

Changing Database Passwords in GroundWork Monitor Enterprise Edition 6.3

- [Background](#)
- [Overview](#)
- [Process](#)
- [Testing](#)
- [Recovering](#)
- [REFERENCES](#)
 - [Scripts](#)
 - [Databases, Files, Hosts and Users](#)

Background

Certain applications of GroundWork Monitor Enterprise Edition may be in locations where security is an issue. As GroundWork uses MySQL database engine to store configuration, collected data, and other information which may be confidential, it might be thought necessary to protect the means of access.

As delivered, the MySQL engine and the various databases are controlled by user name password pairs, which exist in the main "mysql" database and in properties files referenced by executed modules.

The main 'root' access to the 'mysql' engine is password protected, if the customer follows instructions at install time. While customer may not set this as a matter of convenience, our consistent recommendation is always to install a root password. The procedures for setting this password, resetting it, or recovering it are all thoroughly described by MySQL documentation, e.g. <http://dev.mysql.com/doc/refman/5.0/en/resetting-permissions.html>.

The subject of this How To article covers the procedure for changing passwords for users of the included databases to some other values.

Overview

You change the passwords in the MySQL database, and in the properties files and other referencing scripts where the passwords are kept. You must do this all at the same session, with the various programs in a non running state. If you try to change these while gwservices and the other programs are running you may loose some data, or deny access to data due to permissions errors. Take care in making these changes that you do not lock yourself out of a particular database or application.

If you make these changes you should report the fact to GroundWork Support when filing any subsequent Support case. Our ability to provide support for our products requires knowledge of such changes. Your cooperation is necessary to minimize time wasted because of undocumented changes.

Process

This procedure is provided with the expectation that you do not wish to put passwords into a script to automate this process. We expect you will instead enter the passwords at the command line interactively, and this practice is more secure.

1. Stop all groundwork services but restart the mysql demon. To do this, as user root execute the following commands:

```
service groundwork stop
service groundwork start mysql
```

2. Create backups of the mysql and all included databases according to the process described in the Bookshelf. For example for the database called "mysql":

```
source /usr/local/groundwork/scripts/setenv.sh
mysqldump mysql -u root -p > /tmp/mysql.sql
```

Save these backup files in case you corrupt some essential table and must then want to restore the system to a known good state. Refer to the Bookshelf for complete instructions for restoring databases.

3. Create a backup of the entire "/usr/local/groundwork/config" directory and the other files listed in the reference section that follows. The attached script "make-backup-files.sh" will do this for you, putting the results in a datestamp named tar ball in /usr/local/groundwork/tmp. If you prefer to do it yourself, you might use a series of commands like this:

```
cd /usr/local/groundwork/  
cp -Rp config config-backup
```

If you make a serious error and need to restore a file, rename the working copy, and then move the "config-backup" into place as "config". Work with the other backup files in an appropriate manner. Remember to stop the GroundWork services before you do this copying and moving.

4. Access to each database is controlled by a User, Host, Password, Rights combination. Use the attached script "change-pass.pl", as the root user, to change the passwords for each user on a global basis. You will be prompted for the root password to access the mysql database when you start. For each host/user/database combination you will be prompted for a new password to set for that user. If you wish to keep the existing one just press Enter without any other keystrokes.
5. Now, edit the properties files which are listed in the Reference section. In each of the files you will find a section with a user name and password. Samples are noted in the reference below. Repeat the last two steps for each database you wish to change.
6. Once you have changed all the entries restart the services you stopped:

```
service groundwork start
```

This may complain about mysqld already running, which may be ignored.

Testing

Test the changes you have made by accessing the different functions. If a particular function has not been set up on your system, you may skip testing it.

1. Use the portal. Open it and navigate to the Administration tab. If you can do this the "jboss" databases are accessible.
2. Use the Configuration tool. Open the portal, choose Configuration, and then Hosts. If you can see the list of hosts the "monarch" database is accessible.
3. Use the Status Viewer. Open the portal, choose Status, and click on the Network tree to expand hostgroups and hosts. If you can see monitoring system details for your hosts and services the "GWCollageDB" database is accessible.
4. Open the "fping" feeder log file and note the timestamps and any error messages in it. Test to see that the "fping" processes are running with "ps". Fresh timestamps, lack of error messages, and an entry similar to this indicate that the password change was successful:

```
--- NEW LOOP on Saturday, December 25, 2010 at 09:46:23 ---  
Monarch query time = 0s  
IP=127.0.0.1  
HOST=localhost  
parent=none  
fping time was 0s  
inserted into array: IP=127.0.0.1, State=[unknown], Perf=[unknown]  
IP = 127.0.0.1, state = [unknown]  
sending host check for ip 127.0.0.1, host localhost  
Sending 0 results at 1293299183.  
1 data packet(s) sent to host successfully.  
Nagios submit time was 0s
```

5. Look in the nagios user cron table for the two jobs that run around midnight:

```
50 23 * * * /usr/local/groundwork/core/reports/utills/dashboard_data_load.pl >  
/usr/local/groundwork/core/reports/utills/log/dashboard_data_load.log 2>&1  
0 1 * * * /usr/local/groundwork/core/reports/utills/dashboard_avail_load.pl >  
/usr/local/groundwork/core/reports/utills/log/dashboard_avail_load.log 2>&1
```

Run each of them as user nagios and look at the indicated log files to see if there were connectivity problems with the "dashboard" database.

6. Look at the Reports. You should be able to open each report and see valid, recent data.
7. If you are running Cacti, check the portal, Advanced, Cacti tab. Open the System Utilities tab, then the Cacti Log, and look for error messages relating to the database. If there are no errors, and the log indicates that Cacti has completed the cycle of processing within the last 5 minutes, the password change is successful.
 - a. Also run the "check_cacti.pl" script in accordance with the manner it is set up (as a demon, as a plugin, or as a cron job). Look in

- the produced log file to see whether there were database connection errors which would indicate a failure; correct the issue.
8. If you are running Nedi, initiate a manual run by typing `'/usr/local/groundwork/nedi/nedi.pl --t 127.0.0.1'` as user nagios. You should not see an error having to do with failed database connection. If you see an error it means you did not properly change the password.
 9. In the unlikely event you are using the deprecated logreporting subsystem, there is a crontab entry for the import facility which runs the script `"/usr/local/groundwork/log-reporting/bin/importer.pl"`. Run it using the syntax from the cron job as user nagios; if there is an error about connecting to the database you have not successfully changed the password.

This completes the procedure for changing passwords on the database driven facilities in GroundWork Monitor Enterprise Edition 6.3

Recovering

If you find that you have made an error and wish to recover to the state prior to the changes, do this:

1. Stop all the services as above
2. Rename the directory `/usr/local/groundwork/config` to something else
3. Rename the backup directory to `/usr/local/groundwork/config`
4. Copy the backup files you modified in OTHER locations to their original locations.
5. Restore, at a minimum, the 'mysql' database from the backup. Here is the syntax:

```
source /usr/local/groundwork/scripts/setenv.sh
mysql mysql -u root -p < NAME-OF-BACKUP
```

6. Start all the services

```
service groundwork start
```

REFERENCES

Scripts

Name	Size	Creator	Creation Date	Comment
 change-pass.pl	2 kB	Hans Kriel	Dec 26, 2010 16:06	tool to change password
 make-password-backup-files.sh	0.6 kB	Hans Kriel	Dec 26, 2010 16:07	makes backups of properties files

Databases, Files, Hosts and Users

These are the database, file, user, and host combinations for which you may wish to change access. If you have installed distributed servers you may need to change more than is described here.

1. GWCollageDB
 - a. Files

```
/usr/local/groundwork/config/db.properties
/usr/local/groundwork/config/db.properties.foundation
/usr/local/groundwork/gwreports/PerformanceReports/gw-epr-host.rptdesign
/usr/local/groundwork /gwreports/PerformanceReports/gw-epr-host-multi-variable.rptdesign
/usr/local/groundwork
/gwreports/PerformanceReports/gw-epr-hostgroup-multi-variable.rptdesign
/usr/local/groundwork /gwreports/PerformanceReports/gw-epr-hostgroup.rptdesign
/usr/local/groundwork/tools/gwdiags.pl
/usr/local/groundwork/common/bin/gwdiags.pl
/usr/local/groundwork/nagvis/etc/nagvis.ini.php
```

b. Host and User

```
localhost, collage  
localhost, foundation
```

c. Sample Change in Files

```
collage.username=collage  
collage.password=gwrk  
collage.database=GWCollageDB  
collage.dbhost=localhost
```

or, for nagvis.php.ini:

```
; db1: GWCollageDB - foundation database  
dbhost1="localhost"  
dbport1=3306  
dbname1="GWCollageDB"  
dbuser1="foundation"  
dbpass1="gwrk"
```

2. Cacti

a. Files

```
/usr/local/groundwork/common/etc/check_cacti.conf  
/usr/local/groundwork/cacti/htdocs/include/config.php  
/usr/local/groundwork/cacti/extract_cacti.pl
```

b. Host and User

```
localhost, cactiuser
```

c. Sample Change in Files

```
$database_default = "cacti";  
$database_hostname = "localhost";  
$database_username = "cactiuser";  
$database_password = "cactiuser";
```

3. Dashboard

a. Files

```
/usr/local/groundwork/core/reports/etc/gwir.cfg  
/usr/local/groundwork/config/db.properties  
/usr/local/groundwork/config/db.properties.foundation
```

b. Host and User

```
%, root  
localhost, ir
```

c. Sample Change in Files

```
#Insight Reports Datasource
insightreports.dbhost=localhost
insightreports.database=dashboard
insightreports.username=ir
insightreports.password=gwrk
```

4. Logreports

a. Files

```
/usr/local/groundwork/config/db.properties
```

b. Host and User

```
%, root
localhost, logreporting
```

c. Sample Change in Files

```
#Log reporting database
logreporting.username=logreporting
logreporting.password=gwrk
logreporting.database=logreports
logreporting.dbhost=localhost
```

5. Monarch

a. Files

```
/usr/local/groundwork/config/db.properties
/usr/local/groundwork/config/db.properties.foundation
```

```
/usr/local/groundwork/nagvis/etc/nagvis.ini.php
```

b. Host and User

```
%, root
localhost, monarch
```

c. Sample Change in Files

```
#Monarch Datasource
monarch.dbhost=localhost
monarch.database=monarch
monarch.username=monarch
monarch.password=gwrk
```

or, for nagvis.php.ini:

```
; db2: monarch - configuration database
dbhost2="localhost"
dbport2=3306
dbname2="monarch"
dbuser2="monarch"
dbpass2="gwrk"
```

6. nedi

a. Files

```
/usr/local/groundwork/nedi/nedi.conf
```

b. Host and User

```
localhost, nedi
```

c. Sample change in file

```
# DB params

dbname nedi
dbuser nedi
dbpass nedi
```

7. jbossdb

a. Files

```
/usr/local/groundwork/foundation/container/webapps/jboss/default-ds.xml
```

b. Host and User

```
localhost, jboss
```

c. Sample change in file

```
<connection-url>jdbc:mysql://localhost:3306/jbossdb</connection-url>
<driver-class>com.mysql.jdbc.Driver</driver-class>
<user-name>jboss</user-name>
<password>jboss</password>
```

8. jbossportal

a. Files

```
/usr/local/groundwork/foundation/container/webapps/jboss/portal-ds.xml
```

b. Host and User

```
localhost, jboss
```

c. Sample change in file

```
<connection-url>jdbc:mysql://localhost:3306/jbossportal?useServerPrepStmts=false  
&jdbcCompliantTruncation=false</connection-url>  
<driver-class>com.mysql.jdbc.Driver</driver-class>  
<user-name>jboss</user-name>  
<password>jboss</password>
```