

# **Monitoring FoxBox LX800**

## **USER MANUAL**

*(addenda to Gateway FoxBox LX800 Manual)*

**Revision 1.1**

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

*For more informations please write to [luka@kdev.it](mailto:luka@kdev.it)*

# **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 Nagios3 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 Gateway FoxBox LX800 manual on the website ([www.smsfoxbox.it](http://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](http://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.

Remember that supported browsers are:

- Mozilla Firefox
- Opera
- Netscape
- Internet Explorer 6
- Google Chrome

## 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.).

It is possible to use inheritance to simplify the definition of these objects. This is made using host and service templates, which are a particular kind of abstract objects.

On the system, the sample generic host is defined into the file */etc/nagios3/conf.d/generic-host\_nagios2.cfg*.

Hosts could then be defined in different ways.

The classical technique, that is also our default implementation, consist in defining a single host for every configuration file. You can find the sample host in */etc/nagios3/conf.d/localhost\_nagios2.cfg*.

In case of high number of checked hosts, it could be a good idea to reorganize the file structure, for example grouping similar devices in the same configuration file.

Remember that every new configuratio file has to be presented to Nagios engine in order to let it process. To do so, you have to define it using *cfg\_dir* or *cfg\_file* into file */etc/nagios3/nagios.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 */etc/nagios3/conf.d/services\_nagios2.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 */etc/nagios3/commands.cfg*.

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

Periods are defined in */etc/nagios3/conf.d/timeperiods\_nagios2.cfg*.

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

Contacts are defined in */etc/nagios3/conf.d/contacts\_nagios2.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 file is not defined by default.

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 file is not defined by default.

### 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/nagios3* (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, and so on, but it is not possible to edit the configurations files.

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*).

The second interface is the classical FoxBox LX800 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**

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).

## 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/lib/nagios/plugins/`. 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_nagios2.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.

### Presence

The plugin used in this case is called `check_pres`.

## 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](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.