

Easy G2 FoxBox

USER MANUAL

(addenda to SMS FoxBox Manual v2.0)

English Version 1.5

IMPORTANT: This document is not intended as a Nagios 3.x guide

Index of contents

- 1. Introduction**
- 2. Theoretical concepts in Nagios**
- 3. The web interfaces**
- 4. Some configuration hint**
- 5. Sensors integration**
- 6. Bibliographical references**

1. Introduction

This guide will give a brief explanation on the basic concepts on which Nagios 3.x is based, but it will not enter in details more than needed. To get more notions, please refer to the last chapter where you can find some additional material (also the ones used by KDEV when developing this system).

For every topic about the FoxBox not covered in this manual, please refer to the SMS/MMS FoxBox manual on the website (www.smsfoxbox.it).

We remember that Nagios is provided as part of our FoxBox product, but it is not developed or maintained by KDEV. For communications regarding support, bug reports and so on we invite you to refer to the Nagios official website (www.nagios.org). However, on this machine is installed a Nagios 3.2.1 modified. If you install a different version, possibly you will lose the add-ons by KDEV.

2. Theoretical concepts in Nagios

Entities in Nagios are called objects. There are seven types of them:

1. **Hosts:** virtualization of a physical machine. It contains a brief description of it (name, IP address, etc.) and the check parameters (what checks to do, when, who to contact in case of failures, etc.).

Hosts are defined in */usr/local/nagios/etc/hosts.cfg*.

It is possible to use inheritance to simplify the definition of this objects. This is made with host and service templates, a particular kind of abstract objects that are defined in */usr/local/nagios/etc/objects/templates.cfg*.

2. **Services:** virtualization of a functionality related to a certain host (FTP, HTTP, CPU load, etc.). It contains almost the same parameters of the previous object.

Services are defined in */usr/local/nagios/etc/services.cfg*.

Also in this case you can use templates, and the file is the same as above.

3. **Commands:** custom definitions of checks (on hosts or services), notifications or event handlers. These features are used to monitor and manage status-changes.

Commands are defined in */usr/local/nagios/etc/objects/commands.cfg*.

4. **Time periods:** time ranges in which a certain check or a certain notification is executable.

Periods are defined in */usr/local/nagios/etc/objects/timeperiods.cfg*.

5. **Contacts:** identifies someone to be contacted when a check goes bad.

Contacts are defined in */usr/local/nagios/etc/objects/contacts.cfg*.

6. **Dependencies:** a logical definition of the network topology. When an host goes down, Nagios will stop checks on hosts dependent from it, because they will obviously fail not because of a true failure on them.

Dependencies are defined in */usr/local/nagios/etc/dependencies.cfg*.

7. **Escalations (optional):** useful to identify a higher level of assistance. After a given time, the notifications unhandled will be redirected to contacts here specified.

Escalations are defined in */usr/local/nagios/etc/escalations.cfg*.

Note that we indicated the typical file organization, good for standard uses with a limited number of objects involved. However, as indicated in many blogs on the web, you could need to organize the definitions in better ways when managing a large number of hosts and services (configuring appropriately the nagios.cfg file).

3. The web interfaces

There are two separated interfaces on this FoxBox. The first one is the Nagios one, accessible with a browser at the address:

192.168.0.98/nagios (where X.X.0.98 is the standard IP address of FoxBox)

You will need to use the following account:

- **User:** *nagiosadmin*
- **Password:** *nagios*

Generally, from this interface is possible to see the current status of hosts and services configured, access the logs, create reports, schedule checks, etc.

Remember that the access manager works using the contact list you configured. So, the administrator/s can see all the features, while the other users will see only the parts for which they are in list to be notified about (to modify this options, refer to *nagios.cfg*).

KDEV, as previously told, has modified the interface adding on the home the “Quick configuration” tool. This is based on the opensource utility Ignoramus. To use it, just fill the fields with the required informations (the same you would give in a classic text-based configuration) and indicate as target file the related one (if different from the original one).

The second interface is the classical SMS FoxBox one, to manage the messaging services featured on the machine. It's accessible at the address:

192.168.0.98

For more informations, please refer to the FoxBox manual.

4. Some configuration hint

In this chapter I am going to explain a configuration example for a little network with a Linux PC and our FoxBox, just to show what are the files to edit and how. Obviously, I will start from the default file configuration of the FoxBox Nagios.

First of all I open the *contacts.cfg* file to define myself as the user *nagiosadmin*, in the sense that his pager is now my telephone number. This is very important, because this will be the number where the notifications will be sent. In this file, I also check that notification commands are related to FoxBox (I have to read “notify-xxx-by-foxbox”).

Note that contacts, hosts and services have pre-defined templates (in *templates.cfg*) that help us while configuring this objects through the inheritance of all the typical attributes and the default values.

The second step is to define in *hosts.cfg* all the hosts of my network. In this case we have:

- localhost: this is our FoxBox, that will check itself;
- newpc: this is the Linux PC to be remotely checked.

As you can see, the bigger part of the attributes to be defined are inherited from host templates. In this file we also define an hostgroup containing those machines.

Last thing, we have to be sure that, in host definition or at least in his template, we defined the *nagiosadmin* user to be notified about.

The next step is to define services in *services.cfg*, regarding the features we need

to check on every host. Obviously, we don't need to define the same services for both the machines, because their features and uses are not the same. Practically, we will use plugins abstractions contained in *commands.cfg* to make the checks on these services.

Finally, we could define dependencies (in *dependencies.cfg*) and/or escalations (in *escalations.cfg*) and we have finished.

Note that, with the size and complexity of this example, I preferred to configure textually all the files. However, using “*Quick configuration*” tools would have made the same effects.

5. Sensors integration

An option available on this FoxBox are the integrated sensors. With these micro-devices it is possible to check in realtime the status of the environment in which the system is placed.

We need to develop an ad-hoc Nagios plugin for each type of sensor, and place it into the folder `/usr/local/nagios/libexec`. Then we have to register it into `commands.cfg`, and then associate this checks to the FoxBox in `services.cfg`.

Let's now consider the only model actually supported more in detail. As soon as we develop more plugins, we will link them on the FoxBox official site.

Humidity/Temperature

The plugin used in this case is called `check_th`. It is registered twice, because we decided to handle this measures separately.

To customize the warning and critical states thresholds, you need to modify the plugin input parameters in `services.cfg`. Note that humidity is expressed as a percentage [%], while temperature is on Celsius scale [°C].

As you can see, parameters are floating point numbers (except the first one) in the format “`!M!LW!UW!LC!UC`”:

- `!M` indicates the subfunction to call (0 for temperature check, 1 for humidity check);
- The others are the threshold bounds. Respectively, they indicate the lower warning limit, the upper one, the lower critical limit and the upper one.

For example, the default values we chose were: `!14!22!10!26` for temperature, and `!20!40!15!60` for humidity.

6. Bibliographical references

The material used was both online and paper.

For the first class we suggest:

- Official documentation at http://nagios.sourceforge.net/docs/3_0/toc.html;
- Official website at <http://www.nagios.org/>;
- Various system administrator blogs and forums.

Then, the book we strongly recommend is:

- “Learning Nagios 3.0” by W. Kocjan.