

# NagMIN Documentation

## Release 2.0.6

Nagios® Copyright © 1999-2003 - Ethan Galstad

NagMIN® Copyright © 2003 - Frederick Reimers

## Table Of Contents

<b><i>Table Of Contents</i></b> .....	<b><i>1</i></b>
<b><i>Getting Started</i></b> .....	<b><i>2</i></b>
<b><i>NagMIN Installation</i></b> .....	<b><i>3</i></b>
Overview .....	3
Download and Unpack NagMIN Distribution.....	3
Create NagMIN Database .....	3
WebMIN Module Installation.....	4
Initial NagMIN Configuration.....	5
<b><i>Design Strategy</i></b> .....	<b><i>5</i></b>
<b><i>Module Config</i></b> .....	<b><i>6</i></b>
Nagios Configuration Options .....	6
Nagios Service Defaults.....	7
Configurable Options .....	7
<b><i>Inheritance</i></b> .....	<b><i>8</i></b>
<b><i>Notification Aggregator</i></b> .....	<b><i>9</i></b>
Notification Handler (nagmin_notify) .....	9
Notification Transmitter (nagcron_notify.pl) .....	9
Installation.....	9
Command Definitions .....	10
<b><i>NagMIN Navigator</i></b> .....	<b><i>10</i></b>
Installation.....	10
Authentication.....	11
<b><i>NagMIN/Nagios Table Definitions</i></b> .....	<b><i>12</i></b>
Command Table .....	12
Contact Table .....	12
Host Table.....	13

# Getting Started

NagMIN requires an understanding of Nagios and is built upon an existing Nagios installation. You may want to study the Nagios documentation before proceeding if you haven't used it before. <http://www.nagios.org>.

Here are the pre-requisites for taking full advantage of NagMIN's functionality.

1. NagMIN has been developed on RedHat 7.3. It's assumed that it will work on other platforms but not guaranteed.
2. NagMIN must be built upon an existing installation of Nagios 1.0 or 1.1. To get the full benefit of NagMIN, Nagios needs to be built with MySQL database support, except for extended information. You can download Nagios and Nagios plugins from the [NagMIN project page](#). This will ensure that the version you're downloading is supported by NagMIN.
3. Install the latest version of Nmap from <http://www.nmap.org>. Version 3.20+ of Nmap is much faster than previous versions.
4. Install the latest version of WebMIN from <http://www.webmin.com>.
5. Install version 3.23 or later of MySQL from <http://www.mysql.com>.
6. Install the latest version of ucd-snmp if you're going to use the Network Traffic Statistics feature
7. Install the latest version of RRDtool for Traffic Statistics and the Web Portal from <http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/>.

# NagMIN Installation

First read the [Getting Started](#) section to install all the pre-requisite applications.

Come back here for information on installing NagMIN after taking care of the pre-requisites.

## Overview

This installation documentation gives examples assuming that you've installed Nagios in the default `/usr/local/nagios` directory and that WebMIN is also installed off of `/usr/local/webmin-1.nnn`. NagMIN supports the placement of Nagios and NagMIN directories in alternate locations.

NagMIN has a number of plugins and associated utilities that you may or may not choose to implement. Basic NagMIN consists of a WebMIN module, MySQL databases for each local and/or remote Nagios server that you're supporting and a NagMIN working directory, `/usr/local/nagmin` on the central server. NagMIN creates a configuration directory for each database off a `/usr/local/nagmin` where it stores a set of configuration files used for verification and troubleshooting.

The following sections will take you through a standard NagMIN installation, and then through the installation of additional features.

## Download and Unpack NagMIN Distribution

1. Download NagMIN from <http://sourceforge.net/projects/nagmin> into `/usr/local`.
2. `cd /usr/local`
3. `tar -zxvf nagmin-n.n.n.tar.gz`
4. A NagMIN distribution directory, `/usr/local/nagmin-n.n.n`, is created when you untar the distribution for each version.

## Create NagMIN Database

An installation scripts has been provided to help you create your first NagMIN database called **setup**. Setup pulls in your current Nagios template configuration information and creates a database from it. Setup does not support the default (Netsaint) configuration method. Setup does this by reading your `nagios.cfg` and `cgi.cfg` files and looking for `cfg=` and extended information configuration file commands. It then reads each of these files and creates a file of MySQL statements to create the database and load information into it. First, it copies the **nagmin\_init.sql** file that contains all of the table creation commands into **nagmin\_load.sql** and then appends INSERT statements for each Nagios definition from the configuration files.

1. Define a MySQL user account with full access to your NagMIN databases.
2. You can use the WebMIN MySQL module to add the user, through User Permissions. Make sure you give this user full access to all NagMIN databases. This is required if you will be using NagMIN to manage multiple databases.

3. Run setup - /usr/local/nagmin-n.n.n/setup. The routine is self-documenting. Take your time, read the instructions.

Setup will:

- ?? Create required directories
- ?? Create a NagMIN database
- ?? Load your existing configuration into the database
- ?? Create a verification version of nagios.cfg

Please Note! You may lose some host and service configuration information since NagMIN puts as many objects in templates as possible

## WebMIN Module Installation

The NagMIN WebMIN module is contained in the distribution directory. It is a tar file but the name is nagmin-n.n.n-wbm to match the WebMIN standard. Here are the steps to install:

1. Log in to WebMIN with an administrator account
2. Click on the WebMIN icon on the upper left-hand corner
3. Click on WebMIN Configuration
4. Click on WebMIN Modules
5. Select ***Install Module from a local file***
6. Browse for the NagMIN WebMIN module ( /usr/local/nagmin/nagmin-n.n.n.wbm)
7. Install

## Initial NagMIN Configuration

This section will help you with the initial NagMIN configuration. NagMIN is accessed from the WebMIN Server tab. You'll see a NagMIN logo listed as NagMIN Network Monitoring.

1. You'll be asked for a username and password to access MySQL the first time you access it. Enter the username/password that you used to create the database(s) with setup.
2. Click on **Module Config**, a link at the top left of the screen to set Global NagMIN Configuration options. The first field, Databases, is important. Make sure that you have the name(s) of your NagMIN databases listed as space-separated entries. NagMIN will only display databases listed in this field. Once you've set NagMIN options, save the configuration. Module Config needs to be reviewed and saved each time you update to set new options that may have been added in the current release.
3. You now have to define the system that each database is going to be used for. Click on a database from the System screen, and then click on the System table at the bottom of the center column. Add one entry to the table selected the host name describing the location of the Nagios server and its directories.

## Design Strategy

The design strategy for Nagios is make extensive use of Nagios templates, along with NagMIN utilities and features to dynamically create service, escalation and extended information entries to make it easier to implement enhanced Nagios functionality.

NagMIN is designed to support all Nagios object definitions, but in a slightly different way. All possible directives are not contained in the primary definition tables for Hosts, Services, etc. NagMIN has "template" tables for each definition that contain all options. The most common directives and additional NagMIN directives are contained in the primary tables to simplify host and service configuration by minimizing the number of fields configured. This may be different than the way you are currently configuring Nagios. Making better use of templates may ease your configuration burden but may also take some time to implement.

NagMIN supports Nagios object inheritance, to a point, and also adds a global default layer. Inheritance is supported for a limited number of fields in the host, service and contact tables. NagMIN does not support a registered definition also acting as a template and possibly being also linked to another template.

NagMIN displays global default information by prefixing the value with an asterisk \* when displaying objects. Settings taken from templates are prefixed with a tilde ~. As a result, you can easily view the settings for any object that have a default entry. NagMIN also has popup windows for displaying the template information so that you can view the template while modifying an entry.

Additional NagMIN tables have been created to support centralized management and Nagios pre-processing. Pseudo fields have also been added to the Host and ServiceTemplate tables to support the dynamic creation of service and extended host information definitions.

- ?? System Table
  - o Host Name
  - o Address
  - o Alias
- ?? PortScan Table
  - o Host Name
  - o Port
  - o Protocol
  - o Service
- ?? Service Group Table
  - o Service Group Name
  - o Service Templates
- ?? Pseudo Host Table Fields
  - o Extended Host Information
  - o Service Group
  - o Port Scan (Enabled/Disabled)
  - o Active (Enabled/Disabled)
- ?? Pseudo ServiceTemplate Table Fields
  - o Port
  - o Protocol

The Extended Host Information table is also strictly a template file now. NagMIN creates the extended host configuration entry for each host by matching the extended host information field in the host table with an entry in the extended host information table.

NagMIN does not load the `cgi.cfg`, `nagios.cfg` or the `resources.cfg` files into the database since they aren't template based. You can manually edit these files from the web interface as well as view and edit all other configuration files.

## Module Config

Module Config is a feature of WebMIN. NagMIN uses Module Config to store Global Default Information as well as necessary directory information and processing controls. Screen shots of all global default information and configuration options are included below, however, descriptions have not been updated for Version 2.0.0.

## Nagios Configuration Options

These define the working environment for NagMIN. Defaults are set based on the operating system that WebMIN detects from static files. There's no attempt to try to determine your configuration or go looking for where you may have installed Nagios or WebMIN. You will probably have to modify these options from time-to-time.

- ?? Nagios Databases - list all databases to be displayed by NagMIN in the server database section
- ?? Nagios Init File - it's required that you have a startup Nagios init file installed
- ?? Nagios Root Directory - Where's Nagios?
- ?? WebMIN Root Directory - Where's WebMIN
- ?? NagMIN Root Directory - the nagmin directory created when you first untarred the NagMIN distribution is used by NagMIN to store copies of all server configuration files. These files are used for verify your configuration without affecting the running configuration files or having to push files to remote servers while editing and cleaning up your configuration.
- ?? Process NagMIN Port Scans - NagMIN will dynamically create service entries by matching port scan data with service template entries that have matching port and protocol numbers. You can enable or disable port scan processing.
- ?? Process NagMIN Service Groups - NagMIN will dynamically create service entries from groups of services defined in the NagMIN service group table when the service group has been added to a host. You can enable or disable service group processing.
- ?? Create Host Dependencies From Parents - NagMIN will dynamically create host dependencies from parent object lists for hosts. You can enable or disable this processing.
- ?? Enable Global Field Updating - NagMIN modifies host name, contact name, etc., entries that appear in other tables in parent, member, host and contact list objects. You can enable or disable this processing.

## Nagios Service Defaults

NagMIN has added a Global Default feature where entries and option selections made here will be used as the default entries for these values unless overridden in a template or object definition. NagMIN displays the default values prefixed with an \* asterisk when displaying table information.

Please refer to the Nagios documentation for definitions of each of these objects.

## Configurable Options

These options relate to the configuration of MySQL.

# Inheritance

NagMIN adds an additional layer of inheritance for Nagios. The layers are:

1. Nagios defaults (nagios.cfg, cgi.cfg, hard-coded)
2. NagMIN defaults (Module Config)
3. Nagios template definitions
4. Nagios defaults can be found in Nagios documentation.

Module Config sets default entries for most Nagios objects. NagMIN lets you "see-through" field definitions to view where the object configuration will come from. An asterisk (\*) next to a field entry means that NagMIN will use the default information from Module Config.

NagMIN encourages the use of Nagios object templates. Separate template tables have been created for each object definition that supports templates. NagMIN lets you "see-through" field definitions to view where the object configuration will come from. A tilde (~) next to a field entry means that NagMIN will use information from the associated object template.



# Notification Aggregator

NagMIN Notification Aggregator helps solve the problem of getting blasts of notification emails and pages whenever there is a major network outage, or if the Nagios server loses network connectivity temporarily. This makes the receipt of messages more meaningful, reduces operational costs for pagers by reducing messages. Use of the aggregator along with host and service escalations will help control notification to the appropriate personnel.

There are two NagMIN scripts that comprise the aggregation process as described in the next sections.

## Notification Handler (nagmin\_notify)

The notification handler is a command script that replaces the `host_notification_commands` and `service_notification_commands` in the Contact and Contact Template tables. It takes Nagios notification arguments and places the Nagios notification information in the Notifications table of the NagMIN database.

The "plugin" is not written as a standard Nagios plugin since the arguments are not variable. Commands are defined to handle the four notification variations, HOST/EMAIL, HOST/PAGER, SERVICE/EMAIL, SERVICE/PAGER. A table showing the Command Line entries required is included in the Installation section.

## Notification Transmitter (nagcron\_notify.pl)

The notification transmitter aggregates messages by sorting alerts according to contacts, whether it's an email or pager message and then by host. It also checks the current status of the host or service to check one more time to make sure that the host or service is not UP or OK. The transmitter will suppress the message if it has recovered or it will return the current status and output before sending the notifications. The NagMIN Notifications table is updated with a status of whether the notification was sent or suppressed.

## Installation

The Handler and Notification scripts are contained in the `/sbin` sub-directory of the distribution directory.

1. Copy `/sbin/nagmin_notify` to your Nagios libexec directory (`/usr/local/nagios/libexec`). Edit the top of the file with your database information.
2. Copy `/sbin/nagcron_notify.pl` to wherever you like to run cron jobs from. The recommendation is to create an `/sbin` directory of your NagMIN directory (`/usr/local/nagmin/sbin`). Edit the top of the file with your database information.
3. Create a cron job for `nagcron_notify.pl` using WebMIN. The recommendation is to run the transmitter the same as your check interval. Also, run the job as 'nagios' so that the from address in the email will be `nagios@yourserver.com`.
4. Add notification commands for the notification handler to the Command table. The command lines need to be put in exactly as shown in the table below. Once you've added the commands, make changes to `host_notification_commands` and `service_notification_commands` in Contacts and Contact Templates as necessary.

## Command Definitions

Command	Command Line
nagmin_host_notify_email	/usr/local/nagios/libexec/nagmin_notify HOST EMAIL "\$CONTACTNAME\$" "\$HOSTNAME\$" "\$HOSTSTATES\$" "\$OUTPUT\$" "\$NOTIFICATIONTYPE\$" "\$NOTIFICATIONNUMBER\$" "\$TIMET\$"
nagmin_host_notify_pager	/usr/local/nagios/libexec/nagmin_notify HOST PAGER "\$CONTACTNAME\$" "\$HOSTNAME\$" "\$HOSTSTATES\$" "\$OUTPUT\$" "\$NOTIFICATIONTYPE\$" "\$NOTIFICATIONNUMBER\$" "\$TIMET\$"
nagmin_service_notify_email	/usr/local/nagios/libexec/nagmin_notify SERVICE EMAIL "\$CONTACTNAME\$" "\$HOSTNAME\$" "\$HOSTSTATES\$" "\$OUTPUT\$" "\$NOTIFICATIONTYPE\$" "\$NOTIFICATIONNUMBER\$" "\$TIMET\$" "\$SERVICEDESC\$" "\$SERVICESTATES\$"
nagmin_service_notify_pager	/usr/local/nagios/libexec/nagmin_notify SERVICE PAGER "\$CONTACTNAME\$" "\$HOSTNAME\$" "\$HOSTSTATES\$" "\$OUTPUT\$" "\$NOTIFICATIONTYPE\$" "\$NOTIFICATIONNUMBER\$" "\$TIMET\$" "\$SERVICEDESC\$" "\$SERVICESTATES\$"

## NagMIN Navigator

NagMIN Navigator is a replacement for the Nagios "side.html" script that expands navigation capabilities to include selections for individual host groups, and the capability to add more menu options.

NagMIN Navigator is a perl script that generates javascript code based on the SourceForge Struts-Menu project ( <http://struts-menu.sourceforge.net/> ).

Navigator is the first NagMIN module that will use the "Level"

### Installation

The source code for NagMIN Navigator is in the distribution directory "/usr/local/nagmin-2.0.6/extras/side". The Navigator script is "nagmin\_side.pl". There are also 3 image files, square.gif, plus.gif and minus.gif.

Here's an installation scenario for a standard Nagios installation with Apache 1.3.

1. Copy nagmin\_side.pl to /usr/local/nagios/share/
2. Copy \*.gif files to /usr/local/nagios/share/images
3. Set owner and permissions on copied files
4. Edit nagmin\_side.pl and make changes to the \$logo, \$cgibin and %info database variables as required for your installation.

5. Modify your web server configuration to process .pl files with mod\_perl in the /usr/local/nagios/share directory.

mod\_perl 1.0

```
<Files ~ "\.(pl)$">  
  SetHandler perl-script  
  PerlHandler Apache::Registry  
</Files>
```

mod\_perl 2.0

```
<Files ~ "\.(pl)$">  
  SetHandler perl-script  
  PerlHandler ModPerl::Registry  
</Files>
```

6. Edit Nagios index.html ( /usr/local/nagios/share/index.html). Change the "side.html" entry to "nagmin\_side.pl"

## Authentication

Navigator is the first NagMIN module that uses the "Level" field in the Contact table. The functionality mirrors Nagios authentication security controlling which host groups will be displayed for a user.

Setting the Level of a Contact to "User" will only present the user with a view of Host Groups that have an associated Contact Group of which they are a member. Any other level will display all host groups.

# NagMIN/Nagios Table Definitions

The following definitions are a combination of Nagios command definitions, taken from the Nagios documentation, along with definitions of additional fields added by NagMIN.

## Command Table

A command definition is just that. It defines a command. Commands that can be defined include service checks, service notifications, service event handlers, host checks, host notifications, and host event handlers. Command definitions can contain macros, but you must make sure that you include only those macros that are "valid" for the circumstances when the command will be used. More information on what macros are available and when they are "valid" can be found here. The different arguments to a command definition are outlined below.

### Directive Descriptions

**Command Name** - This directive is the short name used to identify the command. It is referenced in contact, host, and service definitions, among other places.

**Command Line** - This directive is used to define what is actually executed by Nagios when the command is used for service or host checks, notifications, or event handlers. Before the command line is executed, all valid macros are replaced with their respective values. See the documentation on macros for determining when you can use different macros. Note that the command line is not surrounded in quotes. Also, if you want to pass a dollar sign (\$) on the command line, you have to escape it with another dollar sign.

**Comment** - A configuration comment that is also written as a comment line to the Command configuration file.

## Contact Table

A contact definition is used to identify someone who should be contacted in the event of a problem on your network. The different arguments to a contact definition are described below.

### Directive Descriptions (\* denotes NagMIN fields)

**\*Template (Use)** - This is really the Nagios **Use** directive. NagMIN displays information from the selected Contact Template in corresponding fields by adding a tilde(~) prefix. Please read the Inheritance section for more information on NagMIN defaults.

**Contact Name** - This directive is used to define a short name used to identify the contact. It is referenced in contact group definitions. Under the right circumstances, the \$CONTACTNAME\$ macro will contain this value.

**Alias** - This directive is used to define a longer name or description for the contact. Under the right circumstances, the \$CONTACTALIAS\$ macro will contain this value.

**Email** - This directive is used to define an email address for the contact. Depending on how you configure your notification commands, it can be used to send out an alert email to the contact. Under the right circumstances, the `$CONTACTEMAIL$` macro will contain this value.

**Pager** - This directive is used to define a pager number for the contact. It can also be an email address to a pager gateway (i.e. `pagejoe@pagenet.com`). Depending on how you configure your notification commands, it can be used to send out an alert page to the contact. Under the right circumstances, the `$CONTACTPAGER$` macro will contain this value.

**Service Notification Period** - This directive is used to specify the short name of the time period during which the contact can be notified about service problems or recoveries. You can think of this as an "on call" time for service notifications for the contact. Read the documentation on time periods for more information on how this works and potential problems that may result from improper use.

**Host Notification Period** - This directive is used to specify the short name of the time period during which the contact can be notified about host problems or recoveries. You can think of this as an "on call" time for host notifications for the contact. Read the documentation on time periods for more information on how this works and potential problems that may result from improper use.

**Host Notification Options** - This directive is used to define the host states for which notifications can be sent out to this contact. Valid options are a combination of one or more of the following: `d` = notify on DOWN host states, `u` = notify on UNREACHABLE host states, and `r` = notify on host recoveries (UP states). If you specify `n` (none) as an option, the contact will not receive any type of host notifications.

**Service Notification Options** - This directive is used to define the service states for which notifications can be sent out to this contact. Valid options are a combination of one or more of the following: `w` = notify on WARNING service states, `u` = notify on UNKNOWN service states, `c` = notify on CRITICAL service states, and `r` = notify on service recoveries (OK states). If you specify `n` (none) as an option, the contact will not receive any type of service notifications.

**Service Notification Commands** - This directive is used to define a list of the short names of the commands used to notify the contact of a service problem or recovery. Multiple notification commands should be separated by commas. All notification commands are executed when the contact needs to be notified. The maximum amount of time that a notification command can run is controlled by the `notification_timeout` option.

**Host Notification Commands** - This directive is used to define a list of the short names of the commands used to notify the contact of a host problem or recovery. Multiple notification commands should be separated by commas. All notification commands are executed when the contact needs to be notified. The maximum amount of time that a notification command can run is controlled by the `notification_timeout` option.

**\*Comment** - A NagMIN comment that is also written as a comment line to the configuration file

## Host Table

A host definition is used to define a physical server, workstation, device, etc. that resides on your network. NagMIN has added fields to the Host Table to help manage and control NagMIN

enhanced features. NagMIN does not support all possible Host directives in the Host table to keep the table manageable and encourage the use of Host Templates for fine tuning Nagios.

## Directive Descriptions (\* denotes NagMIN fields)

**\*Template (Use)** - This is really the Nagios **Use** directive. NagMIN displays information from the selected Host Template in corresponding fields by adding a tilde(~) prefix. Please read the Inheritance section for more information on NagMIN defaults.

**Host Name** - This directive is used to define a short name used to identify the host. It is used in host group and service definitions to reference this particular host. Hosts can have multiple services (which are monitored) associated with them. When used properly, the \$HOSTNAME\$ macro will contain this short name.

**Alias** - This directive is used to define a longer name or description used to identify the host. It is provided in order to allow you to more easily identify a particular host. When used properly, the \$HOSTALIAS\$ macro will contain this alias/description.

**Address** - This directive is used to define the address of the host. Normally, this is an IP address, although it could really be anything you want (so long as it can be used to check the status of the host). You can use a FQDN to identify the host instead of an IP address, but if DNS services are not available this could cause problems. When used properly, the \$HOSTADDRESS\$ macro will contain this address. Note: If you do not specify an address directive in a host definition, the name of the host will be used as its address. A word of caution about doing this, however - if DNS fails, most of your service checks will fail because the plugins will be unable to resolve the host name.

**Parents** - This directive is used to define a comma-delimited list of short names of the "parent" hosts for this particular host. Parent hosts are typically routers, switches, firewalls, etc. that lie between the monitoring host and a remote hosts. A router, switch, etc. which is closest to the remote host is considered to be that host's "parent". Read the "Determining Status and Reachability of Network Hosts" document located here for more information. If this host is on the same network segment as the host doing the monitoring (without any intermediate routers, etc.) the host is considered to be on the local network and will not have a parent host. Leave this value blank if the host does not have a parent host (i.e. it is on the same segment as the Nagios host). The order in which you specify parent hosts has no effect on how things are monitored.

**\*Host Ext Info** - NagMIN adds the ability to select a Host Extended Information Template when configuring the host record.

**\*Contact Groups** - NagMIN uses the Contact Groups selected in a Host definition when dynamically creating service definitions from Port Scan information in the Port Scan table.

**\*Service Groups** - NagMIN adds the ability to select a Service Groups from the NagMIN Service Group table to dynamically create service definitions for each service template contained in the Service Group. This provides the ability to "copy" a group of services for multiple hosts that have common services defined.

**\*Port Scan** - NagMIN adds the ability to dynamically create service definitions from Port Scan information. The Port Scan field in the Host table gives you the ability to enable or disable this feature for specific hosts.

**Check Command** - This directive is used to specify the short name of the command that should be used to check if the host is up or down. Typically, this command would try and ping the host to see if it is "alive". The command must return a status of OK (0) or Nagios will assume the host is down. If you leave this argument blank, the host will not be checked - Nagios will always assume the host is up. This is useful if you are monitoring printers or other devices that are frequently turned off. The maximum amount of time that the notification command can run is controlled by the `host_check_timeout` option.

**Max Check Attempts** - This directive is used to define the number of times that Nagios will retry the host check command if it returns any state other than an OK state. Setting this value to 1 will cause Nagios to generate an alert without retrying the host check again. Note: If you do not want to check the status of the host, you must still set this to a minimum value of 1. To bypass the host check, just leave the `check_command` option blank.

**Notification Interval** - This directive is used to define the number of "time units" to wait before re-notifying a contact that this server is still down or unreachable. Unless you've changed the `interval_length` directive from the default value of 60, this number will mean minutes. If you set this value to 0, Nagios will not re-notify contacts about problems for this host - only one problem notification will be sent out.

**Notification Period** - This directive is used to specify the short name of the time period during which notifications of events for this host can be sent out to contacts. If a host goes down, becomes unreachable, or recovers during a time which is not covered by the time period, no notifications will be sent out.

**Host Notifications Options** - This directive is used to determine when notifications for the host should be sent out. Valid options are a combination of one or more of the following: d = send notifications on a DOWN state, u = send notifications on an UNREACHABLE state, and r = send notifications on recoveries (OK state). If you specify n (none) as an option, no host notifications will be sent out. Example: If you specify d,r in this field, notifications will only be sent out when the host goes DOWN and when it recovers from a DOWN state.

**\*SNMP Community String** - NagMIN uses the SNMP Community String with NagMIN plugins like "nm\_check\_interface" which stores SNMP interface information in RRD databases files which are used by the NagMIN Web Portal to create MRTG-like graphs and much more.

**\*Comment** - A NagMIN comment that is also written as a comment line to the configuration file

# NagMIN Release Notes

---

Version: 2.0.4  
Date: 10/13/2003

---

## Feature / Bug Fix Release

NagMIN Notification Aggregator has been added to this release. This consists of a plugin (`nagmin_notify`) to take notification messages and place them in the Notifications table of the NagMIN database. A new cron utility, (`nagcron_notify.pl`), aggregates the messages by sorting alerts according to contact, then whether it's an email or pager message. It checks for the current status of the host or service and either suppresses the message if it's recovered or returns the most current status and output before sending the emails or pages. The Notifications table is updated with a status of whether the notification was sent or not. The notification utilities are in the `/sbin` directory of the distribution.

## Bug Fixes -

Web Portal - There were a number of bugs in the last release related to new features.

Problems also existed with the setup script if you were adding host and service escalations. In particular, the Web Portal was not able to handle URL aliases (`www.sanctuary.net.org/nagmin/`) didn't work. This has been fixed.

Multiple selection fields with also a default choice allowed you to select more than one but didn't highlight those choices the next time you edited the record. This has been fixed.

There were a number of Module Config option fields missing from the `config.info` file included in 2.0.3. This has been corrected in 2.0.4.

## Notification Implementation Notes

- Copy `/sbin/nagmin_notify` to your Nagios libexec directory. Edit the top of the file with your database information.

- Copy `/sbin/nagcron_notify.pl` to your NagMIN sbin directory, edit the top of the file with your database information and then create a cron job for it.

- Here's a sample of the notification commands. They command values and options

need to be put in exactly as shown. The name of the command can change. Make changes

to your contact templates as necessary.

`nagmin_host_notify_email`



```
/usr/local/nagios/libexec/nagmin_notify HOST EMAIL "$CONTACTNAME$"
"$HOSTNAME$" "$HOSTSTATE$" "$OUTPUT$" "$NOTIFICATIONTYPE$"
"$NOTIFICATIONNUMBER$" "$TIMET$"
```

nagmin\_host\_notify\_pager

```
/usr/local/nagios/libexec/nagmin_notify HOST PAGER "$CONTACTNAME$"
"$HOSTNAME$" "$HOSTSTATE$" "$OUTPUT$" "$NOTIFICATIONTYPE$"
"$NOTIFICATIONNUMBER$" "$TIMET$"
```

nagmin\_service\_notify\_email

```
/usr/local/nagios/libexec/nagmin_notify SERVICE EMAIL "$CONTACTNAME$"
"$HOSTNAME$" "$HOSTSTATE$" "$OUTPUT$" "$NOTIFICATIONTYPE$"
"$NOTIFICATIONNUMBER$" "$TIMET$" "$SERVICEDESC$" "$SERVICESTATE$"
```

nagmin\_service\_notify\_pager

```
/usr/local/nagios/libexec/nagmin_notify SERVICE PAGER "$CONTACTNAME$"
"$HOSTNAME$" "$HOSTSTATE$" "$OUTPUT$" "$NOTIFICATIONTYPE$"
"$NOTIFICATIONNUMBER$" "$TIMET$" "$SERVICEDESC$" "$SERVICESTATE$"
```

```
-----
Version:    2.0.3
Date:      10/16/2003
-----
```

Feature / Bug Fix Release

#### New Features

- Service Escalations can now be dynamically generated from Host Escalations.  
NagMIN will go through all manual and dynamically generated services and create Service Escalations. NagMIN will first look for specific Host or Host Group entries and then for a host wildcard entry(\*) in Host Escalations to generate the service escalation.
- HostGroup support has been added to the the Host and Service Escalation tables.
- A Notifications table is added in this release, but not used yet. NagMIN will have a Notification aggregation feature in the next release.

#### Bug Fixes

- Web Portal - the current script only works off of a base url. The fix will work with http://fqdn/nagmin/ or whatever.
- build-nagios - the Nagios build script had a misspelling for adding mysql-comment support. The Nagios configure script doesn't return an error.

## Notes

- The Web Portal requires Mod\_Perl 1.0 Apache::Registry or Mod\_Perl 2.0 ModPerl::Registry for the perl script to work. Here's my Apache 1.3 conf.

```
Alias /nagmin/ /usr/local/nagmin1/share/
<Directory "/usr/local/nagmin1/share/">
  <Files ~ "\.(pl)$">
    SetHandler perl-script
    PerlHandler Apache::Registry
  </Files>
  AllowOverride AuthConfig
  Options ExecCGI
  AuthName NagMIN
  AuthType Basic
  Require valid-user
  Order allow,deny
  Allow from all
</Directory>
```

- RRD uses two directories /usr/local/nagmin/rrd/db and /usr/local/nagmin/rrd/graphs  
You'll need to manually set permissions on these directories for your web server to be able to access. The portal usually shows the permissions errors.

- nagcron.pl - creates the db directory and generates all of the .rrd dbs for  
hosts the first time it runs. This can take a 30 seconds or more  
depending on the number of hosts. Please don't stop it.

- nagcron.pl has to run for the traffic statistics to be generated from check\_ping.

---

Version: 2.0.2  
Date: 10/14/2003

---

### Feature / Bug Fix Release

Some minor problems corrected in the language file for missing text fields for the SNMP Scan Page, and, the Nagios DB Support option was missing from Module Config to turn on Nagios support.

Help text is starting to become available. Click Help on the main page for help on that screen. Clicking Help from the NagMIN Table Edit page will display help on the table selected. Host, Host Template, Command and Contact tables are completed so far.

---

Version: 2.0.1  
Date: 10/11/2003

---

MANDATORY UPDATE FOR 2.0.0!!

Feature / Bug Fix Release (Beta)

2.0.0 was only available for a short time, however, if you're one of the few, there are some major problems with installing NagMIN for the first time using version 2.0.0. Updating was OK.

If you didn't. Please read all of the 2.0.1 and 2.0.0 release notes for information on what's change and how to update. The nagmin.sourceforge.net site may have some helpful information on installation and configuration although documentation is still lacking.

NOTE: Make sure that you apply the sql updates for 2.0.0 and 2.0.1.

NOTE: Make sure that you carefully go through Module Config and set all options.

NOTE: All previous and current release notes have been placed in a "release\_notes" directory in the distribution directory.

NOTE: Documentation will start appearing at nagmin.sourceforge.net in the near future.

-----  
Feature Enhancements (2.0.1)  
-----

- Port Scanning has been modified and improved. There are two new fields,  
Active and Lock Port. Toggling 'Active' will control whether or not NagMIN creates a service definition for the port. "Lock Port" controls whether subsequent port scans delete and/or rewrite the port entry.  
  
Network scanning has been changed. It no longer deletes host entries before rescanning. It will simply ignore any existing entries so that customized settings will be preserved.
- Reload and Force Reload buttons are added to the Nagios Control Bar.
- NagMIN will stop you from activating a configuration before you've defined a System table entry.
- All Host, Alias and Address fields have been expanded to 80 characters to support the use of FQDNs.
- All feature requests reported to SourceForge have been either implemented or responded to.

-----  
BUG FIXES ( 2.0.1 )

-----  
-----  
- All bugs reported to SourceForge have been fixed or responded to.  
Thanks  
to all who took the time to let me know.

- Service and Host Dependencies had major problems. This has been corrected.

- All

-----  
Version: 2.0.0  
Date: 10/9/2003  
-----

-----  
Major Feature / Bug Fix Release (Beta)  
-----

This is a major release of NagMIN. I'm releasing it now, since it has been quite a while since the last release and there are a number of bug fixes.

Documentation is still almost non-existent, you're on your own. I'm going to provide some help in this document for implementing the new features that you'll have to translate for your own environment.

I'm sure that there are instructions missing, but it's fairly intuitive.

I'll be responding quicker to help requests on SourceForge and have some volunteers to help with documentation.

-----  
Feature Enhancements (2.0.0)  
-----

Version 2.0.0 of NagMIN adds a number of major features.

- Nagios MySQL Database Support - this is done by building Nagios with MySQL support, adding the Nagios MySQL tables to the NagMIN database and pointing Nagios to it.

- SNMP Interface Scan

NagMIN will scan a hosts interfaces via SNMP MIB II, learn the available interfaces, and store the interface description, index and speed in a new Interfaces table. NagMIN dynamically creates services using a check interface service template that is configured for the new NagMIN

nm\_check\_interface plugin. The scan also stores community strings for hosts in the Host table.

- SNMP MIB Walker

This is a simple popup that allows you to select a SNMP enabled host and view MIB II information.

- Host Response Time Statistics Collection (PING/RRDtool)

NagMIN includes a cron script to be run every 5 minutes that searches the Nagios service status table for PING host services, parses the output from 'check\_ping' and updates the hosts RRD database table.

- Network Traffic Statistics Collection (SNMP/RRDtool)

NagMIN includes a network traffic plugin (nm\_check\_interface) that polls traffic information via SNMP from a learned interface, updating its RRD database and returning status based on warning and critical thresholds.

- NagMIN Traffic Statistics Web Portal

The Web Portal generates RRD Graphs for individual hosts RTT and availability, and, interface traffic statistics for SNMP enabled hosts. The portal pulls host group and host information from the NagMIN database and host/service status information from the Nagios database tables.

- NagMIN Host/Traffic Statistics Notes URL Integration

NagMIN integrates its web portal with Notes\_URL Extended Information by dynamically generating the correct notes\_url entries to view information about specific hosts or interface service checks.

- Host and Service Alert Event Handler

A NagMIN event handler is included in 2.0.0 that is implemented best as a global event handler. It will log alert information to two new NagMIN tables, HostAlerts and ServiceAlerts. NagMIN will be enhanced in the future to provide standard reports for event information and correlation with traffic statistics.

- NagMIN Stylesheets and customized Nagios html files

These are contained in /usr/local/nagmin-2.0.0/nagstyle.tar

-----  
-----  
BUG FIXES ( 2.0.0 )  
-----  
-----

There are a lot of bug fixes. Probably too many to detail, but definitely too many to remember right now. The major fixes are:

- Global Field Updating - host name and host group name changes should properly

update in all tables now, including deleting a host or host group from member, host and host group lists.

- Extended Information - there were a number of problems with NagMIN not showing thumbnails of the image files. Also, template and default information overrode specific information changed in the HostExtInfo table.
- Command popup display didn't work if arguments were added. This is fixed.
- Selecting multiple host groups in Service Escalations is now enabled.

-----  
-----  
UPGRADING  
-----  
-----

1 - Update NagMIN databases

NOTE:: Please make sure that you backup your database before applying these changes.

Recommendation: Apply the update to a test database first.

The sql update file is /usr/local/nagmin-2.0.0/update/update-2.0.0.sql

Make sure that you apply all previous sql updates before applying this one.

```
mysql -ppassword 'database' < update-2.0.0.sql
```

2 - Build Nagios Tables (Optional)

Nagios' DB schema is included in the distribution. Enable Nagios DB Support in module\_config after implementing Nagios MySQL support.

```
mysql -ppassword 'database' < /usr/local/nagios-1.0/contrib/database/create_mysql
```

3 - Update the NagMIN WebMIN module

Update the NagMIN module in WebMIN using /usr/local/nagmin-2.0.0/nagmin-2.0.0.wbm

4 - Access NagMIN, then module config and check the configuration. There are a number of new options for the SNMP that must be configured. Defaults are included and must be changed if your implementation is different.

## 5 - Configure the web portal (Optional)

The portal script is `/usr/local/nagmin-2.0.0/sbin/nagmin.pl`. Edit the top of the file to match your system. I created a share and image directory off of

`/usr/local/nagmin` to make the Apache configuration pretty much the same as Nagios.

Then copy the portal script into share and the images directory to the share directory.

```
mkdir /usr/local/nagmin/share
```

```
cp /usr/local/nagmin-2.0.0/sbin/nagmin.pl  
/usr/local/nagmin/share/index.pl
```

```
cp -r /usr/local/nagmin-2.0.0/images /usr/local/nagmin/share
```

Modify permissions and owner/group information as required. There's a

`module_config` option that must be set to the same `user:group` setting that you use.

Modify Apache or whatever web server you're using as needed.

## 6 - Install ucd-snmp (Optional)

2.0.0 was developed using `ucd-snmp-4.2.5-7.73.0.i386.rpm` for RedHat 7.3.

Check `module_config` and set the proper location of the snmp executables.

## 7 - Install RRDtool (Optional)

2.0.0 was developed using `rrdtool-1.0.45`.

Check `module_config` and set the proper location of the rrdtool executables and perl libraries.

## 8 - Build Nagios with MySQL support. (Optional)

NagMIN requires Nagios to be built with MySQL support for everything except extended information. Make sure to follow the Nagios documentation to

modify `nagios.cfg` and `cgi.cfg` for database support.

The script `/usr/local/nagmin-2.0.0/build-nagios` is included below detailing NagMIN requirements:

```
cd /usr/local/nagios-1.0  
make clean  
CPPFLAGS="-I/usr/include -I/usr/include/mysql"
```

```

LDFLAGS="-L/usr/lib -L/usr/lib/mysql -L/lib"
./configure --prefix=/usr/local/nagios \
  --with-mysql-comments \
  --with-mysql-downtime \
  --with-mysql-retention \
  --with-mysql-status \
  --with-mysql-inc=/usr/include/mysql \
  --with-mysql-lib=/usr/lib/mysql
make all
#make install
# make install-init
# make install-commandmode
# make install-config

```

#### 9 - Create a Command for the SNMP Interface Check Plugin (Optional)

First copy the plugin `/usr/local/nagmin-2.0.0/sbin/nm_check_interface` to `/usr/local/nagios/libexec`.

Create a command in the Command table. This command must correspond to the

SNMP Interface Check Command entry in `module_config`.

```

check_interface =
  $USER1$/nm_check_interface -H $HOSTNAME$ -I $HOSTADDRESS$ -i $ARG1$ -
s $ARG2$ -C $ARG3$ -w $ARG4$ -c $ARG5$

```

#### 10 - Create a ServiceTemplate for SNMP Interface Checks (Optional)

This is the service template that NagMIN uses to dynamically create interface service

checks for interfaces in the Interfaces table after performing an SNMP Interface Scan of selected hosts.

The name of the service template must correspond to the SNMP Interface Service Template entry in `module_config`.

The template DOES NOT require entering a 'Check Command' entry. NagMIN uses the

SNMP Interface Check Command entry in `module_config` and modifies it as required.

#### 11 - Create NagMIN Scheduled Cron

The file `/usr/local/nagmin-2.0.0/sbin/nagcron.pl` is the cron script that reads the Nagios `servicestatus` table for hosts that have a 'check\_ping' service check.

Move `nagcron.pl` to wherever you run jobs from. Modify the top of the file as required.

Most important is entering the service description of the 'check\_ping' service



you're using. The service description needs to be the same for all hosts for nagcron.pl to find it.

RRD databases will be created for hosts the first time that nagcron.pl runs as opposed to the host interface databases that are created when you perform the SNMP interface scan.

## 12 - NagMIN Event Handler

The NagMIN event handler writes host and service alert information to the new NagMIN HostAlert and ServiceAlert tables. These tables are displayed under the Nagios Tables section in the bottom of the right-hand column.

```
Copy /usr/local/nagmin-2.0.0/sbin/nagmin_event to
/usr/local/nagios/libexec/eventhandlers.
```

Modify the DB username/password combo as required.

Create two Command table entries for nagmin\_event, one for the Host events and the other for Service events.

```
host_event_handler =
/usr/local/nagios/libexec/eventhandlers/nagmin_event "$HOSTNAME$"
"$HOSTSTATE$" "$OUTPUT$" "$EXECUTIONTIME$" "$TIMET$" "$LASTCHECK$"
"$LASTSTATECHANGE$" "$STATETYPE$" "$HOSTATTEMPT$"
```

```
service_event_handler =
/usr/local/nagios/libexec/eventhandlers/nagmin_event "$HOSTNAME$"
"$HOSTSTATE$" "$SERVICEDESC$" "$SERVICESTATE$" "$OUTPUT$"
"$EXECUTIONTIME$" "$LATENCY$" "$TIMET$" "$LASTCHECK$"
"$LASTSTATECHANGE$" "$STATETYPE$" "$SERVICEATTEMPT$"
```

Modify nagios.cfg if you want to use these services as your global host and service event handlers.

```
global_host_event_handler=host_event_handler
global_service_event_handler=service_event_handler
```