Chapter 18

NagVis

NagVis[1] is an addon for Nagios that displays host and service states against a background image selected by the user. This must be available in the PNG format; the choice is yours whether you use a map, a diagram of your own system documentation, a photo of the server room, or a schematic diagram of the system environment as a background, as shown in Figure 18.1.

Via the Web interface, you can place objects wherever you want on the background. NagVis displays different icons, depending on the state of the object: red for the CRITICAL state, yellow for WARNING, green for OK, and a question mark on a gray background for UNKNOWN. If an acknowledgment was set, this is indicated by a green button with a picture of a worker on it.

There are different icons for hosts and services; in the default template, host icons are rectangular, and service icons are round. A finished NagVis

[1] `http://www.nagvis.org/`

display—NagVis refers to this as a *map*—is shown in Figure 18.2. Further examples—such as using a geographical map, or a photo of the server room, as a background—are provided on the NagVis homepage.[2]

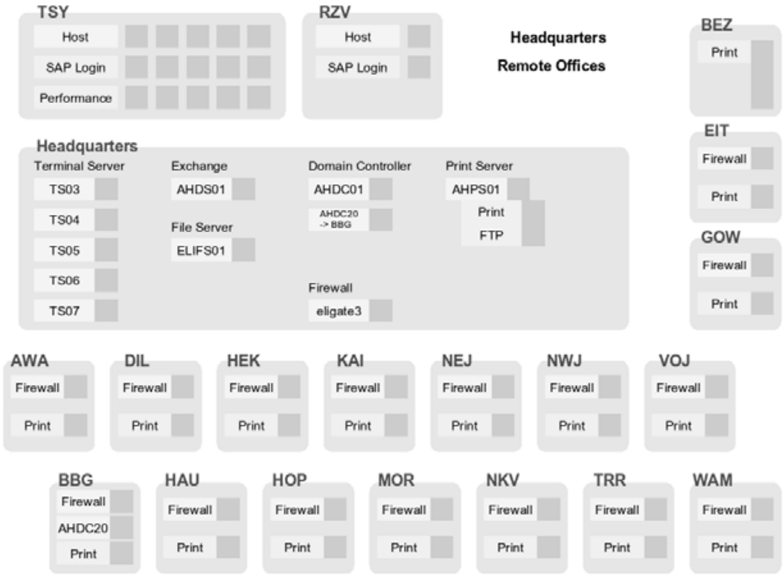


Figure 18.1: Schematic diagram of a system environment as a template for NagVis

In addition to hosts and services, host and service groups can also be integrated into a NagVis display, as well as additional maps. Thus a geographical overview map could be used for the start page, which has an icon for each location monitored that links to a detailed NagVis map specifically for that location.

If an icon contains several states, as is the case for host and service groups, for instance, NagVis displays the state with the highest priority. CRITICAL has a higher priority than WARNING, WARNING trumps UNKNOWN, UNKNOWN gets more attention than an acknowledgment, and OK has the lowest priority of all. If any host in a host group assumes the CRITICAL state, this is shown accordingly for the entire host group.

For hosts and host groups, NagVis offers you the choice of having only host states considered in determining the state that is displayed, or having the services dependent on these hosts are included as well (see page 394). In the latter case, a red stop light is displayed if even a single service of a host is in the critical state.

[2] `http://www.nagvis.org/screenshots`

Of particular interest is NagVis's ability to evaluate only hard states (page 394). For routine work with the Web interface, it turns out to be quite useful if not every temporary soft CRITICAL state immediately generates a red light.

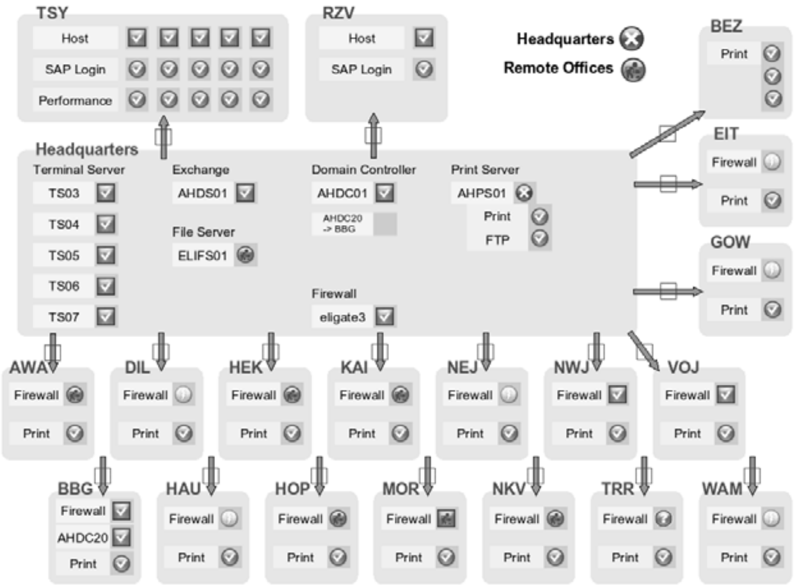


Figure 18.2: Displaying the system environment

It will especially please fans of object-oriented programming that NagVis makes full use of object-oriented concepts. For example, the system inherits its defaults from the global configuration for individual maps and settings on the map level and passes these on to individual objects, with the option of overwriting settings locally always available. This simplifies the configuration to a considerable extent, and NagVis also indicates in the graphical editor (also called the *Web user interface* or WUI) which settings are object-specific and which have been inherited (Figure 18.8 on page 399).

NagVis is published under the GNU Public License Version 2 (GPLv2); the description below refers to version 1.3.

18.1 Installation

NagVis makes use of NDOUtils and is implemented in PHP. Therefore, besides an NDO database in running order, as described in Chapter 17 from page 375, you need a Web server with PHP 4.2 or higher, as well as the packages php-mysql for access to the NDO database and php-gd to be able to draw lines.[3] Depending on the distribution and the PHP version used, the package names may vary slightly. For Debian "Etch" and PHP5 you need the packages libapache2-mod-php5, php5, php5-common, php5-gd, and php5-mysql.

NagVis does not necessarily have to be installed on the same computer as Nagios and the NDOUtils, although in many cases they are packed onto one host. The configuration of the NDO database—NagVis documentation refers to this as the *backend*—can refer (as described on page 395) to any system you please. NagVis even allows the backend to be selected separately for each individual object so that maps can be generated that combine several Nagios installations in a single graphic.

18.1.1 Installing the source code

The NagVis source code, from http://www.nagvis.org/downloads, is unpacked in a directory of your choice:

```
linux:~ # tar xvzf /pfad/zu/nagvis-1.3.tar.gz
...
```

If a previous installation exists, you should back this up first. Then you copy the directory that has been created (in our case, nagvis-1.3) with the name nagvis to /usr/local/nagios/share:

```
linux:~ # mv nagvis-1.3 /usr/local/nagios/share/nagvis
linux:~ # ls -F /usr/local/nagios/share/nagvis
INSTALL LICENCE README config.php etc/ index.php nagvis/ var/ wui/
```

The duplicated directory name nagvis can sometimes lead to confusion, but it is correct:

```
/usr/local/nagios/share/nagvis
/usr/local/nagios/share/nagvis/nagvis
/usr/local/nagios/share/nagvis/wui
```

/usr/local/nagios/share/nagvis represents the main directory of the NagVis installation, while the subdirectory nagvis contains the NagVis application, together with its configuration. Finally the subdirectory wui contains the graphic editor that enables NagVis maps to be edited via the browser.

[3] You can manage without using the GD libraries if the parameter usedgdlibs in the NagVis configuration file config.ini.php (see page 393) is set to 0.

At present, the correct access permissions for directories and files must be set manually. To do this, you first need to determine the user with whose permissions the Web server is running (see also Section 1.2, page 39):

```
linux:~ # grep "^User" /etc/apache2/apache2.conf
User www-data
linux:~ # id www-data
uid=33(www-data) gid=33(www-data) Groups=33(www-data),9001(nagcmd)
```

The first `grep` command looks for the corresponding user in the configuration file for the Web server—in this case, Apache2—and then the `id` command searches for the primary group of this user. This can be found after the `gid=` specification. The access permissions are now set accordingly:

```
linux:~ # chown www-data.www-data -R /usr/local/nagios/share/nagvis
linux:~ # chmod 664 /usr/local/nagios/share/nagvis/etc/nagvis.ini.php
linux:~ # chmod 775 /usr/local/nagios/share/nagvis/nagvis/images/maps
linux:~ # chmod 664 /usr/local/nagios/share/nagvis/nagvis/images/maps/*
linux:~ # chmod 775 /usr/local/nagios/share/nagvis/etc/maps
linux:~ # chmod 664 /usr/local/nagios/share/nagvis/etc/maps/*
linux:~ # chmod 775 /usr/local/nagios/share/nagvis/var
linux:~ # chmod 664 /usr/local/nagios/share/nagvis/var/*
```

Before the Web user interface can be used, you must create the central configuration file and ensure that access to NagVis is only possible after successful authentication.

18.1.2 Initial configuration

A template for the central NagVis configuration file `config.ini.php` can be found in the directory `/usr/local/nagios/share/nagvis/etc`, which only needs to be renamed and modified:

```
linux:~ # cd /usr/local/nagios/share/nagvis/etc
linux:nagvis/etc # cp config.ini.php.dist config.ini.php
linux:nagvis/etc # chown www-data.www-data config.ini.php
linux:nagvis/etc # chmod 664 config.ini.php
```

The commands `chown` and `chmod` ensure that the correct access permissions are set for the Web user (here, `www-data`) and his group.

Apart from the configuration of the backend (that is, the NDO database), the included `config.ini.php` already has usable defaults. The following description is therefore limited to introducing the most important parameters:[4]

[4] The complete documentation can be found at `http://www.nagvis.org/docs/1.3/nagvis_config_format_description`.

```
[global]
language="german"
refreshtime=60
```

In the `[global]` section you can set the language with `language`; the default is `english`. `refreshtime` defines every how many seconds the display in the brower is refreshed.

The section `[defaults]` specifies defaults that are inherited by the defined objects from the map configuration. The values can be overwritten via the map, if required. It is best to define settings here that are identical for the majority of objects, in order to avoid the repeated work of defining them explicitly in the object definitions:

```
[defaults]
backend="ndomy_1"
icons="std_medium"
recognizeservices=1
onlyhardstates=1
```

`backend` specifies which NDO database is used as the default backend. The name for this can be anything you like, but the backend itself must still be defined in a separate section (see page 395). If you are just getting started, it is best to keep the name supplied, `ndomy_1`.

The parameter `icons` defines the icon set from the directory `./nagvis/nagvis/images/iconsets` that is to be used. Four sets are included: `std_small`, `std_medium`, `std_big`, and `folder`. Other icons can be downloaded from the NagVis homepage,[5] or you can create them yourself.[6]

The setting `recognizeserivces=1` ensures for hosts and host groups that the current states of the accompanying services are included when the overall state is being determined. The value 0 switches off this behavior.

`onlyhardstates=1` on the other hand instructs NagVis to take only hard states into account. The default `0` also includes soft states.

The `[wui]` section allows settings to be made for the NagVis editor:

```
[wui]
autoupdatefreq=25
maplocktime=5
```

`autoupdatefreq` determines how often (in number of seconds) the Web user interface automatically saves changes, while `maplocktime` specifies the number of minutes after which any further changes to a map that is

[5] `http://www.nagvis.org/downloads`

[6] A corresponding guide can be found at `http://www.nagvis.org/docs/extending/iconsets`.

currently being edited should be blocked, from the time of the last change. This is intended to prevent several users from simultaneously editing the same map.

The paths to the NagVis installation from the perspective of the file system (`base`) and—separately for NagVis data and NagVis CGIs—from the perspective of the browser, are specified in Section `[paths]`:

```
[paths]
base="/usr/local/nagios/share/nagvis/"
htmlbase="/nagios/nagvis"
htmlcgi="/nagios/cgi-bin"
```

The defaults listed here match the standard installation described above.

The configuration for the backend, that is, for accessing the NDO database, follows at the bottom of the file:

```
[backend_ndomy_1]
backendtype="ndomy"
dbhost="localhost"
dbport=3306
dbname="nagios"
dbuser="nagios"
dbpass="verysecret"
dbprefix="nagios_"
dbinstancename="default"
maxtimewithoutupdate=180
```

This name of this section must contain the name specified with the `backend` parameter under `[defaults]`, according to the pattern `[backend_`*`value_of_backend`*`]`. The default here is `ndomy_1`. If the `backend` parameter value does not match any of the defined backend sections, NagVis will refuse to work.

`backendtype` defines the type of backend, and for now `ndomy`—an NDO database based on MySQL—is the only possible value.

`dbhost` and `dbport` specify the host name or IP address and the accompanying TCP port for access to the database. `dbname` contains the name of the NDO database, and `dbuser` and `dbpass` give the user and the password for access.

The values defined by default for `dbprefix` and `dbinstancename` are set for a NDOUtils standard installation. Provided that you have not changed the parameter `instance_name` in the file `ndomod.cfg` (see Chapter 17.4.1 from page 384) and `db_prefix` in the file `ndo2db.cfg` (see Chapter 17.4.2 from page 385), you can use the values given here.

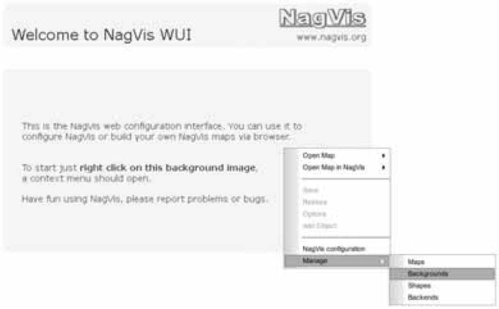
One parameter to which you should pay particular attention is `maxtimewithoutupdate`: This defines how much time in seconds is allowed for the status update of Nagios to appear. If the time specified here is exceeded, NagVis assumes that the data is obsolete and displays this as an error. If NagVis accesses NDO databases that are distributed across several servers, it is essential that the clock times of the servers be synchronized with one another, otherwise NagVis will refuse to work if it comes across a time difference greater than `maxtimewithoutupdate` seconds.

At this point we shall mention another problem involving data exchange between the NDO database and NagVis: NagVis evaluates the current program status. Nagios versions prior to 3.0b1, however, only write this to the NDO database *after* the nightly change of the log files. Starting with version 3.0b1, Nagios updates the status every five seconds, so that NagVis always has up-to-date information.

18.1.3 User authentication

NagVis demands authentication from the user. Without user authentication, it will just issue an error message.[7]

If the `share` directory of Nagios is not accessible for authentication, as in the Apache configuration on page 48, you should change this in the Apache configuration file `/etc/apache2/conf.d/nagios`. The authentication data are best taken from the CGI directory (see Section 1.5 from page 47).



Welcome to NagVis WUI

This is the NagVis web configuration interface. You can use it to configure NagVis or build your own NagVis maps via browser.

To start just **right click on this background image**, a context menu should open.

Have fun using NagVis, please report problems or bugs.

Figure 18.3: After a right mouse click on the graphic displayed, the menu will appear

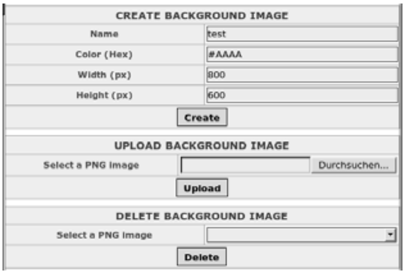
18.2 Creating NagVis Maps

The NagVis configuration interface is accessed through the URL `http://`*`nagiosserver`*`/nagios/wui/index.php`. Here you replace *`nagiosserver`* with your own Nagios host name (Figure 18.3 shows the start page).

[7] The FAQ entry `http://www.nagvis.org/docs/general/faq#how/to/run/nagvis/without/authentication/` describes a way of using NagVis without authentication, but you should avoid this, for reasons of security.

It is operated through a menu that opens when you click the right mouse button. For browsers that overlay the NagVis menu with their own menu, a second click with the right mouse button will cause the browser menu to disappear.

If you don't have a suitable background graphic available in PNG format, you can create a new, empty background image via the menu item **Manage | Backgrounds**, as shown in Figure 18.4. In this example, a gray image of size 800x600 pixels is created.



CREATE BACKGROUND IMAGE

Name	test
Color (Hex)	#AAAA
Width (px)	800
Height (px)	600

UPLOAD BACKGROUND IMAGE

Select a PNG image	

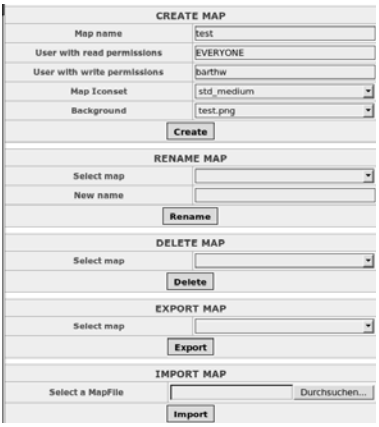
DELETE BACKGROUND IMAGE

Select a PNG image	

Figure 18.4: Creating a new, empty background

This image is displayed on screen in its original size, that is, not scaled. To avoid scrolling, it should not be too large, with the proper size depending on the screen resolution and taking into account possible window frames.

Later on the image size can only be altered to a limited extent: although you can upload or create a background image at any time, objects that have already been placed are not affected by any changes in the size or other characteristics of the background, and so they may not fit properly into a new background image. The only option then is to reposition all the objects. The best approach is to experiment at first with just a few objects in the definitive environment before you set up an extensive map.



CREATE MAP

Map name	test
User with read permissions	EVERYONE
User with write permissions	barthw
Map Iconset	std_medium
Background	test.png

RENAME MAP

Select map	
New name	

DELETE MAP

Select map	

EXPORT MAP

Select map	

IMPORT MAP

Select a MapFile	

Figure 18.5: If the entry for Map Iconset remains empty, NagVis uses the defaults from the central configuration.

You now create a new map for the generated or uploaded image. Via the menu entry **Manage | Maps** (Figure 18.3) you navigate to the dialog shown in Figure 18.5. There you define the name of the map and select the background image. If every authenticated user should see the map, you enter `EVERYONE` as **User with read permissions**. For the `User with write permissions`, on the other hand, you would probably enter a specific user, or several user names, separated by commas. Defining the icon sets is optional at this point, although the example shown specifies them.

Newly created maps are automatically opened by NagVis in editing mode. You can reach this later on via the menu item **Open Map** in the context menu of the opening dialog or by calling it directly with the URL `http://`*`nagiosserver`*`/nagios/nagvis/wui/index.php?map=`*`mapname`*.

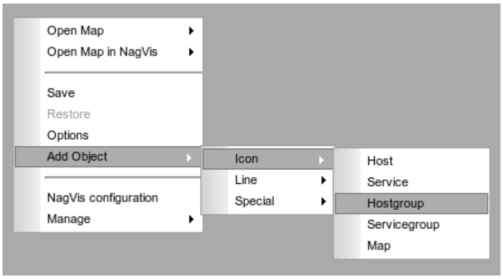


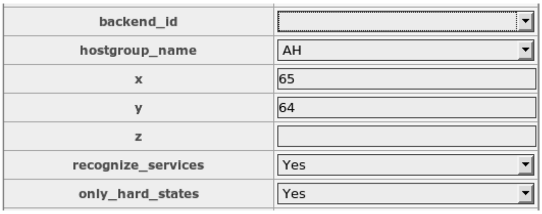
Figure 18.6: Inserting objects via the graphical interface

The *`mapname`* for the map created in Figure 18.5 is `Test`. In the map itself you now insert objects using the right mouse button (see Figure 18.6). An object can be an icon, a line, or a special object. Icons and lines represent the current state of a host or service and can also stand for entire host or service groups. Icons can only reflect the overall state of a map. Special objects are graphics representing stateless objects (which might be icons as well, for example) or text boxes, which can also be provided with a hyperlink.

To insert a host group you select **Add Object | Icon | Hostgroup** in the menu. Then you place the mouse over the desired position and define the destination of the icon with a left mouse click (the position can be changed later on).

When this is done a dialog opens, as shown in Figure 18.7. The entry **backend_id** can remain empty, and NagVis will then use the value of the `backend` parameter from the central configuration file `config.php.ini`. In the **hostgroup_name** pulldown menu, NagVis allows you to select from all defined host groups.

Icons can be positioned according to the **x** and **y** coordinates by either entering numerical values or using the mouse. For simple icons, the point (x,y) corresponds to the center, and for lines to the start or end points. The z coordinate is only used when icons overlap. The value 0 describes the underlying image at the rear and is reserved for the background, and the object with the highest z value is right at the front. If the z coordinate is explicitly left empty when the object is inserted, it automatically receives the value 1.



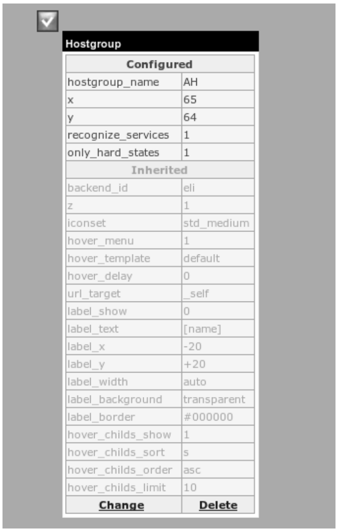
backend_id	
hostgroup_name	AH
x	65
y	64
z	
recognize_services	Yes
only_hard_states	Yes

Figure 18.7: Defining a host group in the graphical interface

The parameter `recognize_services` allows the `recognizeservices` setting from the `config.ini.php` (see page 394) to be overridden, and `only_hard_states` does the same for the parameter `onlyhardstates` (page 394).

The object inserted in this way always appears in the graphical editor in the form of the OK icon from the icon set chosen; the Web interface takes no account of its actual state.

If you move the mouse over the icon, a hover menu opens, as shown in Figure 18.8. It clearly distinguishes which settings are inherited and which ones have been specified directly in the object. If you follow the **Change** link there, the settings can be changed again.



Hostgroup

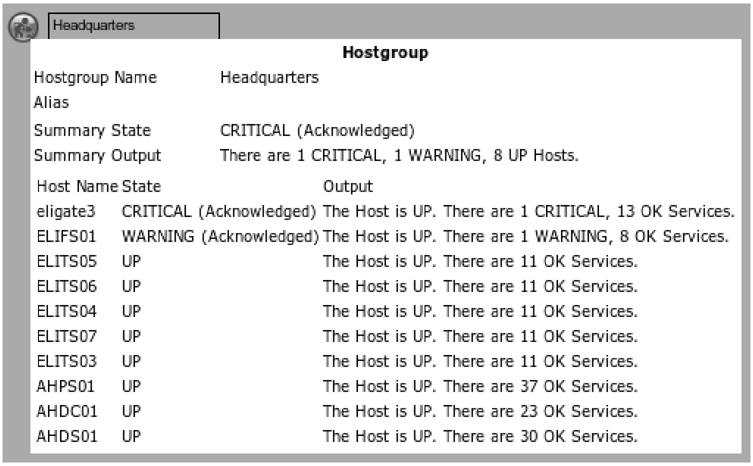
Configured	
hostgroup_name	AH
x	65
y	64
recognize_services	1
only_hard_states	1
Inherited	
backend_id	eli
z	1
iconset	std_medium
hover_menu	1
hover_template	default
hover_delay	0
url_target	_self
label_show	0
label_text	[name]
label_x	-20
label_y	+20
label_width	auto
label_background	transparent
label_border	#000000
hover_childs_show	1
hover_childs_sort	s
hover_childs_order	asc
hover_childs_limit	10
Change	**Delete**

Figure 18.8: If you move the mouse over the inserted object, a hover menu opens

When your work is finished, don't forget to save your changes via the context menu entry **Save** (see Figure 18.3 on page 396). The menu item **Open Map in NagVis** will then take you to the finished view, which now does display the actual states.

In Figure 18.9 a text box has been added to the host group icon. The field displayed beneath this is a hover menu, which shows information on the object and its state if you move the mouse over the object. This example shows that two hosts of the host group display a Not OK state and that this has already been confirmed with an acknowledgment.

Figure 18.9: Object in the final view with text box and hover menu



Headquarters

Hostgroup

Hostgroup Name	Headquarters
Alias	
Summary State	CRITICAL (Acknowledged)
Summary Output	There are 1 CRITICAL, 1 WARNING, 8 UP Hosts.

Host Name	State	Output
eligate3	CRITICAL (Acknowledged)	The Host is UP. There are 1 CRITICAL, 13 OK Services.
ELIFS01	WARNING (Acknowledged)	The Host is UP. There are 1 WARNING, 8 OK Services.
ELITS05	UP	The Host is UP. There are 11 OK Services.
ELITS06	UP	The Host is UP. There are 11 OK Services.
ELITS04	UP	The Host is UP. There are 11 OK Services.
ELITS07	UP	The Host is UP. There are 11 OK Services.
ELITS03	UP	The Host is UP. There are 11 OK Services.
AHPS01	UP	The Host is UP. There are 37 OK Services.
AHDC01	UP	The Host is UP. There are 23 OK Services.
AHDS01	UP	The Host is UP. There are 30 OK Services.

The finished map can be called directly via the URL `http://`*`nagiosserver`*`/nagios/nagvis/nagvis/index.php?map=`*`mapname`*.

18.2.1 Editing the configuration in text form

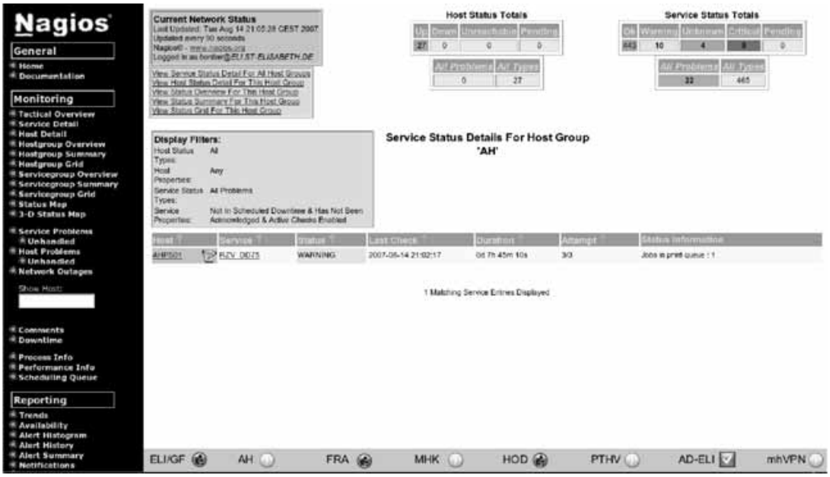
NagVis stores the entire configuration of a map in text files, which can also be edited with a text editor. The files are located in the directory `/usr/local/nagios/share/nagvis/etc/maps/`. If you are using a background image with a known raster, you can insert several objects in the WUI and continue editing the map in the editor using the coordinates just determined. This is how the map shown in Figure 18.2 was created. The background image (Figure 18.1) was created with OpenOffice in order to obtain a reproducible raster; the OpenOffice drawing is subsequently exported as a PNG file.

The configuration options for the text files, which altogether are very extensive, are described in the online documentation.[8]

[8] `http://www.nagvis.org/docs`

18.2.2 Adding NagVis maps to the Nagios Web interface

NagVis maps can also be integrated into the Nagios Web interface. Figure 18.10 shows these after a third frame has been added to the `index.html` page, which binds a 32-pixel-high map beneath the main window. No matter what the administrator is currently working on, the most important states (here these are the host groups) are always displayed directly and can be reached with a single mouse click. There are no limits to your user interface dreams when using NagVis!



Current Network Status
Last Updated: Tue Aug 14 21:05:28 CEST 2007
Updated every 90 seconds
Nagios® - www.nagios.org
Logged in as bontler@ELI.ST-ELISABETH.DE

View Service Status Detail For All Host Groups
View Host Status Detail For This Host Group
View Status Overview For This Host Group
View Status Summary For This Host Group
View Status Grid For This Host Group

Host Status Totals

Up	Down	Unreachable	Pending
27	0	0	0

All Problems	All Types
0	27

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
443	10	4	8	0

All Problems	All Types
33	465

Display Filters:

Host Status Types:	All
Host Properties:	Any
Service Status Types:	All Problems
Service Properties:	Not In Scheduled Downtime & Has Not Been Acknowledged & Active Checks Enabled

Service Status Details For Host Group 'AH'

Host	Service	Status	Last Check	Duration	Attempt	Status Information
AHPS01	RZV_DD75	WARNING	2007-08-14 21:02:17	0d 7h 45m 10s	3/3	Jobs in print queue : 1

1 Matching Service Entries Displayed

Figure 18.10: NagVis map as a "footnote" in the Nagios Web interface