

NagiosQL 3.2

Installation

2012 – Version 2.0
Martin Willisegger

Table of contents

1	Installation.....	3
1.1	Prerequisites.....	3
1.2	Prepare directory structure.....	3
1.3	Nagios configuration files.....	4
1.4	Prepare Apache configuration.....	5
1.5	Install PHP modules (SSH)	6
1.6	Install NagiosQL.....	6
1.7	NagiosQL installer (new installation).....	7
1.8	Update NagiosQL.....	12
2	Configure NagiosQL.....	15
2.1	Base configuration.....	15
2.2	Domain config (paths)	16
2.2.1	Data domains.....	16
2.2.2	Configuration domain.....	18
2.3	2.3 Setup remote access (FTP or SSH/SFTP).....	20
2.3.1	Remote access via FTP.....	20
2.3.2	Remote access via SSH/SFTP	20
3	Nagios Configuration.....	22
3.1	Nagios Configuration file.....	22
3.2	Additional permissions.....	22
3.2.1	Check command.....	22
3.2.2	Configuration editor	23
3.2.3	Restart command	23
3.3	Support page	23
3.3.1	Common checks	23
3.3.2	Config checks.....	24

1 Installation

1.1 Prerequisites

NagiosQL is a web application and requires a running web server including PHP support and a database server.

- Apache web server
- MySQL Server version 4.1 or above
- PHP version 5.2 or above
- Nagios 2.x or 3.x
- additional PHP modules:
 - session
 - MySQL (php5-mysql)
 - gettext
 - filter
 - ftp (optional – for FTP connections to remote servers)
 - SSH (PECL module – optional – for SSH/SCP connections to remote servers)

1.2 Prepare directory structure

NagiosQL stores the Nagios configuration files in a separate directory structure. This should preferably be created by a super user (root). Where the files are stored is left to the administrator, in this guide we will use the folder **/etc/nagiosql**. Please note that this directory structure has to be added to the Nagios configuration file **nagios.cfg** later on so Nagios can find its own configuration files.

This directory structure has to be created on the server running Nagios.

```
# mkdir /etc/nagiosql
# mkdir /etc/nagiosql/hosts
# mkdir /etc/nagiosql/services
# mkdir /etc/nagiosql/backup
# mkdir /etc/nagiosql/backup/hosts
# mkdir /etc/nagiosql/backup/services
```

This way NagiosQL will store all configuration files below **/etc/nagiosql**. Host configurations will be stored in **/etc/nagiosql/hosts**, service configurations in **/etc/nagiosql/services**. The backup structure will hold the previous version of a configuration file.

NagiosQL will run as a web application so you have to make sure that the web server daemon has write permissions on these folders if this is a local installation. Furthermore the Nagios daemon must have read permissions. If the users of both daemons are unknown you can use the following commands:

```
# ps -ef | grep http
root      2999      1  0 Jul04 ?    00:00:10 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf
wwwrun    3123    2999  0 Jul04 ?    00:00:00 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf
```

The user running the web server daemon is **wwwrun**.

```
# ps -ef | grep nagios
nagios    2902      1  0 Jun22 ?    00:05:39 /usr/sbin/nagios -d /etc/nagios/nagios.cfg
```

The user running the Nagios daemon is **nagios** but in this case we need the group.

```
# id nagios
uid=111(nagios) gid=111(nagios) groups=111(nagios)
```

The group of the user running the Nagios daemon is called **nagios** as well.

Now we can set the permissions of the structure as follows:

```
# chown -R wwwrun.nagios /etc/nagiosql
```

Alternatively the permissions can be set to 750 (directories), and 640 (cfg files), respectively.

Using remote Nagios installations being administered by NagiosQL via FTP or SSH/SCP this directory structure has to be created on the remote system. Then instead of the web server daemon user you have to set write permissions on the directories for the user accessing via FTP or SSH/SCP.

1.3 Nagios configuration files

NagiosQL needs read/write permissions on some Nagios configuration files as well. The following files are affected::

Nagios main configuration file: **nagios.cfg** (write permissions)

Nagios CGI configuration file: **cgi.cfg** (write permissions)

Nagios process file: **nagios.pid** or **nagios.lock** (read permissions)

Nagios command pipe: **nagios.cmd** (write permissions)

The location of these files can be determined using the „find“ command:

```
# find / -name nagios.cfg
/etc/nagios/nagios.cfg
# find / -name cgi.cfg
/etc/nagios/cgi.cfg
# find / -name nagios.pid
/var/run/nagios.pid
# find / -name nagios.lock
/var/run/nagios.lock
# find / -name nagios.cmd
/var/spool/nagios/nagios.cmd
```

Please note that the web server daemon and the Nagios daemon have to get the necessary permissions. This can be accomplished in several ways via user/group permissions. Following an example for a local installation:

```
# chown -R wwwrun.nagios /etc/nagios/nagios.cfg
# chown -R wwwrun.nagios /etc/nagios/cgi.cfg
# chown -R wwwrun.nagios /var/spool/nagios/nagios.cmd
# chmod 640 /etc/nagios/nagios.cfg
# chmod 640 /etc/nagios/cgi.cfg
# chmod 660 /var/spool/nagios/nagios.cmd
```

Read permissions on the process file are normally achieved via standard permissions 644 but in this case the path should be remembered. All other paths should be jotted down as well so you can fill them in into the NagiosQL configuration domain.

1.4 Prepare Apache configuration

Now a web server directory has to be created where the NagiosQL web application can be installed later on. This guide will use **/opt/nagiosql**.

```
# mkdir /opt/nagiosql
# chown wwwrun /opt/nagiosql
```

The setting of two options in **php.ini** of the Apache web server should be verified (using openSuSE/SLES the location is **/etc/php5/apache/php.ini**, Fedora, CentOS **/etc/php.ini**, Debian **/etc/php5/apache2/php.ini**):

```
file_uploads = On
session.auto_start = 0
date.timezone = "Europe/Zurich" (bzw. "Europe/Berlin" oder "Europe/London" etc.)
```

Please create a configuration file for the web server. The location differs between distributions. Using openSuSE/SLES/Debian the location is **/etc/apache2/conf.d**, Fedora, CentOS **/etc/httpd/conf.d**.

```
# touch /etc/apache2/conf.d/nagiosql.conf
```

The file can be modified using your favourite editor (e.g. vi) and should have the following contents:

```
# NAGIOSQL SAMPLE CONFIG SNIPPETS FOR APACHE WEB SERVER
# Last Modified: 03-24-2011
#
# This file contains examples of entries that need
# to be incorporated into your Apache web server
# configuration file. Customize the paths, etc. as
# needed to fit your system.
```

```
Alias /nagiosql "/opt/nagiosql"
```

```
<Directory "/opt/nagiosql">
    Options None
    AllowOverride None
    Order allow,deny
    Allow from all
    # Order deny,allow
    # Deny from all
    # Allow from 127.0.0.1
    # AuthName "NagiosQL Access"
    # AuthType Basic
    # AuthUserFile /etc/nagiosql/auth/nagiosql.users
    # Require valid-user
</Directory>
```

If you want to use basic Apache authentication the lines in red have to be activated, the blue ones deactivated. The path to **AuthUserFile** has to be adapted to your environment. Besides basic authentication you can use every other authentication method supported by Apache (LDAP/Kerberos/MySQL/etc.) – please consult the Apache documentation for details.

Please verify that Apache processes these *.conf files at all. This might be done via http.conf or using a suitable VirtualHost configuration:

```
Include /etc/apache2/conf.d/nagiosql.conf
```

Don't forget to create the web folder, test the new configuration and activate the changes:

```
# rcapache2 configtest
# rcapache2 restart
```

1.5 Install PHP modules (SSH)

If PHP modules are missing (prerequisites or installer), they have to be installed before continuing. In most cases this is possible using the tools included in your distribution (YaST/apt-get/rpm). A special case is the SSH module. Following some instructions on how to install SSH for NagiosQL.

To install the SSH module the tool PECL is needed. Normally this can be installed using the package tool (YaST/apt-get/rpm).

If PECL is installed it can be called issuing:

```
# pecl
```

Without any parameters some help text is shown. If the system is behind a proxy server, it might be necessary to configure PECL to use it:

```
# pecl config-set http_proxy http://proxy.myserver.tld:8080
# pecl config-show
```

Now the SSH module can be installed:

```
# pecl install ssh2
```

Then it has to be included in php.ini. Don't forget to restart Apache:

```
extension = ssh2.so

# rcapache2 restart
```

The setup of a user for SSH is shown later on in this guide.

1.6 Install NagiosQL

Now you can install NagiosQL into the new web folder. Please copy it to /opt (not /opt/nagiosql) and unpack it:

```
# cd /opt
# gunzip -c nagiosql_320.tar.gz | tar xf -
# mv nagiosql32 nagiosql
# chown -R wwwrun /opt/nagiosql
# chmod 750 /opt/nagiosql/config
```

Now NagiosQL should be reachable via your web browser:

<http://localhost/nagiosql>

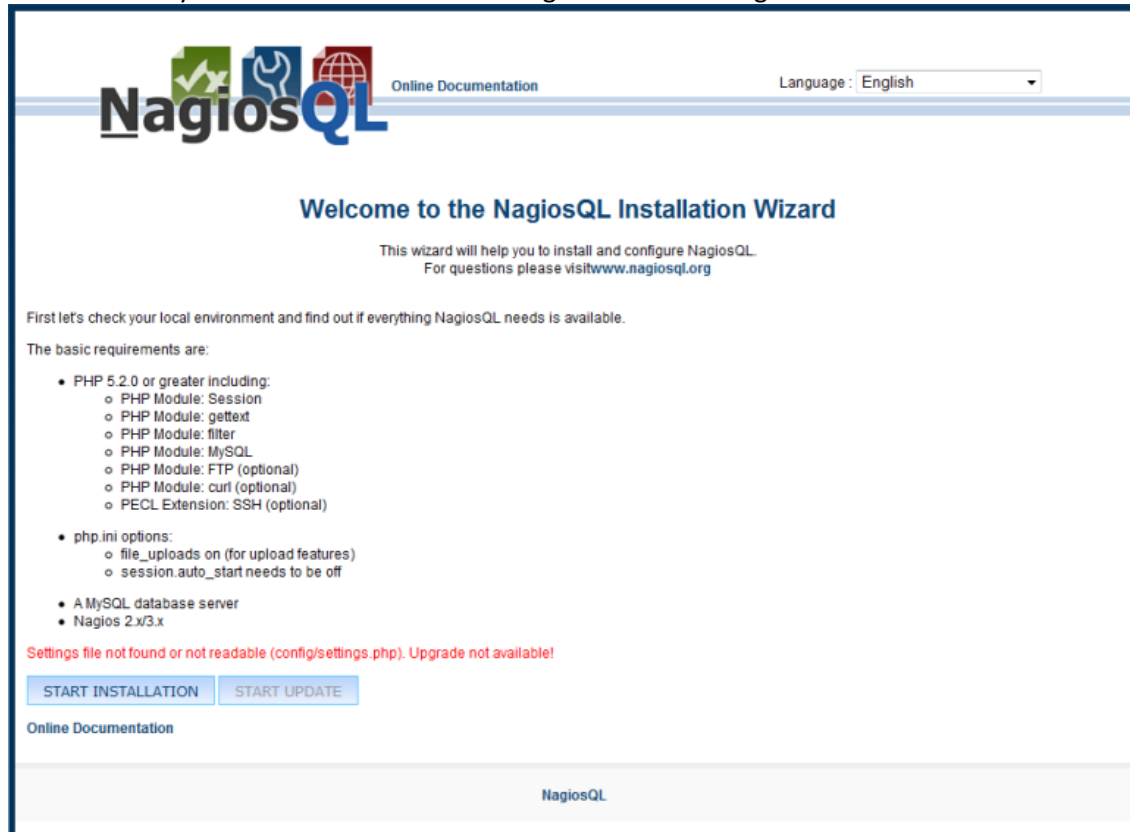
<http://mein.virtualhost.tld/nagiosql>

1.7 NagiosQL installer (new installation)

Normally the installer is started automatically during the first call of NagiosQL. You might start it manually as well:

<http://localhost/nagiosql/install/index.php>

If all is correct your browser should show something like the following:



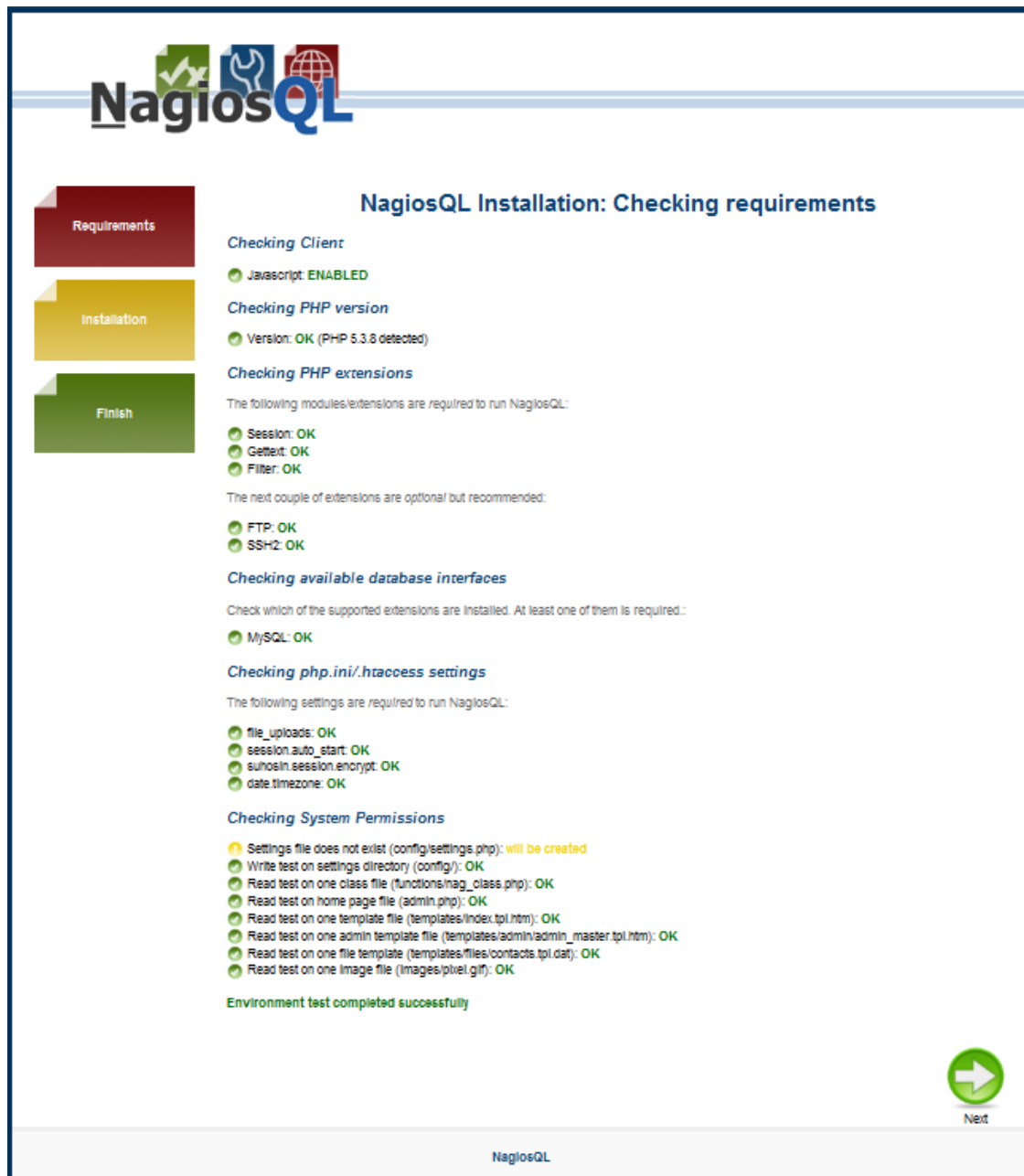
This page just shows a short list of the current prerequisites for the installation of NagiosQL. Additionally it is possible to select your favourite language.

START UPDATE is only available if a valid version of settings.php of a previous version is copied to /opt/nagiosql/config. The update is described later on in 1.8.

Click **START INSTALLATION** to continue.

Next are the environment tests divided into seven sections.

The following image shows an executable environment:



which are

Checking Client:

It is checked if the browser processes JavaScript and if this is activated. Some browser add-ons can disable JavaScript – if this is the case JavaScript has to be enabled for NagiosQL explicitly.

Checkin PHP version:

It is checked if the PHP version is adequate. If this is not the case you have to update your PHP version. Consult the documentation of your distribution for details.

Checking PHP extensions:

The first PHP extensions are mandatory. Missing ones must be installed to be able to use NagiosQL. The optional extensions only have to be installed if you want to use the appropriate options of NagiosQL.

Checking available database connections:

NagiosQL requires at least one PHP module to interface with a database (e.g. module MySQL).

Checking php.ini/.htaccess settings:

It is checked if the required settings in php.ini are correct (see 1.4)

Checking System Permissions:

It is checked if all NagiosQL files are readable/writeable. At the most some permissions have to be modified. If you followed this guide all settings should be correct already.

You can recheck the environment by calling `install/index.php` and clicking **START INSTALLATION** once again.

If all tests are successful you can click on „Next“ to get to the next page:

NagiosQL Installation: Setup

Please complete the form below. Mandatory fields marked *.

Database Configuration

Database Type:

Database Server *:

Local hostname or IP address *:

Database Server Port *:

Database name *:

NagiosQL DB User *:

NagiosQL DB Password *:

Administrative Database User *:

Administrative Database Password *:

Drop database if already exists? ☐

NagiosQL User Setup

Initial NagiosQL User *:

Initial NagiosQL Password *:

Please repeat the password *:

Nagios Configuration

Import Nagios sample config? ☐

NagiosQL path values

Create NagiosQL config paths? ☐

NagiosQL config path:

Nagios config path:

Both path values were stored in your configuration target settings for localhost.
If you select the create path option, be sure that the NagiosQL base path exist and the webserver demon has write access to it. So the installer will create the required subdirectories in your localhost's filesystem (hosts, services, backup etc.)

Next

On this page the information for the database server is specified.

Database Type: Select the database type (not yet active in 3.2.0)

Database Server: Host name of the database server

Local hostname or IP address: Host name or IP address of the web server

Important: if the database server is not running on the same system as the web server the entry „local hostname“ is used to specify the access (source) for the database server!

(allow access for: nagiosql_user@webserver)

Database Server Port: TCP port of the database server daemon

Database name: Name of the database to be created for NagiosQL

NagiosQL Database User: User name for the NagiosQL database

NagiosQL Database Password: Password of the NagiosQL database user

The NagiosQL database user will not exist during a new installation and will be created according the information given here.

Administrative Database User: Admin user

Administrative Database Password: Admin password

The administrative user (e.g. root) has to exist already and must have sufficient rights to create a new database and a user which will be given access to the new database. The data of the administrative user will NOT be stored in NagiosQL – it will only be used to create the database.

Drop database if already exists: Delete a database with the same name (**Attention**)

Initial NagiosQL User: NagiosQL admin user (admin)

Initial NagiosQL Password: NagiosQL admin password

Administrative user for login to the NagiosQL web application. This user cannot be deleted later on.

„Import Nagios sample config“ would import some sample data into the database.

„Create NagiosQL config paths“ creates a directory structure as described in 1.2 (which isn't necessary anymore if you following this guide). One requirement would be that the NagiosQL config directory is writeable for the webserver daemon. A modification of UID/GID is not possible so it has to be done later on manually.

NagiosQL config path: Base directory for the config files

Nagios config path: Base directory for the Nagios config files (contains nagios.cfg)

This information is used to preconfigure the „localhost“ domain.

If all information was entered correctly and the database could be created, you get the following output:



Afterwards the directory `/opt/nagiosql/install` should be deleted.

```
# rm -rf /opt/nagiosql/install
```

After successful installation some base configurations within NagiosQL have to be checked or set. The next chapter deals with this task.

1.8 Update NagiosQL

Update is similar to installation as described in 1.7. Please note that the file **config/settings.php** of the old NagiosQL installation will not be modified and is copied to the config directory of the new installation.

You must never copy the files of a new version into the existing directory of an old/previous NagiosQL version!

It is very important to backup the existing database before you start! Otherwise if anything goes wrong during the update it **cannot be repeated** and the database may be corrupt and in an undefined state. In most cases you then have to run a new installation and import all configuration data.

A backup can be made very easily using common tools like phpMyAdmin. Alternatively there are command line tools like mysqldump.



Click **START UPDATE** to continue.

Afterwards the same environment tests are executed as during a new installation. Please consult the page for details.

If all tests are successful you can click on „Next“ to get to the next page:



The screenshot shows the 'NagiosQL Update: Setup' interface. On the left, there are three tabs: 'Requirements' (red), 'Update' (yellow), and 'Finish' (green). The 'Update' tab is active. The main area contains a 'Database Configuration' form with the following fields and values:

Database Configuration	
Database Type	mysql
Database Server *	localhost
Local hostname or IP address *	localhost
Database Server Port *	3306
Database name *	db_nagiosql_v32
Administrative Database User *	root
Administrative Database Password *	

Below the form, there is a green circular button with a white right-pointing arrow, labeled 'Next'. At the bottom center, the text 'NagiosQL' is displayed.

To run the update as expected you only have to enter the information regarding the administrative database user. The other data is modifiable as well but you should only alter it if you are sure that you know what you are doing.

Administrative Database User: Admin user

Administrative Database Password: Admin password

The administrative user (e.g. root user) has to exist already and must have sufficient rights to create a new database and a user which will be given access to the new database. The data of the administrative user will NOT be stored in NagiosQL – it will only be used to create the database.

If all information was entered correctly and the database could be created, you get the following output:



Afterwards the directory `/opt/nagiosql/install` should be deleted.

```
# rm -rf /opt/nagiosql/install
```

2 Configure NagiosQL

2.1 Base configuration

The NagiosQL base configuration can be accessed in the web application clicking **Administration – Settings**. Here you will find some other data which you entered using the installer. The blue question mark symbol next to each item will show a short help text.

Administration

Administration -> Administration -> Settings

Domain: localhost Logged in: admin Logout

Configure Settings

Path

Temporary Directory * /tmp ?

Server protocol * http ?

Language

Language English ?

Encoding utf-8 ?

Database

MySQL Server * localhost ?

MySQL Server Port * 3306 ?

Database name * db_nagiosql_v32 ?

Database user * nagiosql_user ?

Database password ?

Security

Session auto logoff time 3600 ?

Authentication type* NagiosQL ?

Common

Data lines per page 15 ?

Selection method NagiosQL 3 ?

Template warn message ☐ Enable ☒ Disable ?

Online version check ☒ Enable ☐ Disable ?

Proxyserver ☐ Enable ☒ Disable ?

[Save](#) [Abort](#) * required

NagiosQL 3.2.0

Worth mentioning are the following values:

- Temporary Directory:** The directory NagiosQL uses to store temporary files. Normally this will be **/tmp**.
- Server protocol:** If NagiosQL is installed in an SSL environment you can choose **https**. Setting an incorrect value might affect/end the operation!
- Language:** The NagiosQL interface might be changed to use another language. Sometimes the browser cache has to be cleared, the browser and/or Apache to be restarted so that the new setting takes effect.
- Authentication type:** If the Apache authentication is to be used this value has to be set accordingly.
- Data lines per page:** The number of lines in list views can be set here.
- Online version check:** The NagiosQL support page will show if a new NagiosQL version is available.
Environments using a proxy server will have to set the appropriate values.

2.2 Domain config (paths)

The different domains and paths of the Nagios installations to be controlled are determined using the domain administration. This is to be found below **Administration**.

There are two sorts of domains:

2.2.1 Data domains

Data domains are used for data storage, i.e. they contain the configuration objects. The selection field in the upper left of the view always shows the currently active data domain.

The screenshot shows the NagiosQL Administration interface. The top navigation bar includes 'Administration' and 'Data domains'. The left sidebar lists various administration options, with 'Administration' selected. The main content area is titled 'Data domain administration' and displays a table of data domains. The table has columns for 'Data domain', 'Description', 'Active', and 'Function'. Two domains are listed: 'common' (Global common domain) and 'localhost' (Local installation). Both are marked as 'Active'. Below the table, there is an 'Add' button and a 'Marked:' dropdown menu with a 'Do it' button. The bottom of the page shows the version 'NagiosQL 3.2.0'.

Data domain	Description	Active	Function
common	Global common domain	Yes	
localhost	Local installation	Yes	

There are two given data domains which cannot be deleted or renamed:

Common Data Domain

A general data domain which can be included into every other data domain. This might be useful for contacts, time periods, or commands, which are identical in every data domain. There are only very few settings available.

Localhost Data Domain

The data domain of the local server. Apart from the fact that this data domain cannot be renamed there is no difference to other data domains.

Configuration options

The screenshot shows the NagiosQL Administration interface. The top navigation bar includes 'Administration' and 'Data domains'. The left sidebar lists various administration tasks. The main content area is titled 'Data domain administration' and contains a form with the following fields:

- Domain ***: localhost
- Description ***: Local installation
- Configuration target ***: localhost
- Nagios version**: 3.0 +
- Use common domain**: Enable
- Access group**: Unrestricted access
- Active**: ☒

At the bottom of the form are 'Save' and 'Abort' buttons, and a note '* required'. The footer indicates 'NagiosQL 3.2.0'.

- Domain:** Internal name of the data domain (will be shown in the config views).
- Description:** Detailed description of the data domain
- Configuration target:** Selection of the target configuration domain (see 2.2.2).
- Nagios Version:** Version of the Nagios installation (version 2.x or 3.x are supported) – depending on the version additional options are displayed in the views.
- Use common domain:** The common data domain can be switched on or off. As soon as the common data domain is switched on and objects are defined, these objects will be written into the config files of the current data domain.
- Access group:** Select which user group gets access to this data domain.
- Active:** Define whether the data domain is active or not.

2.2.2 Configuration domain

Configuration domains are used to define a Nagios installation (site), they contain the paths and connection settings.

Configuration options

The screenshot shows the NagiosQL web interface for 'Configuration domain administration'. The interface has a top navigation bar with 'Administration' selected. A left sidebar contains a menu with options like 'Main page', 'Supervision', 'Alerting', 'Commands', 'Specialties', 'Tools', and 'Administration' (which is expanded to show sub-options like 'New password', 'User admin', 'Group admin', 'Menu access', 'Data domains', 'Config targets', 'Logbook', 'Settings', 'Help editor', and 'Support'). The main content area is titled 'Configuration domain administration' and contains several form fields for configuring a target system. The 'Domain' is set to 'localhost'. The 'Description' is 'Local installation'. The 'Server name' is 'localhost'. The 'Method' is 'SSH/SFTP'. The 'Username' is 'nagiosql_usr'. The 'Password' field is empty. The 'Directory with SSH key pair' is '/etc/nagiosql/ssh/'. Below these are 'Configuration directories' for base, host, service, backup, and host backup, all pointing to paths under '/etc/nagiosql/'. There are also fields for 'Nagios command file', 'Nagios binary file', 'Nagios process file', and 'Nagios config file *', with values like '/var/nagios/rw/nagios.cmd', '/opt/nagios/bin/nagios', '/var/nagios/nagios.lock', and '/etc/nagios/nagios.cfg'. The 'Nagios version' is set to '3.0 +' and the 'Access group' is 'Unrestricted access'. An 'Active' checkbox is checked. At the bottom, there are 'Save' and 'Abort' buttons, and a note '* required'.

- Domain:** Internal name of the configuration domain
(will be shown in data domain – config target)
- Description:** Detailed description of domain
- Server name:** Host name of the system running the Nagios installation of this domain – must be resolvable to an IP address
- Method:** The connection type specifies how NagiosQL will access the config files. Valid options are:
- Fileaccess The files are stored on the local file system of the server or are accessible from there (NFS/SMB/CIFS etc.)
 - FTP The config files are accessed via FTP
 - SSH/SFTP The config files are accessed via SSH/SFTP
- In the next chapter (2.3) we will give some short instructions on how to setup FTP and SSH/SFTP*
- User name:** User name to login onto the target system (only FTP or SSH/SFTP)
- Password:** Password to login onto the target system (only FTP or SSH/SFTP)
- Dir. to SSH key pair:** Local directory where the SSH key is stored (only SSH/SFTP)

Base directory:	Base directory for Nagios config files. All config files apart from those for hosts and services are stored in this folder. <i>/etc/nagiosql</i>
Host directory :	Directory for the Nagios host config files <i>/etc/nagiosql/hosts</i>
Service directory:	Directory for the Nagios service config files <i>/etc/nagiosql/services</i>
Backup directory:	Directory where NagiosQL will store backup versions of config files (except for hosts/services) <i>/etc/nagiosql/backup</i>
Host backup dir.:	Directory where NagiosQL will store backup versions of host config files <i>/etc/nagiosql/backup/hosts</i>
Service backup dir.:	Directory where NagiosQL will store backup versions of service config files <i>/etc/nagiosql/backup/services</i>
Nagios base dir.:	Base directory of the Nagios installation. The files nagios.cfg and cgi.cfg are stored in this directory and NagiosQL expects them to be there! <i>/etc/nagios</i>
Import directory:	Alternative import directory. Config files in this directory are offered to be imported as well <i>/etc/nagios/objects</i>
Nagios image dir.:	Alternative image directory. Normally the path to images is static in Nagios (e.g. <i>/usr/local/nagios/share/images/logos</i>). This directory can be used as a second base directory for additional files/directories. Other definitions using the folder are relative to this entry.
Nagios Command:	Path to the Nagios command file (nagios.cmd). This file is required to send external commands to the Nagios daemon. Please note that this file is only created / read when check_external_commands=1 is set in nagios.cfg ! <i>/var/nagios/rw/nagios.cmd</i>
Nagios Binary:	Path to the Nagios binary <i>/opt/nagios/bin/nagios</i>
Nagios Process file:	Path to the Nagios process file (PID or LOCK file) <i>/var/nagios/nagios.lock</i>
Nagios config file:	Path to nagios.cfg <i>/etc/nagios/nagios.cfg</i>
Nagios version:	Version of the Nagios installation (Version 2.x or 3.x are supported).
Access group:	Select which user group gets access to this data domain
Active:	Define whether the data domain is active or not

2.3 2.3 Setup remote access (FTP or SSH/SFTP)

2.3.1 Remote access via FTP

On the local system the PHP module **ftp** has to be installed.

On the target system an FTP server has to be installed. Additionally a separate user for NagiosQL is set up.

The NagiosQL directory structure as described in 1.2 will be created on the target system. Furthermore the permissions on the remote system have to be set up as described in 1.2 and 1.3. Instead of the web server daemon the NagiosQL FTP user has to get write permissions.

The credentials of the NagiosQL FTP user have to be filled into the fields user and password in the domain configuration (2.2).

Method	FTP
Username *	nagiosql_usr
Password *

It is recommended to check the FTP connection using an FTP client before you enter the values.

Using FTP the functionality in „Tools – Nagios control“ is limited. „Check configuration files“ and „Restart Nagios“ via FTP will only work on a server supporting REMOTE EXEC – but due to security limitations nowadays this is rarely seen.

2.3.2 Remote access via SSH/SFTP

On the local system the PHP module ssh2 has to be installed (1.5).

On the target system an SSH server has to be installed. Allow access (firewall etc.) from the local system to the target system.

Create a new user for NagiosQL on the target system, e.g. nagiosql_usr. Please note that some distributions may limit the length of user names. Then you have to create a key pair as follows.

Local system:

1. Create a path for the SSH keys:

```
# mkdir /etc/nagiosql/ssh
```
2. Create a key pair – only rsa is supported:

```
# ssh-keygen -t rsa
```

Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): /etc/nagiosql/ssh/id_rsa
-> leave the password empty!
The name of the key file must be **id_rsa**!
3. The key is to be copied to the target system:

```
# ssh-copy-id -i /etc/nagiosql/ssh/id_rsa/id_rsa.pub nagiosql_usr@remote-system
```
4. The key in /etc/nagiosql/ssh has to be readable for the web server daemon!

```
# chown -R wwwrun.nagios /etc/nagiosql/ssh
```


```
# chown -R wwwrun.nagios /etc/nagiosql/ssh/id_rsa
```
5. Please check login manually:

```
# ssh -i /etc/nagiosql/ssh/id_rsa nagiosql_usr@remote-system
```

-> Login has to be possible without entering a password!

The name of the NagiosQL SSH user (nagiosql_usr) will be set in the domain config (2.2) filling in “User name”. The field for the SSH key pair directory will hold the value of the directory (/etc/nagiosql/ssh/).

Method	SSH/SFTP
Username *	nagiosql_usr
Password *	
Directory with SSH key pair *	/etc/nagiosql/ssh/



The field “password” can be left empty because it is not used.

An authentication using user and password (without key) is possible but strongly discouraged because it is generally more error prone. If you still want to use it then you have to set the following value in sshd_config on the target system:

```
PasswordAuthentication yes
```

3 Nagios Configuration

3.1 Nagios Configuration file

The Nagios configuration file **nagios.cfg** has to be adapted so that Nagios can find and use the configuration files written by NagiosQL. This can be done using the integrated editor of NagiosQL, if the access permissions were set correctly (1.3).

Important are the paths of the standard config files:

```
cfg_file=/etc/nagiosql/contacttemplates.cfg
cfg_file=/etc/nagiosql/contactgroups.cfg
cfg_file=/etc/nagiosql/contacts.cfg
cfg_file=/etc/nagiosql/timeperiods.cfg
cfg_file=/etc/nagiosql/commands.cfg
cfg_file=/etc/nagiosql/hostgroups.cfg
cfg_file=/etc/nagiosql/servicegroups.cfg

cfg_dir=/etc/nagiosql/hosts
cfg_dir=/etc/nagiosql/services
```

Some additional config files if they are used:

```
cfg_file=/etc/nagiosql/hosttemplates.cfg
cfg_file=/etc/nagiosql/servicetemplates.cfg
cfg_file=/etc/nagiosql/servicedependencies.cfg
cfg_file=/etc/nagiosql/serviceescalations.cfg
cfg_file=/etc/nagiosql/hostdependencies.cfg
cfg_file=/etc/nagiosql/hostescalations.cfg
cfg_file=/etc/nagiosql/hostextinfo.cfg
cfg_file=/etc/nagiosql/serviceextinfo.cfg
```

All existing lines starting with **cfg_file=** or **cfg_dir=** have to be **removed** or **deactivated** using a leading hash sign (#)! There must be no other cfg lines except the ones mentioned above.

Restarting Nagios from NagiosQL requires that Nagios has to accept external commands which is controlled by the following line in nagios.cfg:

```
check_external_commands=1
```

3.2 Additional permissions

For complete functionality of NagiosQL some Nagios files have to be accessible for NagiosQL (see 1.3). For local installations the user running the web server daemon has to be permitted, remote installations require the NagiosQL FTP or SSH/SCP user to be permitted. Subsequently the guide uses the term „user“.

The paths to these files have to be set in the domain configuration (see 2.2). Regarding the permissions there are several ways (file permissions, sudo, acl). Furthermore the installations differ individually.

3.2.1 Check command

To check the curring configuration NagiosQL will use the internal verify method of Nagios:

```
# nagios -v nagios.cfg
<nagios-binary> -v <nagios configuration file>
```

The user must have permissions to execute the Nagios binary and read permissions on **nagios.cfg**.

3.2.2 Configuration editor

The files **nagios.cfg** and **cgi.cfg** can be edited directly from within NagiosQL using a simple editor.

The user must have write permissions on the config files **nagios.cfg** and **cgi.cfg**.

3.2.3 Restart command

To activate a new configuration directly from within NagiosQL permissions on the command file **nagios.cmd** and the process file **nagios.lock** are required.

The user must have write permissions on the command file **nagios.cmd** and read permissions on the process file **nagios.lock**, or **nagios.pid**, respectively.

3.3 Support page

Since NagiosQL 3.2.0 we have a support page running the most important internal checks and showing the results. Common misconfigurations are detected quickly and you are able to solve your problems.

3.3.1 Common checks

NagiosQL support page

Support contact information

For questions, the online support forum or contact information visit our website:
[NagiosQL](#)

Donate to support NagiosQL

If you like NagiosQL and it simplifies your daily work, then you may want to support the project by making a donation. This helps us to keep NagiosQL alive and to cover our costs. Thank you for your donation!
[Donate for NagiosQL on sourceforge](#)

Translation services

NagiosQL was translated into various languages. Since some translators are no longer available in later versions, there may be untranslated words or phrases. If you want to help us complete the translation, correct them or introduce a new language, then sign up with us now! The translations are simply feasible online - we use an open translation service where you can register for free at any time:
[Transifex translation service](#)

Version check

The online version check connects the NagiosQL page to find out, if your version is still up to date.

Installed	Available	Information
3.2.0	3.2.0	You already have the latest version installed

Environment check

Javascript	ok	PHP ini settings: file_uploads	ok
PHP version	ok (5.3.8)	PHP ini settings: session.auto_start	ok
PHP module: session	ok	PHP ini settings: suhosin.session.encrypt	ok
PHP module: gettext	ok	PHP ini settings: date.timezone	ok (Europe/Zurich)
PHP module: ftp	ok	Read/Write access: settings.php	ok
PHP module: ssh	ok		
MySQL version	ok (5.5.16-log)		

You will find some helpful links. Basic environment checks are executed. If something doesn't work after a system upgrade you might find a hint.

If enabled it will be checked if the most current version of NagiosQL is installed.

3.3.2 Config checks

Config domain checks

The checks below are based on your data domain and config domain settings. To change the data domain, use the pull down menu in the upper right corner. Repeat this check for any data domain you have configured. To change the config domain, use the data domain menu and select a different config domain value.

Config domain name	localhost	Nagios config file	ok
Connection type	Fileaccess	Nagios process file	ok
Connection check	ok	Nagios command file	ok
		Nagios binary file	ok

Verify configuration files and demon configuration

Configuration name	Used in data domain	Included in demon configuration (nagios.cfg)
Hosts	ok (4)	ok (cfg_dir=/opt/nagios/inst1/etc/nagiosql/hosts)
Services	ok (21)	ok (cfg_dir=/opt/nagios/inst1/etc/nagiosql/services)
Hostgroups	ok (4)	ok (cfg_file=/opt/nagios/inst1/etc/nagiosql/hostgroups.cfg)
Servicegroups	not used	ok (cfg definition missed, but actually not used)
Hosttemplates	ok (5)	ok (cfg_file=/opt/nagios/inst1/etc/nagiosql/hosttemplates.cfg)
servicetemplates	ok (2)	ok (cfg_file=/opt/nagios/inst1/etc/nagiosql/servicetemplates.cfg)
Contacts	ok (1)	ok (cfg_file=/opt/nagios/inst1/etc/nagiosql/contacts.cfg)
Contactgroups	ok (1)	ok (cfg_file=/opt/nagios/inst1/etc/nagiosql/contactgroups.cfg)
Timeperiods	ok (5)	ok (cfg_file=/opt/nagios/inst1/etc/nagiosql/timeperiods.cfg)
Contacttemplates	ok (1)	ok (cfg_file=/opt/nagios/inst1/etc/nagiosql/contacttemplates.cfg)
Commands	ok (24)	ok (cfg_file=/opt/nagios/inst1/etc/nagiosql/commands.cfg)
Hostdependencies	not used	ok (cfg definition missed, but actually not used)
Hostescalations	not used	ok (cfg definition missed, but actually not used)
Hostextinfo	not used	ok (cfg definition missed, but actually not used)
Servicedependencies	not used	ok (cfg definition missed, but actually not used)
Serviceescalations	not used	ok (cfg definition missed, but actually not used)
Serviceextinfo	not used	ok (cfg definition missed, but actually not used)

NagiosQL 3.2.0

It will be checked if all relevant Nagios files are available and if the user permissions are correct (see 1.3).

The last part processes the Nagios config file (nagios.cfg) and checks if the necessary path settings (see 3.1) were made and if they are equal to the ones in the selected config domain.

Generally all red entries on the support page indicate a problem which must be solved. Yellow entries indicate problems which might pop up later.