Authentication specific options

The following configuration options are available under *config/authentication.php*:

General

AUTH_ENABLED

Enable or disable user authentication. If disabled, no login is prompted to access the GUI and all features will be available (admin privs).

```
Possible values: "0", "1"
Default value:   "0"
```

AUTH_TYPE

How to authenticate, if authentication is enabled.

```
Possible values:
"file"     - authenticate using a textfile with "user::pass" pairs
"ldap"     - authenticate using an LDAP server
"ad_ldap"  - authenticate using an Active Directory LDAP server
"sql"      - authenticate using an SQL database
Default value: "file"
```

AUTH_METHOD

Select the login authentication method.

```
Possible values:
"login"    - authenticate using the ordinary NConf login form
"basic"    - authenticate using HTTP Basic Authentication (pop-up window for username/password)
Default value:   "login"
```

Using HTTP Basic Auth can be useful if you want users to only authenticate once for both NConf and Nagios.

BASICAUTH_REALM

The HTTP Basic Auth Realm to display when using auth method “basic”.

```
Possible value: "your text"
Default value:   "NConf Basic Auth"
```

AUTH_FEEDBACK_AS_WELCOME_NAME

This defines the user name in the history table and in the welcome message.

```
Possible values: "0", "1"
Default value:   "0"
```

If set to “0”, the username will be used. If set to “1”, the real name will be fetched, depending on which AUTH_TYPE you selected.

- **file**:
the last attribute will be the user's full name (details in Auth by File)
- **ldap**:
the “cn” attribute will be user's full name (details in Auth by LDAP)
- **ad_ldap**:
the configured AD_USERNAME_ATTRIBUTE (default: *displayname*) attribute will be user's full name (details in Auth by AD LDAP)
- **sql**:
the result returned from your SQL query will be the user's full name (details in Auth by SQL)

LOG_REMOTE_IP_HISTORY

Enable / disable logging of the remote-IP / hostname to the history.
If set to “1”, the remote-IP is written to the history after a user logs in.
In case “HostnameLookups” is set to On in the apache config, the hostname will be used instead.

```
Possible values: "0", "1"
Default value:   "1"
```

Group

GROUP_USER

When NConf parses the output from one of the authentication modules, it will look for this pattern to determine if an account should be regarded as an ordinary user.

```
Default value: "user"
```

GROUP_ADMIN

When NConf parses the output from one of the authentication modules, it will look for this pattern to determine if an account should be regarded as an admin account.

```
Default value: "admin"
```

GROUP_NOBODY

Do not change this

```
Default value: "0"
```

Types

- Auth by File
- Auth by LDAP
- Auth by Active Directory
- Auth by SQL
- Auth by NConf contacts

Auth by File

When using “Auth by File”, make sure your PASSWD_ENC constant matches the password encryption you are using in your user account file.
The account file is stored under:

- `config/.file_accounts.php`

You can manage users by simply adding more rows. The syntax is:

```
username::password::authorization(user|admin)::[[user's|full name (optional)]]::
```

For example, this is a basic user:

```
john::1234::user::John Smith::
```

Make sure the pattern ”::” does not appear in any of the data fields!

Changes as of NConf 1.2.5

- The delimiter has been changed to ”::” (2 colons)
- The file 'config/.file_accounts' is now a PHP file: 'config/.file_accounts.php'

encryption

If you want to use encrypted passwords, setup your accounts as follows:
Each encryption has its own TYPE definition in brackets, in front of the encrypted password.
This is an example for **crypt**

```
# using encrypted passwords
user2::{CRYPT}s7FkIgzTWZia2::user::User with a CRYPT password::
```

quick help

crypt

1. create your crypt password for a user
   - for example using openssl:

     ```
     openssl passwd YOUR_PASSWORD_HERE
     ```

2. This will generate you a random string:

   ```
   WP8CFXlYfGOJ6
   ```

3. Use this in the password file this way:
   - `{CRYPT}WP8CFXlYfGOJ6`
   - example file row:

     ```
     user2::{CRYPT}WP8CFXlYfGOJ6::user::full name::
     ```

4. save the file and try to log in in NConf with the created user

md5

1. create your crypt password for a user
    - for example using openssl:

      ```
      echo -n YOUR_PASSWORD_HERE | openssl md5
      ```

2. This will generate you a random string:

   ```
   098f6bcd4621d373cade4e832627b4f6
   ```

3. Use this in the password file this way:
    - `{MD5}098f6bcd4621d373cade4e832627b4f6`
    - example file row:

      ```
      user2::{MD5}098f6bcd4621d373cade4e832627b4f6::user::full name::
      ```

4. save the file and try to log in in NConf with the created user

sha1

1. create your crypt password for a user
    - for example using openssl:

      ```
      echo -n YOUR_PASSWORD_HERE | openssl sha1
      ```

2. This will generate you a random string:

   ```
   a94a8fe5ccb19ba61c4c0873d391e987982fbbd3
   ```

3. Use this in the password file this way:
    - `{SHA1}a94a8fe5ccb19ba61c4c0873d391e987982fbbd3`
    - example file row:

      ```
      user2::{SHA1}a94a8fe5ccb19ba61c4c0873d391e987982fbbd3::user::full name::
      ```

4. save the file and try to log in in NConf with the created user

nconf/help/documentation/detail/authentication/file.txt · Last modified: 09.12.2011 01:48 (external edit)

Auth by LDAP

When using Auth by LDAP, make sure your PASSWD_ENC constant ist set to “clear”, regardless of the password encryption you are actually using in LDAP.

LDAP_SERVER

The LDAP connection string, with or without `"[ldap[s]://]"`. LDAP v3 is required.

Your LDAP tree design (DIT) must be pam_ldap / nss_ldap compliant, meaning the attributes and the structure you use must be the same ones that PAM would require.

```
Usage: "[ldap[s]://]hostname"
Default value: "ldaps://ldaphost.mydomain.com"
```

LDAP_PORT

The LDAP port to connect to. This constant is ignored when using URL notation in the LDAP_SERVER constant.

```
Default value: "389"
```

BASE_DN

The “base dn” to where the user entries are located in LDAP. ”<username>” is a placeholder and can be configured with the USER_REPLACEMENT constant.

```
Default value: "uid=<username>,ou=People,dc=mydomain,dc=com"
```

USER_REPLACEMENT

This constant defines the placeholder which is to be replaced by the username of the actual user that is logging in.

```
Default value: "<username>"
```

GROUP_DN

The “dn” to where the groups are located in LDAP.

```
Default value: "ou=Group,dc=mydomain,dc=com"
```

USER_GROUP

The name of the ordinary “user group”. Any user who wants to access NConf, and is not an admin, has to be in this LDAP group. Users, who are whether in the USER_GROUP nor in the ADMIN_GROUP will not be able to access NConf.

```
Default value: "cn=sysadmin"
```

ADMIN_GROUP

The name of the “admin group”. Users who want to be “NConf admin” have to be in this LDAP group. This group should only be assigned to NConf superusers. If a user is in the admin group, he does not need to be in the USER_GROUP as well.

```
Default value: "cn=nagiosadmin"
```

Auth by Active Directory

New feature introduced with NConf 1.3

When using Auth by Active Directory LDAP, make sure your PASSWD_ENC constant ist set to “clear”, regardless of the password encryption you are actually using in AD LDAP.

Active Directory will handle this by itself.

Default config

```
### Active Directory
define('AD_LDAP_SERVER',        "ldap://ad-ldaphost.mydomain.com");
define('AD_LDAP_PORT',          "389");
define('AD_BASE_DN',            "CN=<username>,OU=All,OU=Users,DC=my,DC=domain,DC=com");
define('AD_USER_REPLACEMENT',   "<username>");
define('AD_GROUP_ATTRIBUTE',    "memberof");
define('AD_USERNAME_ATTRIBUTE', "displayname");
define('AD_GROUP_DN',           "OU=Group,DC=my,DC=domain,DC=com");
define('AD_ADMIN_GROUP',        "CN=nagiosadmin");
define('AD_USER_GROUP',         "CN=sysadmin");

# if AD_GROUP_DN differs for admins and users:
# you can define FIX GROUPS: (needs empty GROUP_DN)
//define('AD_GROUP_DN',         "");
//define('ADMIN_GROUP',         "CN=nagiosadmin,OU=Group,DC=my,DC=domain,DC=com");
//define('USER_GROUP',          "CN=sysadmin,OU=Group,DC=my,DC=domain,DC=com");
```

Description of config values

You should know your AD LDAP tree design (DIT) and configure the config values accordingly.

AD_LDAP_SERVER

The AD LDAP connection string, with or without `"[ldap[s]://]"`.
LDAP v3 is required.

```
Usage: "[ldap[s]://]hostname"
Default value: "ldap://ad-ldaphost.mydomain.com"
```

AD_LDAP_PORT

The LDAP port to connect to. This constant is ignored when using URL notation in the AD_LDAP_SERVER constant.

```
Default value: "389"
```

AD_BASE_DN

The “base dn” to where the user entries are located in the Active Directory.
”*<username>*” is a placeholder and can be configured with the AD_USER_REPLACEMENT constant. *(basically you do not have to change this, nconf will put your login username in there)*

```
Default value: "CN=<username>,OU=All,OU=Users,DC=my,DC=domain,DC=com"
```

AD_USER_REPLACEMENT

This constant defines the placeholder which is to be replaced by the username of the actual user that is logging in.

```
Default value: "<username>"
```

AD_GROUP_ATTRIBUTE

This constant defines the attribute which holds the groups of your account. Change this if you want to use an other attribute to locate the groups.

```
Default value: "memberof"
```

AD_USERNAME_ATTRIBUTE

This constant defines the attribute which holds the users full name. Change this if you want to use an other attribute.

```
Default value: "displayname"
```

AD_GROUP_DN

The basic “dn” of your groups. You can define this if your user and admin group is located on the same dn.

```
Default value: "OU=Group,DC=my,DC=domain,DC=com"
```

ADMIN_GROUP

The name of the “admin group”. Users who want to be “NConf admin” have to be in this group. This group should only be assigned to NConf superusers. If a user is in the admin group, he does not need to be in the USER_GROUP as well.

Define your Group name like this, if your admin/user groups are located in the same DN

```
Default value: "CN=nagiosadmin"
```

If the BASE of the user group differs from the admin group, you could enter the whole DN like this:

```
Default value: "CN=nagiosadmin,OU=group,DC=my,DC=domain,DC=com"
```

USER_GROUP

The name of the ordinary “user group”. Any user who wants to access NConf, and is not an admin, has to be in this group. Users, who are whether in the USER_GROUP nor in the ADMIN_GROUP will not be able to access NConf. Define your group name like this, if your admin/user groups are located in the same DN

```
Default value: "CN=sysadmin"
```

If the BASE of the user group differs from the admin group, you could enter the whole DN like this:

```
Default value: "CN=sysadmin,OU=different_group,DC=my,DC=domain,DC=com"
```

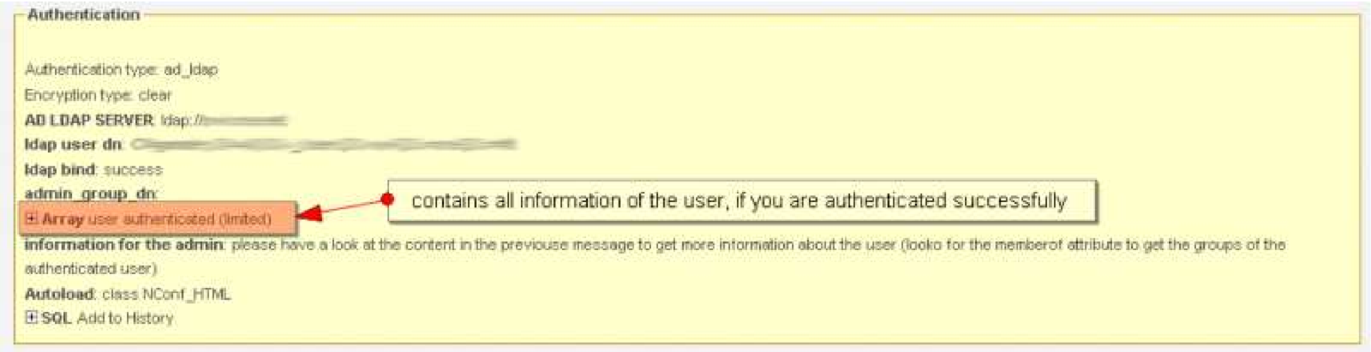
showing AD information of your user

If you want to display all available information of your user, you can do the following:
(although you should know your AD structure and perhaps also created 2 groups for NConf admins and users)
This could help you to find the correct DN of your groups.

1. activate NConf debugging
2. configure all other AD config variables, but define the AD_ADMIN_GROUP and AD_USER_GROUP as empty

   ```
   define('AD_ADMIN_GROUP',         "");
   define('AD_USER_GROUP',          "");
   ```

3. Login with AD user and have a look at the debug section at the bottom



Authentication

Authentication type: ad_ldap
Encryption type: clear
AD LDAP SERVER: ldap://[redacted]
ldap user dn: [redacted]
ldap bind: success
admin_group_dn:
⊞ **Array** user authenticated (limited)
information for the admin: please have a look at the content in the previouse message to get more information about the user (looko for the memberof attribute to get the groups of the authenticated user)
Autoload: class NConf_HTML
⊞ **SQL** Add to History

contains all information of the user, if you are authenticated successfully

nconf/help/documentation/detail/authentication/ad_ldap.txt · Last modified: 09.12.2011 01:48 (external edit)

Auth by SQL

When using Auth by SQL, make sure your PASSWD_ENC constant matches the password encryption you are using in your SQL database.

NConf contacts authentication

If you want to use your contacts in NConf for authentication, you do not have to configure the following AUTH_ DB settings. NConf will use your NConf DB.

For more information about how to authenticate using **NConf contacts**, have a look at the following How-to: NConf contacts authentication

external DB authentication

You can use the following constants to configure a different MySQL server with your own accounting queries.

AUTH_DBHOST

The hostname or IP address of the MySQL server.

```
Usage: "hostname[[:port]]" or "IP[[:port]]"\\
Default value: "localhost"
```

AUTH_DBNAME

The name of the MySQL database for authentication.

```
Default value: "NConf"
```

AUTH_DBUSER

The name of the database user.

```
Default value: "nconf"
```

AUTH_DBPASS

The password for AUTH_DBUSER.

Custom SQL query

You can define your own SQL queries for NConf authentication to be run in any available user database.

The query should return exactly one (1) record if:

- the username exists

- the password is correct
- any additional attrs are set (optional for permission check etc.)

!!!USERNAME!!! and *!!!PASSWORD!!!* are placeholders that will be replaced with the username and password from the login page.

Example The following is an example database structure for better understanding:

```
+---------+----------+-------+--------------------+
| account | password | group | username           |
+---------+----------+-------+--------------------+
| admin   | nconf    | admin | NConf Administrator |
| user    | 1234     | user  | Normal User        |
+---------+----------+-------+--------------------+
```

By default, NConf comes predefined with two queries (AUTH_SQLQUERY_USER and AUTH_SQLQUERY_ADMIN) which allow you to authenticate using the NConf database as authentication source itself (more precisely: your NConf contact items). If you wish to use a different MySQL database for authentication, feel free to modify the queries appropriately.

AUTH_SQLQUERY_USER

The query which identifies the user as an ordinary user:

Example query:

```
SELECT username FROM users
WHERE `account` = "!!!USERNAME!!!"
AND   `password` = "!!!PASSWORD!!!"
AND   `group`    = "'.GROUP_USER.'"
```

AUTH_SQLQUERY_ADMIN

The query which identifies the user as an admin user. This query is optional, and will be run after the USER query, to check for possible users with higher privileges.

Example value:

```
SELECT username
FROM users
WHERE `account`= "!!!USERNAME!!!"
AND   `password`= "!!!PASSWORD!!!"
AND   `group`   = "'.GROUP_ADMIN.'"
```

nconf/help/documentation/detail/authentication/sql.txt · Last modified: 09.12.2011 01:48 (external edit)

NConf contacts authentication

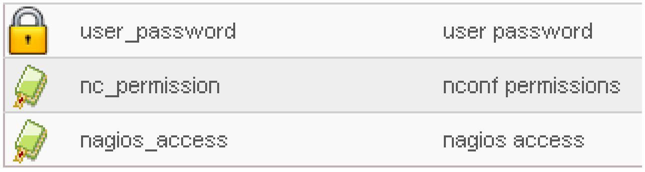
NConf offers the possibility to use any external MySQL database for user authentication. There are multiple configuration options which can be used to define the target database, the credentials, as well as your own customized SQL queries. Refer to Auth by SQL to understand how to configure SQL authentication.

A powerful feature of NConf is that it also has built-in functionality to manage user accounts within NConf itself. This enables you to control users, passwords and privileges for NConf and Nagios directly within NConf. Doing this involves telling the NConf authentication mechanism that you wish to authenticate using the NConf database as auth source (more precisely: the 'contact' items within NConf).

Manage NConf access

There are 3 predefined attributes which must be activated first, in order to manage user passwords and permissions:

1. In NConf, go to the menu "**Administration**" > "**Attributes**" > "**Show**" and select the "**contact**" attributes.
2. Edit the following attributes and set "*attribute is visible = yes*" for each one:
    - user_password
    - nc_permission
    - nagios_access

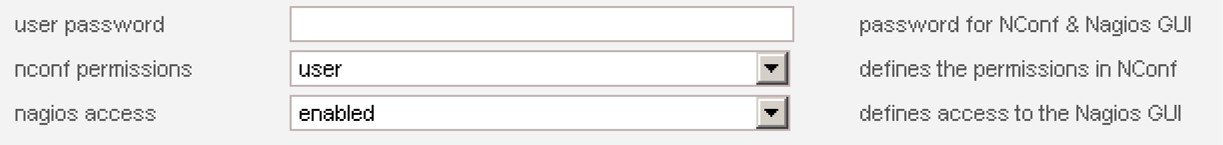


🔒	user_password	user password
	nc_permission	nconf permissions
	nagios_access	nagios access

3. Now go to "**Additional Items**" > "**Contacts**":
    - set a password for your own user, to make sure you can still log in later.
    - Also give yourself "admin" rights.
4. In *config/authentication.php*, set **AUTH_TYPE** to "*sql*"

```
define('AUTH_TYPE', "sql");
```

    - Because you want to use the NConf DB, you do not have to configure the other AUTH_ settings (*AUTH_DBHOST,AUTH_DBNAME,AUTH_DBUSER,AUTH_DBPASS*), just leave it commented.
    - also don't change any of the **AUTH_SQLQUERY** constants.
      They will tell NConf to authenticate your **contacts**.
5. Now you can set a password and the permissions for each user by editing the contacts in NConf



user password		password for NConf & Nagios GUI
nconf permissions	user	defines the permissions in NConf
nagios access	enabled	defines access to the Nagios GUI

Add contact with the new attributes

- Next time you log in, the NConf *contacts* will be used for authentication.

Password encryption

You might also want to enable password encryption in 'config/nconf.php'. Do this right from the start, because once you have already set several passwords, and you later decide to change the encryption type, you will have to modify all stored passwords. For further information, refer to Configuration: Password attributes.

How to proceed for changing encryption

1. Log in on NConf with admin rights
2. Change the encryption type in `config/nconf.php` and save the file
3. Go to your contacts and edit the users password or create them
4. you can verify the encryption by looking for the password in the detail view
5. the most secure way to test the log in:
    - open a different browser and try to login
6. otherwise, just log out and log in again

enter password for user

Enter the plain password for the user or admin without the {} tags.

NConf will automatically:

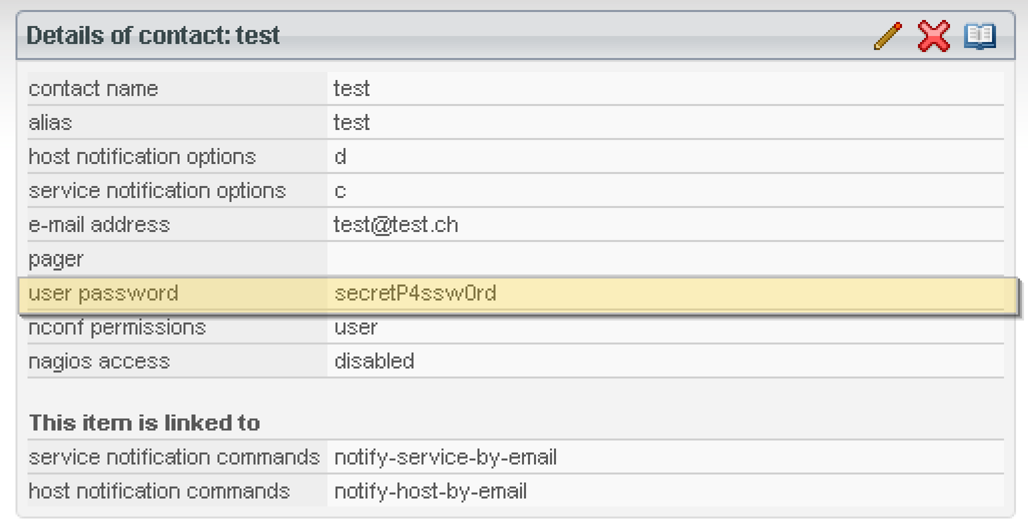
- Add the {…} *(encryption type)*
- Encrypt the password

→ *see screenshots*

screenshots

Here you see some screenshots on how the password field should look regarding the encryption type.

clear



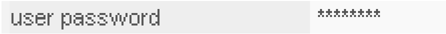
Details of contact: test

contact name	test
alias	test
host notification options	d
service notification options	c
e-mail address	test@test.ch
pager	
user password	secretP4ssw0rd
nconf permissions	user
nagios access	disabled

This item is linked to

service notification commands	notify-service-by-email
host notification commands	notify-host-by-email

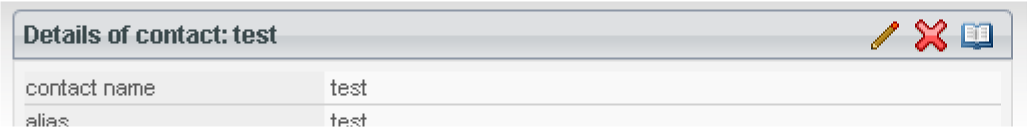
To see the passwords plain text, this would be not acceptable for most admins.
Therefore NConf gives you a way to hide them, please have a look here:



user password	********

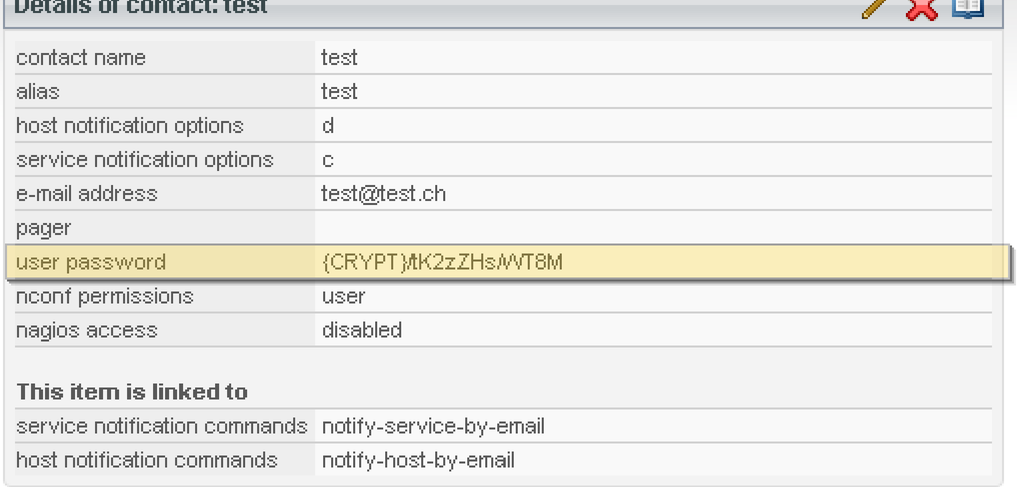
- Password attributes documentation for the **PASSWD_DISPLAY** part.

crypt



Details of contact: test

contact name	test
alias	test



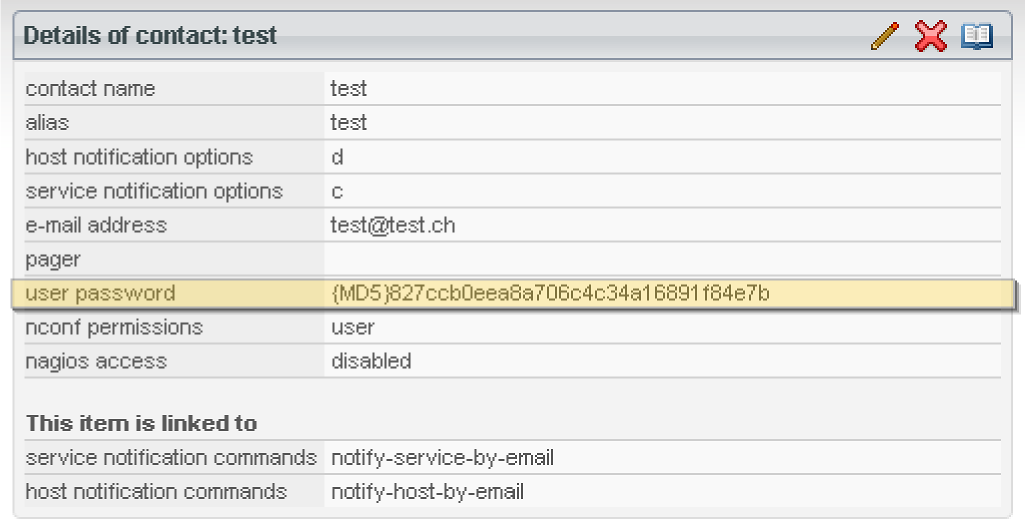
Details of contact: test

contact name	test
alias	test
host notification options	d
service notification options	c
e-mail address	test@test.ch
pager	
user password	{CRYPT}/tK2zZHs/WT8M
nconf permissions	user
nagios access	disabled

This item is linked to

service notification commands	notify-service-by-email
host notification commands	notify-host-by-email

md5



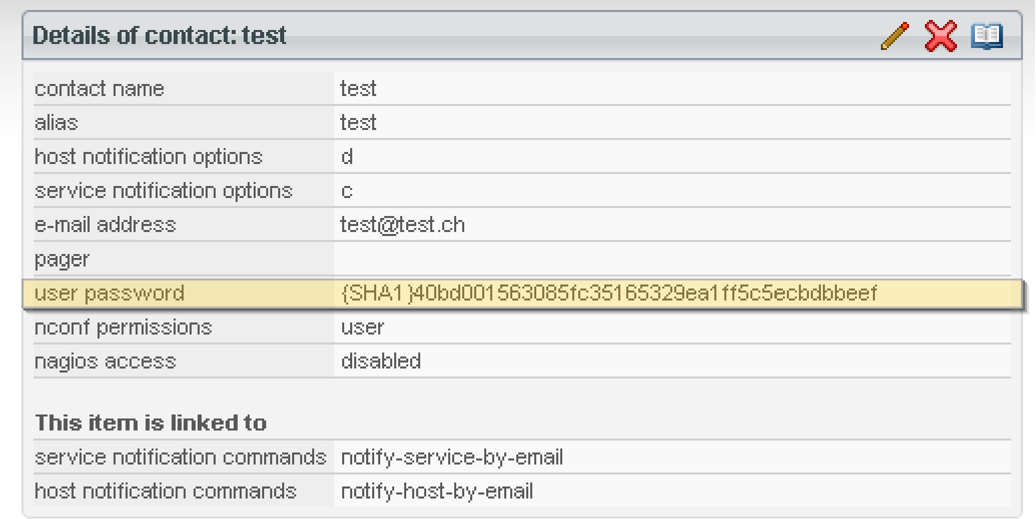
Details of contact: test

contact name	test
alias	test
host notification options	d
service notification options	c
e-mail address	test@test.ch
pager	
user password	{MD5}827ccb0eea8a706c4c34a16891f84e7b
nconf permissions	user
nagios access	disabled

This item is linked to

service notification commands	notify-service-by-email
host notification commands	notify-host-by-email

sha1



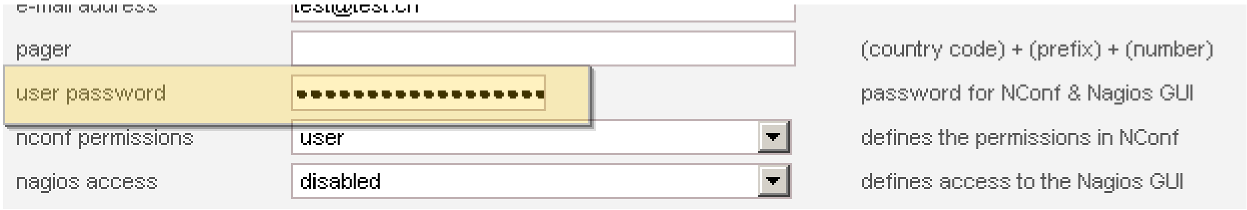
Details of contact: test

contact name	test
alias	test
host notification options	d
service notification options	c
e-mail address	test@test.ch
pager	
user password	{SHA1}40bd001563085fc35165329ea1ff5c5ecbdbbeef
nconf permissions	user
nagios access	disabled

This item is linked to

service notification commands	notify-service-by-email
host notification commands	notify-host-by-email

modify contact



e-mail address	test@test.ch	
pager		(country code) + (prefix) + (number)
user password	••••••••••••••••••••	password for NConf & Nagios GUI
nconf permissions	user	defines the permissions in NConf
nagios access	disabled	defines access to the Nagios GUI

On the modify pages, you will see the password in a password input field.
For changing the password, enter a new one.
Otherwise just leave it as is.

Manage Nagios webaccess

As soon as you start using password attributes for your contacts, NConf will start generating a file called 'global/nagios.htpasswd'. This file is part of the generated output. It is a standard password file for Apache which you can use on your Nagios server to authenticate your users.

If you would like to use the generated .htpasswd file to control access to your Nagios webaccess, you must set the encryption type to either “crypt” or “sha_raw”, because that is what Apache supports in .htpasswd files (“sha_raw” is what we call the implementation of SHA1 that Apache supports).