point the setup script to the directory containing remote configs, ...
3) See chapter "NagMIN Generated Nagios Config".
4) See chapter "Configuration"/ "SSH".
5) Now try to beam the configuration to the remote host pressing "Activate".

And as a little present there is a script called config_remote in <nagmin-src>/install/ doing that work for you. More about that in the next version of this document.

7.3 Nagios Database Integration

Compile Nagios with MySQL database support and add the Nagios DB schema to the NagMIN database. NagMIN allows you to view this information and also uses the service status information in its web portal.

7.4 SNMP Interface Scanning

NagMIN will perform an SNMP MIB II walk of SNMP enabled devices to learn their network interfaces, descriptions, index numbers and speed.

That feature really helps all network administrators who have to monitor of their router and switches. Unfortunately, there is some work to do to get it working.

```
cp nagmin-src/sbin/nm_check_interface /usr/local/nagios/libexec/
```

Did you install all the required Perl modules from Chapter “Requirements”?

Create a

```
Check Command         nm_check_interface
```

with

```
Command  Name         nm_check_interface
nm_check_interface    $USER1$/nm_check_interface -H $HOSTNAME$ -I
$HOSTADDRESS$ -C $ARG3$ -i $ARG1$ -s $ARG2$ -w $ARG4$ -c $ARG5$
```

Create a

```
Service Template      nm_check_interface
```

with

```
Check Command         nm_check_interface
```

Give the Nagios group writing permission to the nagmin/rrd/db directory.

7.5 SNMP Interface Performance Monitoring With Threshold Alerting

NagMIN uses the information stored in its Interfaces table, along with a NagMIN plugin that polls SNMP information (input and output octets and errors) for each interface and stores the results in an RRD database for each interface.

7.6 NagMIN Web Portal

NagMIN comes with a web portal that dynamically generates RRD graphs for hosts and interfaces and allows you to drill down to the retrieved information that also includes input and output interface errors as well as input and output octets. The Web Portal shows a combined graph of all hosts per host group along with an calculated average for the entire host group.

First you have to set up Apache to understand what to do with Perl files (*.pl).
Next add NagMIN to your Apache config. Have a look at the chapter “Configuration” / “Apache” for that.
And again some more work to do:

```
cp <nagmin-src>/sbin/index.pl /usr/local/nagmin/sbin/index.pl
cp -R <nagmin-src>/lib /usr/local/nagmin/
vi /usr/locaL/nagmin/lib/NagMIN/Config.pm
```

- fix the config-path:
  - from :   unless (open(CFG,"$nagminRoot/etc/$configFile")) {
  - to: unless (open(CFG,"$nagminRoot/<databae-name>/$configFile")) {

Sad, but true, there is something to see but very incomplete:

1) The summary tables do not show all configured hosts, services etc.
2) I can't get the graphs to work
   1. there are no graphs in http://myhost/nagmin/images/graphs//-gr-day.gif
3) Executing index.pl via browser results in some /va/log/messages:

```
DBD::mysql::st execute failed: Column: 'rtt_day' in field list is ambiguous at
/usr/local/nagmin/lib/NagMIN/Utils.pm line 72, <CMD> line 3.
DBD::mysql::st execute failed: Unknown column 'time_zone' in 'field list' at
/usr/local/nagmin/lib/NagMIN/Utils.pm line 72, <CMD> line 3.
DBD::mysql::st execute failed: Unknown column 'time_zone' in 'where clause' at
/usr/local/nagmin/lib/NagMIN/Utils.pm line 72, <CMD> line 3.
```

To fix this you have to extend the rtt_* and sla_* by the appropriate Table. My guess is
in HostGroup.pm with HostGroup.*
in HostGroupSummary.pm with HostGroup.*
in Host.pm with Host.*

```
NagMIN/HostGroup.pm: $sql .= "truncate(rtt_day,1),";
NagMIN/HostGroup.pm: $sql .= "truncate(rtt_week,1),";
NagMIN/HostGroup.pm: $sql .= "truncate(rtt_month,1),";
NagMIN/HostGroup.pm: $sql .= "truncate(rtt_year,1),";
NagMIN/HostGroup.pm: $sql .= "truncate(sla_day,2),";
NagMIN/HostGroup.pm: $sql .= "truncate(sla_week,2),";
NagMIN/HostGroup.pm: $sql .= "truncate(sla_month,2),";
NagMIN/HostGroup.pm: $sql .= "truncate(sla_year,2),";

NagMIN/HostGroupSummary.pm:       $sql .= "truncate(avg(rtt_day),0),";
NagMIN/HostGroupSummary.pm:     $sql .= "truncate(avg(rtt_week),0),";
NagMIN/HostGroupSummary.pm:     $sql .= "truncate(avg(rtt_month),0),";
NagMIN/HostGroupSummary.pm:     $sql .= "truncate(avg(rtt_year),0),";
NagMIN/HostGroupSummary.pm:       $bsql .= "truncate(avg(rtt_day),0),";
NagMIN/HostGroupSummary.pm:     $bsql .= "truncate(avg(rtt_week),0),";
NagMIN/HostGroupSummary.pm:     $bsql .= "truncate(avg(rtt_month),0),";
NagMIN/HostGroupSummary.pm:     $bsql .= "truncate(avg(rtt_year),0),";
NagMIN/HostGroupSummary.pm:       $sql .= "truncate(avg(sla_day),2),";
NagMIN/HostGroupSummary.pm:     $sql .= "truncate(avg(sla_week),2),";
NagMIN/HostGroupSummary.pm:     $sql .= "truncate(avg(sla_month),2),";
NagMIN/HostGroupSummary.pm:     $sql .= "truncate(avg(sla_year),2),";
NagMIN/HostGroupSummary.pm:       $bsql .= "truncate(avg(sla_day),2),";
NagMIN/HostGroupSummary.pm:     $bsql .= "truncate(avg(sla_week),2),";
NagMIN/HostGroupSummary.pm:     $bsql .= "truncate(avg(sla_month),2),";
NagMIN/HostGroupSummary.pm:     $bsql .= "truncate(avg(sla_year),2),";

NagMIN/Host.pm: $sql .= "truncate(rtt_day,1),";
NagMIN/Host.pm: $sql .= "truncate(rtt_week,1),";
NagMIN/Host.pm: $sql .= "truncate(rtt_month,1),";
NagMIN/Host.pm: $sql .= "truncate(rtt_year,1),";
NagMIN/Host.pm: $sql .= "truncate(sla_day,1),";
NagMIN/Host.pm: $sql .= "truncate(sla_week,1),";
NagMIN/Host.pm: $sql .= "truncate(sla_month,1),";
NagMIN/Host.pm: $sql .= "truncate(sla_year,1),";
```

```
chmod 777 /usr/local/nagmin/tmp
```

7.7 Nagios Extended Information Integration

NagMIN's web portal is designed to work with notes_urls in host and service extended information to provide one-click access to network performance information.

7.8 NMAP Port Scanning

NagMIN can perform an NMAP port scan on all or selected Nagios hosts or can scan a network from the centralized or any remote NagMIN server. Output is written to a new PortScan table in the system database so that you have a good inventory of your network.

7.9 Nagios Logo Management

NagMIN makes it easy to see what logos you've assigned to extended host or service information along with an upload and delete feature. Logos uploaded into NagMIN are automatically synchronized with your local and remote Nagios servers.

Note that you really have to upload the logos via the Logo Manager because he does not synchronize into the other direction.

7.10 Nagios Plugin Tester

View all the Nagios plugins, see the help text for configuration options and run the plugins to test your commands.

7.11 Configuration Upload

The NagMIN setup utility will process your existing Nagios configuration files and load them into a NagMIN database.

This is what we have done in chapter “NagMIN Generated Nagios Config”.

7.12 ServiceGroups

Nagios supports multiple hosts per service. NagMIN service groups support multiple services per host. This gives you the ability to group services, like Nagios_statd, local services, etc. into a service group. Then you assign a service group or groups to a host. NagMIN will dynamically create the services for the host when activating a new configuration.

Do not be confused with the Nagios feature of Service Groups introduced in Nagios2. It is the same idea, but not compatible!

7.13 Dynamic Host Creation

NagMIN can create host entries and populate host groups while performing network NMAP scans. NagMIN can learn your network for you.

7.14 Dynamic Service Creation

Protocol and Port fields have been added to the Service Template table that effectively create a NagMIN service table similar to /etc/services. NagMIN dynamically creates service definitions when activating a configuration by matching the PortScan table to ServiceTemplates. You control which services are dynamically created by configuring the ServiceTemplates. NagMIN can dynamically create hundreds or thousands of service definitions.

7.15 Verification Feature

NagMIN has a verification feature where you create a NagMIN set of configuration files that Nagios verification is run against. You perform all of your configuration and verification without affecting your production configuration. You can view all of your NagMIN generated active and verification sets of configuration files to assist with trouble-shooting.

This is one of the features that are working out of the box without you even noticing it. Pressing the “Verify” button is to test your config offline. When no error occurs during verification you can “Activate” the config. The config is copied to nagios/etc and Nagios is restarted.

7.16 Default Data Entry

NagMIN supports default settings for many of the host, command and service template settings. You only need to change a setting if you want to set it different from the default. It also makes it easier for you to easily affect global changes by using the defaults as much as possible.

A pretty nice feature. Have you noticed the * (asterisk) and ~ (tilde) symbols?
The asterisk stands for global default values. That are the values configured in “Module Config”.
The tilde stands for the template default values. That are the values you configured in your templates.
You can use these default values in you object definitions which makes the creation of services much simpler end less time consuming. Of course you can change the default values as necessary.
Practicing working with templates right from the start will save you a lot of time.

7.17 Global Data Updates

NagMIN searches through Nagios tables to update host names, contact names, etc., wherever they occur.

Another fabulous feature. Remember the times crawling through all you text configs because there is still a service left for a host you just deleted?
That will no longer happen to you because NagMIN is doing all the work for you. You just delete a host and NagMIN is deleting the rest like service entries, hostgroup entries etc..

7.18 Multiple Database Support

NagMIN is a modified version of the MySQL WebMIN module. You can create and load multiple databases and activate the data of your choice through NagMIN.

If you worked through this documentation you already know how to do this. Have a look at the chapter “NagMIN Generated Nagios Config”.

7.19 Search Capabilities

NagMIN has all the functionality of the MySQL module including excellent search capabilities.

No searching in endless text configs any more. At the end of every table several searching options are available making your life easier.

7.20 Multi-Lingual Support

NagMIN can be configured to support many languages by editing language files. Right now NagMIN only speaks English. It might be able to make a poor stab at Spanish but will need help to do more.

The English version is just perfect. Who needs a funny translation? OK, nice feature :-)

7.21 Easy Database Editing

NagMIN is configured to use the minimum number of object definitions for Hosts and Services. You may lose some settings if you have an existing Nagios configuration when loading your data into NagMIN. However, simply add a field to the Host or Service table definition, that matches the Nagios name exactly, and NagMIN will automatically detect the field without having to make any other changes to NagMIN.

That is perhaps the way to fix the incompatibility between Nagios 1.* and 2. If I'll have some time within the next days ...

7.22 Reports

NagMIN already has some standard reports that will help troubleshoot and manage Nagios and give you information regarding your network.

7.23 Database Backup

The MySQL module database backup routine. Active/Inactive Object Support - This is not fully implemented yet, but you will be able to set a host, host group, service template, etc., as active or not. If the object is deactivated it will be removed from all configurations. This is difficult since hosts, contacts, etc. are embedded in lists, but not impossible. For now, particularly in the beta, make sure that Active is set to Yes for everything.

8 System start and system services

8.1 Daemon

/etc/init.d/mysql.server
/etc/init.d/nagios
/etc/init.d/webmin

8.2 Cron-Jobs

This is optional and does not have anything to do with NagMIN, but perhaps it is always good to have a safe copy.

- mysqldump (see crontab config and chapter about backup)

9 Troubleshooting

9.1 Checklist

- is MySQL running?
  - No errors in MySQL log? (hostname.log)

- Anything suspicious in /var/log/messages ?
- Anything suspicious in mail? (e.g. info about cron)

10 Backup

10.1 Paths and Files

/usr/local/mysql/mysql.dump
/usr/local/nagmin-2.2.0/
/usr/local/src/

10.2 MySQL

I wrote a little script controled by cron called
/usr/local/mysql/bin/mysql_dump.sh
looking simmilar to

```
mysqldump --password=somepw -K -e --result-file=/tmp/nagios.sql --compatible=mysql40 nagios
mysqldump --password=somepw -K -e --result-file=/tmp/nagmin.sql --compatible=mysql40 nagmin
tar -cvf /usr/local/mysql/mysql_dump.tar /tmp/*.sql
gzip /usr/local/mysql/mysql_dump.tar
rm /tmp/*.sql
```

If you have no company wide backup mechanisms you can use, at least store the backup to another machine for the case of hardware failure.
And of course practice to setup the system using the backed up files !!!

11 ToDo

- Create some patches from the knowledge in this document
- Apply all the fixes described in this paper to the sources.
- Nagios2 support

12 Glossary

13 References

In unsorted manner

- Nagios, Plugins and a list of Nagios related software
  - www.nagios.org
- NagMIN
  - nagmin.sourceforge.net
- Net-SNMP
  - www.net-snmp.org
- MySQL
  - www.mysql.org
- Perl Modules
  - www.cpan.org
- OpenSSH
  - www.openssh.com