



eGenVar data-management system administrator's manual

(For versions *beta* 3.x, Last edited 14 October 2014)

Authors: Sabry Razick (sabryr@gmail.com) & Pål Sætrom

1. Basic

1.1. You would need the following to start

1.2. Installing

1.3. Selecting and resetting encryption key

1.4. Starting the server for the first time

1.5. Starting the server without configuring email

1.6. Stopping the server

1.7. Starting the server after installation.

1.8. Renewing certificates.

1.8. Startup options

2. Advanced

2.1. The system architecture

2.2. The configuration file.

2.3. Creating own certificates.

2.4. Configuring mail service.

2.5. Certificate related issues.

2.6. Firewall

2.7. Creating tags

2.7.1 Special considerations required when creating tags

3. Troubleshooting and recovery

3.1. password recovery.

3.1. Resetting encryption key.

4. Error messages:

4.1 Invalid Java version error

4.2 Fail to start using script.

4.3. Mailing error:

5.4. Mailing error:

5.5. Application server related errors

[4.6. INFO: Cannot find javadb client jar file, derby jdbc driver will not be available by default.](#)

[4.7. Out of memory errors](#)

[4.8. Email server authentication errors](#)

[4.9 Database server communication errors.](#)

[4.6. Command line tool \(egenv\) related error 1](#)

[4.7. Derby related technical](#)

[4.8 Mail certificate expired](#)

[6. Upgrade](#)

[7 User management](#)

[7.1. Activating user accounts.](#)

The eGenVar Data management system (EGDMS) is a file cataloging system for data from the life sciences domain. This manual is part of a three-part documentation. The other two are the command line tutorial and the web interface tutorial. This document explains how to install EGDMS and has two sections, “basic” and “advanced”. The basic section provides just enough information to start and run the system, without the need to for a deep understanding of the technical details. The advanced section provides details to customize the server and assumes the user is familiar with database management systems, application servers, secure communication (including how to generate certificates and to add certificates to trusted list), and firewall setup.

The EGDMS comes with an embedded Application server (Glassfish, <https://glassfish.java.net/>) and an embedded database management system (JavaDB: a version of Derby, <http://www.oracle.com/technetwork/java/javadb/overview/index.html>). Default configuration options are also defined for all operations. So, for the basic installation described below, you would require only the EGDMS installation package (<http://bigr.medisin.ntnu.no/data/eGenVar/>) as most computers already has other requirements like Java (<https://www.java.com/en/>).

1. Basic

1.1. You would need the following to start

- A. A Unix-based OS (RedHat Linux, Ubuntu, Mac OS etc..). (There is currently only an alpha version available for Microsoft windows.)
- B. Java Runtime Environment 7 (JRE) or later. Earlier Java versions **WILL NOT** work with the system. This is mainly due to the system relying on the improved file handling methods introduced in Java 1.7
 - a. Check Java version
 - i. Check 1: type **java -version** and you should see something similar to the following:

BOX_1:

```
>java -version
```

```
java version "1.7.0" Java(TM) SE Runtime Environment (build  
1.7.0-b147)Java HotSpot(TM) 64-Bit Server VM (build 21.0-b17, mixed  
mode) .
```

BOX_2:

If running **java -version** gives an error like the following then, you need to install Java and set the system path before installing EGDMS. Check the advanced section for solutions.

Command 'java' is available in '/usr/bin/java'

The command could not be located because '/usr/bin' is not included in the PATH environment variable.

java: command not found

Tip: To check the location of the Java installation :

- Using browser
 - <http://www.java.com/en/download/installed.jsp>
- On Unix:
 - **which java**
 - **echo \$JAVA_HOME**
- On Windows:
 - **where java**
 - **echo %java_home%**

- C. A google account (<https://accounts.google.com/SignUp?service=mail>). The easiest way to set up the system's mail server is to have a Gmail account. As all mail server configurations and the certificate required for sending system email through Gmail are provided with the EGDMS, you only need a user name and a password to start using the system. If you want to use some other mail server, follow the instructions given in the advanced section.
- D. The EGDMS server package. If you have not already done so, obtain the **egenvar.tar.gz** from <http://bigr.medisin.ntnu.no/data/eGenVar/>.
- E. The following ports of the machine should be available (that is, not used by other applications; to use other ports, use the configurations options explained in the advanced section):
 - i. 8185
 - ii. 8085
 - iii. 1557

1.2. Installing

- Place the **egenvar.tar.gz** archive in a folder where you have write permission.
 - Test write permission with **mkdir test**. This command should create a directory named test without causing any errors. If you get an error, change permissions or choose a different location.
 - Uncompress the package using the following command (or use your favourite decompression method)
 - **tar xvfz unix_egenvar.tar.gz**
 - This will install all packages and dependencies in the current folder. If the installation was a success, then you should see a directory named **eganv** and the following files and folders inside it.
 - data
 - lib
 - README.TXT
 - start.sh
 - config
 - Start_EgenVar1.jar
 - war

1.3. Selecting and resetting encryption key

The default encryption key is 01234567. If you want to use your own. Then include the new key in a file named “key.file” and place it in the directory where the **Start_EgenVar1.jar** is located. If you want to use the default then you could skip rest of this section (1.3) . You have the option to change it later.

(Following instructions are only if you want to change the default encryption key, this step is not required for the first time installation and you could change this key later)

This is the most important decision during installation, if you are not using a custom configuration file with all the passwords (to override password autogeneration). The encryption key is the only value that cannot be recovered. If you know it, you could recover all other configuration options, including passwords for the databases. If you lose this key you will not be able to recover anything and all information including the data will be permanently locked. There is no back-door built in, so the developers could not be any help in a situation like this.

To change the encryption key

- Create a text file named “key.file” and place it in the directory where the **Start_EgenVar1.jar** is.
- Include a key. The key should be 8 characters long and should be only numbers (E.g. 00034578 is a valid key, 34578 is not valid).
- Start the server.

1.4. Starting the server for the first time

When the server is started for the first time, the following steps will be done by the system through interaction with the user:

1. Required directory tree will be created.
2. The database required for user authentication will be created. If the embedded database management system (Derby) is used, then the database for the data management will also be created.
3. Required certificates configurations for the secure server will be generated with default values. These certificates can be replaced by users with their own certificates later if required. Certificates related to the communication with the gmail server and other secure communication channels will be added to the trusted store.
4. The email account will be configured, the required certificate generated, and one user account will be created.

The first time start (+installation) requires user interaction and cannot be invoked automatically (for example, as a startup script). When passwords are not provided by the user through the config.txt file (path specified as an argument during execution), passwords will be generated by the server. A example config file is located in the /egenvar/defaults/config.txt location. The rules to generate the default config options are included in the software (i.e. not

read from the example config file) and will be used if no config file is specified as an argument. A copy of the generated passwords will be emailed to the administrator. (For details on advanced password recovery refer to the “password recovery” in the advanced section.)

(For possible errors related to configuration and solutions for them please refer to the *Error messages* section at the end of this document.)

There are four ways to start the server for the first time.

1. Start without providing any configuration options. The server will generate default values for all options and will email the administrator the configurations used at the end.
 - `sh ./start.sh`
2. Use the default configuration file
 - `sh ./start.sh defaults/config.txt`
 - OR
 - `sh ./start.sh -c defaults/config.txt`
3. Use your own configuration file (see the advanced section)
 - `sh ./start.sh <config_file_full_path>`
 - OR
 - `sh ./start.sh -c <config_file_full_path>`
4. Force to use localhost as server address (may be required when testing using restricted WIFI or mobile broadband)
 - `sh ./start.sh -localhost`

*To reset using the localhost, use `-resetlocalhost`

During the installation, you will be asked to provide an email username and password for the Gmail account. After the email is verified, the database server and the application server will be configured. This configuration process may take a few minutes.
When the server is started you will see a message similar to what is in BOX_3.

BOX_3:

egenvar server started ports used S_PORT=8185|PORT=8085|DB_PORT=1557|DB_PORT_MEM=1567

Web interface:

https://egenvar.ntnu.no:8185/eGenVar_web/

WSDL for webservices:

http://egenvar.ntnu.no:8085/eGenVar_WebService/Authenticate_service?wsdl

To get the egenv tool:

wget <http://egenvar.ntnu.no:8085/egenv.tar.gz>

To get the R client tool:

wget <https://egenvar.ntnu.no:8185/eGenR.R>

To get the Python client :

wget <http://egenvar.ntnu.no:8085/eGenPy.py>

to stop the server from other terminals:

Start_EgenVar -stop 23895@dmed4929

STARTED

Optimising connections

Server is ready. (Time taken: 53 seconds)

(STOP -to stop server| M -more commands):

The EGDMS has server and client components. The server will be started at the end of the above procedure and needs to be up (keep running) for the client components to work. The clients are the command line tool *egenv* and the eGenVar_web web interface. The links to get these two clients are dynamically generated and will be displayed at the end of the starting process. In the above sample output, the tools are available from “<https://egenvar.ntnu.no:8185/>” and “<http://egenvar.ntnu.no:8085/>”. If you access these services from the machine where the server is installed, using *localhost* as the server name will also work; that is, the web interface will be accessible from http://localhost:8085/eGenVar_web/.

To access the server from other computers in the network, the firewall needs to be configured as explained in the advanced section.

1.5. Starting the server without configuring email

The EGDMS depends on a email server for user communication. The recommended option is to open a dedicated Google account to use for sending system emails to users. If for some reason you do not want to open a Google account, then you have to configure the system as described in the advanced section. You could also attempt to use a local host mail server if you have configured one (this is not fully tested yet).

You can start the server so that it does not use email communication by using the following command:

- `sh ./start.sh -nomail`
 - This command will disable the email facility and you have to take down the passwords and other assigned parameters manually as they are printed on screen. This is feature should be used for testing purposes only!

To use a local host mail service, start the server as described above, select “N” when asked “Do you want to use the mail account provided in the config file (by default this is an Gmail account)?” and then select “Y” for “Do you want to try using the system mail on localhost” (This has not been fully tested.)

1.6. Stopping the server

The application server, database server, and the web applications are loaded as a single process (which will have only one PID). There are no daemon process or dependencies started independently by the EGDMS. Therefore, when the server is terminated all system resources will be freed. The important thing is the order of shutting down the components. This is best achieved by issuing the **stop** command through the server management menu (as shown in **BOX_3** (STOP -to stop server| M -more commands)).

You will also notice the message “to stop the server from other terminals:

Start_EgenVar -stop 15839@dmed4929”. This method could be used if the terminal where the server was started from is not available. You could also use **kill -9 <PID>** to kill the process (PID displayed at the end of the starting process, in the **BOX_3** example the command would be **kill -9 15839**. Killing the process is not recommended and the STOP should be used whenever possible.

1.7. Starting the server after installation.

After the first time installation+start, the following command can be used to start the server. This could also be included as a startup event.

- `sh ./start.sh -fromlast`

The **-fromlast** instruction is used to start the server using the configuration options used in the last successful attempt.

1.8. Renewing certificates.

If the default certificates are used, they are set to expire in one year (from the date of creation). So they may have been expired by the time the software is being installed or they will expire at some point. The steps to follow to renew the certificates on a Unix based system is as follows.

1. Stop the server
2. Go to the config directory (egenvar/egenvar/config/)
3. Delete the following files
 - 3.1. cacerts.jks
 - 3.2. keystore.jks
 - 3.3. server.cer
4. Run the following script
 - 4.1. sh certify.sh
5. This will extend the certificate validity by an another year.

*if something goes wrong during this. Delete all the content of the config folder, start the server, stop it and follow the above instructions again (the server restart will recreate any missing files)

* It is highly recommended to use the certificate creation steps to create you own certificates rather than using the defaults. The defaults are for testing purposes only.

1.8. Startup options

The following table summarises the additional instructions that can be issued to the server during the startup.

Option	Description
-nomail	
-localhost	
-resethost	
-find_free_port	
-resetlocalhost	
-force	
-makePaths	
-upgrade	
-test	
-resetkey	
-version	
-stop	
-recover	
-fromlast	

2. Advanced

2.1. The system architecture

EGDMS uses two relational databases to store the data collected (EGDMS_DATAENTRY) and to manage user accounts (EGDMS_USERMANGE). If default settings are used, the embedded Java DB (Oracle supported version of Apache Derby) will be used as the database management system (DBMS). Therefore, EGDMS could work without having to install a dedicated DBMS or do any additional configurations.

The EGDMS also has an embedded version of the Sun Java Application server (Glassfish). There are two packages that use this application server. The first, Egen_web, is the web portal, which provides a graphical user interface to the system. The second, Egen_webservice, is a set of web-services providing a machine interface for the system. A precompiled version of a web-service client-based command line tool called egenv is included in the package. Therefore, the EGDMS has everything it needs in a single package and requires only a valid Java runtime environment (JRE) and an email account for installation and to run.

The EGDMS does not have a mail server. Neither is it designed to use the default mail client already configured in the system. It is by default designed to use an existing Google account for emailing purposes. Therefore, we recommend a dedicated Google account for your local installation. EGDMS will ask for a user name and a password of a Google account during the installation. It is possible to use other mailing services; please refer to the section about the configuration file for more details.

It is also advisable to check your firewall settings to control how the open ports are used and the access restrictions to the machine. This is not a requirement for the functioning of the system in the localhost mode, but is required for network mode.

2.2. The configuration file.

A configuration file is used to provide all administrative parameters to the system. For example the passwords for the two databases and email account information could be provided in this file. The system has default values for all configuration options except the email user name and the password. So the system can be installed without ever creating a configuration file or with configuration options only for some of the parameters. The configuration file should be a text file with UTF-8 text encoding. Table 1 provides a description of each parameter.

Table 1:

Parameter name	Details
DERBY_CONNECTION_PARAM	Provides additional parameters to the database connectivity URL. The default is: territory=en_US;collation=TERRITORY_BASED:PRIMARY The default value configures Derby to use en_US as the default character set and to ignore case sensitivity during searches.
DATAENTRY_ACCOUNT_PASSWORD	The password for the EGEN_DATAENTRY database. The default is 0000 .
mail.smtps.auth	Email account authentication option. The default is TRUE .
mail.smtps.host	The email server host name. The default is smtp.gmail.com
"--fromaddress"	The email address. no default value
SRC_ROOT	The application server root. Should point to a valid local location with write and execute permission. This must be a location on the same hard drive partition as the rest of the EGDMS package is installed. Network locations, temporary drive mounts or virtual links will result in system crash. Default is the current working directory . (Retrieved using method similar to pwd command in Unix)
DB_LOG	The name of the Derby log file. The default is db_log.txt
mail.smtp.port	The smtp port of the email server. The default is the port for Google: 465
SOURCE_DATA	The location of the additional data like the sql statements to create the databases and the ontology tables. All the data required to start the system are packaged with the installation packages. The default is the data directory in the current working directory
DB_DATAENTRY	The name of database used to store data. The default

	is EGEN_DATAENTRY, highly recommended not to change unless there is a really good reason.
DATAVUPDATE_ACCOUNT_PASSWORD	0
--mailuser	egenvar
CREATE_DB	SHA-1
SOURCE_WAR	/home/sabryr/tmp/server/egenvar/war/
SERVER__LIB	/home/sabryr/tmp/server/egenvar/egenvar/lib/
CMD_VERSION	0.943
USER_MANAGE_ACCOUNT_PASSWORD	0
DBHOST	derby://localhost
SERVER_DOCROOT	/home/sabryr/tmp/server/egenvar/egenvar/docroot/
mail.smtp.socketFactory.port	465
mail.smtp.password	
SOURCE_LIB	/home/sabryr/tmp/server/egenvar/lib/
SERVER_CONFIG	/home/sabryr/tmp/server/egenvar/egenvar/config/
DRIVERCLASSNAME	org.apache.derby.jdbc.ClientDriver
USER_MANAGE_ACCOUNT_NAME	usermanage
mail.smtp.socketFactory.fallback	FALSE
DATAVIEW_ACCOUNT_NAME	egendataview
S_PORT	8185
PORT	8085
DATAENTRY_ACCOUNT_NAME	egendataentry
SERVER_INSTALL_ROOT	/home/sabryr/tmp/server/egenvar/egenvar/

PRC_JAR	/home/sabryr/tmp/server/egenvar/lib/Procedures_eGenv_JAVADB.jar
DOC_ROOT	null
DB_USERS	EGEN_USERS
mail.smtp.socketFactory.class	javax.net.ssl.SSLSocketFactory
--mailhost	smtp.gmail.com
FILE_DB_CREATE_DERBY	data/create_tables_egen_dataEntry_simplified_derby.sql
DATASOURCECLASSNAME	org.apache.derby.jdbc.ClientDataSource
DATAUPDATE_ACCOUNT_NAME	egendataupdate
DATAVIEW_ACCOUNT_PASSWORD	0
DB_PORT	1557
mail.transport.protocol	smtps
mail.smtp.auth	TRUE

2.3. Creating own certificates.

The EGDMS comes with a self signed certificate that does not contain any organization or user specific details. If you want to include these details or if you have a certificate signed by an authority, then you could include this as described in the following steps:

1. Stop the server
2. Delete everything in the config directory (you could just delete the certificate related files, but deleting all is safer and required files will be generated when you restart the server)
3. Place or create (see BOX_2) the certificates inside the config directory.

BOX_2 : How to create certificates manually

```
keytool -genkey -alias s1as -keyalg RSA -keypass changeit -validity 365  
-storepass changeit -keystore keystore.jks -dname ;
```

```
keytool -export -alias s1as -file server.cer -keystore keystore.jks -storepass  
changeit ;
```

```
keytool -import -v -trustcacerts -alias s1as -file server.cer -keystore  
cacerts.jks -keypass changeit
```

```
keytool -genkey -alias glassfish-instance -keyalg RSA -keypass changeit  
-validity 365 -storepass changeit -keystore keystore.jks -dname
```

```
keytool -export -alias glassfish-instance -storepass changeit -file server.cer  
-keystore keystore.jks
```

```
keytool -import -v -trustcacerts -alias glassfish-instance -file server.cer  
-keystore cacerts.jks -keypass changeit;
```

*Remember to add the mail (Gmail) certificates (see the certificate related issues section) whenever you do the above steps manually.

If something goes wrong, stop the server, delete the content of the config directory and restart the server to get the default certificates. You could also do this by issuing the following command.

- `sh ./start.sh -fromlast -recreate_server_certificates`

Then to recreate the Gmail certificate.

- `sh ./start.sh -fromlast -recreate_mail_certificates`

2.4. Configuring mail service.

The EGDMS uses Gmail to send mails and activation codes. Therefore, you need an account with Google and its username and password to start the installation. Other email services could be used if you provide the following information in the config.txt file.

- A. `--fromaddress=<The full email address>`
- B. `--mailuser=<Username for the email account>`
- C. `--mailhost=<The host name>` E.g. `smtp.gmail.com`
- D. `mail.smtp.password=<Password for the email account>`
- E. `mail.smtp.auth=<Boolean, yes/no>` whether to use SMTP authentication
`mail.smtp.socketFactory.fallback=`
- F. `mail.smtp.port=<Integer, The SMTP port>` E.g. 465
- G. `mail.smtp.socketFactory.port=<Integer, Specifies the port to connect to when using the specified socket factory>`

(more details on this parameters

<https://javamail.java.net/nonav/docs/api/com/sun/mail/smtp/package-summary.html>)

2.5. Certificate related issues.

***This section describes how to obtain and trust certificates related to Gmail. If you are using some other mail service with secure communication requirements then you could use the description as an example to configure your mail server.**

Certificates are used when connecting to the mail server using secure ports. By default the EGDMS obtains a certificate for Gmail and adds to the trusted list. When using other mail servers the end user needs to obtain a certificate in addition to changing the configuration options above. The certificates are located in the config directory. (e.g. `.../egenvar/egenvar/config/`).

Although we have included self diagnostics for Gmail and ways to automatically rectify most of the common problems, manual intervention may be required at some instances. In addition, if Google changes the way clients connect, the default methods may just fail.

There are three methods to refresh the Gmail certificates.

1. Delete the content in the config directory (the directory with the .jks files `...egenvar/egenvar/config/`) and restart the server

OR

2. Issue the following command, when the server is not running

- `sh ./start.sh -recreate_mail_certificates`

OR

3. Add the certificates manually (BOX_3)

Box_3 adding GMail certificates manually

1. Go to the directory where the .cert file is (the config directory of the EGDMS server)
2. `openssl s_client -CApath . -connect smtp.gmail.com:465`
3. Copy and save the lines between (including the beginning and end)
i.e. between "-----BEGIN CERTIFICATE-----" and "-----END CERTIFICATE-----" into a file, say, gmail.cert:
4. Delete existing entry:
`keytool -delete -alias smtp.gmail.com -keystore cacerts.jks -storepass changeit`
5. Add the new key.
`keytool -import -alias smtp.gmail.com -keystore cacerts.jks -file gmail.txt`

**If the exact procedure above is not followed then you will encounter errors later on. For example, "keytool error: java.lang.Exception: Input not an X.509 certificate" will be thrown.*

2.6. Firewall

Example firewall settings for Ubuntu using iptables rules are given below (More details on iptables, please refer <https://help.ubuntu.com/community/IptablesHowTo>). These settings will open ports 8185 and 8085 for computers with the IP range 78.91.0.0 to 78.91.254.254.

```
iptables -F
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT -i eth0 -s 78.91.0.0/16 -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -i eth0 -s 78.91.0.0/16 -p tcp --dport 60000:65535 -j ACCEPT
iptables -A INPUT -i eth0 -s 78.91.0.0/16 -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -i eth0 -s 78.91.0.0/16 -p tcp --dport 8085 -j ACCEPT
iptables -A INPUT -i eth0 -s 78.91.0.0/16 -p tcp --dport 8090 -j ACCEPT
iptables -A INPUT -i eth0 -s 78.91.0.0/16 -p tcp --dport 8185 -j ACCEPT
iptables -A INPUT -i eth0 -s 78.91.0.0/16 -p tcp --dport 21 -j ACCEPT
iptables -A INPUT -i eth0 -s 78.91.0.0/16 -p tcp --dport 2121 -j ACCEPT
iptables -A INPUT -i eth0 -s 78.91.0.0/16 -p tcp --dport 2123 -j ACCEPT
iptables -A INPUT -i eth0 -s 78.91.0.0/16 -p tcp --dport 2125 -j ACCEPT
iptables -A INPUT -i eth0 -s 78.91.0.0/16 -p tcp --dport 2127 -j ACCEPT
iptables -A INPUT -i eth0 -s 78.91.0.0/16 -p tcp --dport 2141 -j ACCEPT
```

```
iptables -A INPUT -i eth0 -s 78.91.0.0/16 -p tcp --dport 990 -j ACCEPT
iptables -A INPUT -i eth0 -s 78.91.0.0/16 -p tcp --dport 60000:65535 -j ACCEPT
iptables -I INPUT -p tcp -i eth0 --dport 22 -m state --state NEW -m recent --name sshprobe --set -j ACCEPT
iptables -I INPUT -p tcp -i eth0 --dport 22 -m state --state NEW -m recent --name sshprobe --update --seconds 60
--hitcount 3 --rttl -j DROP
iptables -A INPUT -j DROP
iptables -t nat -A OUTPUT -o lo -p tcp --dport 80 -j REDIRECT --to-port 8185
iptables -t nat -A PREROUTING -p tcp -m tcp --dport 80 -j REDIRECT --to-ports 8185
iptables -I INPUT 5 -m limit --limit 5/min -j LOG --log-prefix "iptables denied: " --log-level 7
iptables -L
```

2.7. Creating tags

Tags are used to describe the content of the files and to provide provenance information. Tag source files are used to provide pre-created sets of tags, which are extendable by the user. The **BOX_4** provide a sample tag source file.

BOX_4: Tag source for biological ontologies

```
#id|parent_id|name|obo_id
1||0|Biological_process.obo||1
2||1|molecular_function|GO:0003674
3||2|protein binding trans....|GO:0000988
4||3|transcription factor ....|GO:0000989
5||4|RNA polymerase III .....|GO:0001007
6||5|TFIIIB-type transcr.....|GO:0001026
7||6|RNA polymerase III .....|GO:0001027
....
....
...
..
```

The following are the requirements for tag source files:

- The file should be UTF-8 plain text. If you make this file in a spreadsheet application, remember to export as plain text.
- The first row starts with a **#** and contains the column headings separated by two pipes (i.e. ||).
- The three headings id, parent:id and name are compulsory and other columns are optional.
- The id should be unique (no two rows should have the same id).
- A name could be repeated given that the parent_id is different.
- The parent_id is the id of the parent or 0 if this is the top-most component.
- The filename should end with **_tagsource.txt**. (i.e. **openBiologicalontologies_tagsource.txt**)
- The file created should be placed in the data directory (found in the same level as the **start.sh**)

2.7.1 Special considerations required when creating tags

1. If you are using the default RDBMS (JavaDb/Derby) remember that dashes and spaces (and non-alphanumeric characters) should be avoided if possible. This is because when deciding constraints for example both U-266 and U266 would be considered equal and will throw an error.ws

3. Troubleshooting and recovery

3.1. password recovery.

The first time start (+installation) requires user interactivity and cannot be invoked automatically (i.e. as a startup script). When passwords are not provided by the user through the config.txt file, passwords will be generated by the server. These same passwords must be used for the subsequent starting of the server, otherwise the encrypted database will not be read. If you lose the administrator passwords they can not be recovered remotely and you would need to setup a new instance (delete the database directories).

To recover passwords and other configuration parameters, shutdown the server and type the command below.

- `sh ./start.sh -recover`

This will prompt you to type in a encryption key. When the correct encryption key is given (default is 12345678) the details will be printed on the screen.

3.1. Resetting encryption key.

To change the key from the default use the steps below.

1. Stop the server if it running
2. Create a file with the new key
3. Name the file "key.file"
4. Place it in the directory where the **Start_EgenVar1.jar** is located
5. `sh ./start.sh -resetkey`
6. When asked for the Enter current encryption key: use 12345678
7. If setting the was successful you would see the message encryption key reset was successful. New key is in /.../egenvar/key.file

To change the key from the existing key.file value.

1. Stop the server if it running
2. Write down the value in the current key.file
3. Create a file with the new key and name it "key.file"
4. Place it in the directory where the **Start_EgenVar1.jar** is located
5. `sh ./start.sh -resetkey`
6. When asked for the Enter current encryption key: use the value in the old file (the value you wrote down in step 2)
7. If setting the was successful you would see the message encryption key reset was successful. New key is in /.../egenvar/key.file

4. Error messages:

4.1 Invalid Java version error

```
java version "1.6.0_27"
OpenJDK Runtime Environment (IcedTea6 1.12.6) (6b27-1.12.6-1ubuntu0.10.04.4)
OpenJDK 64-Bit Server VM (build 20.0-b12, mixed mode)
rm: cannot remove `*trace': No such file or directory
Exception in thread "main" java.lang.UnsupportedClassVersionError:
no/ntnu/medisin/egenvar/server/Start_EgenVar1 : Unsupported major.minor version 51.0
    at java.lang.ClassLoader.defineClass1(Native Method)
    at java.lang.ClassLoader.defineClass(ClassLoader.java:643)
    at java.security.SecureClassLoader.defineClass(SecureClassLoader.java:142)
    at java.net.URLClassLoader.defineClass(URLClassLoader.java:277)
    at java.net.URLClassLoader.access$000(URLClassLoader.java:73)
    at java.net.URLClassLoader$1.run(URLClassLoader.java:212)
    at java.security.AccessController.doPrivileged(Native Method)
    at java.net.URLClassLoader.findClass(URLClassLoader.java:205)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:323)
    at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:294)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:268)
Could not find the main class: no.ntnu.medisin.egenvar.server.Start_EgenVar1. Program will
exit.
```

The above error was simulated by trying to run the startup script while pointing to a version of java older than 1.7. As emphasized in the beginning of this document, the software WILL NOT work if you have a older version of JAVA. The most robust solution is to update your JAVA. If there is any issue preventing the upgrade, then you could download the latest JRE and manually point to it (override the system default). ‘

(e.g.

1. echo \$PATH, then copy and save the path,
 2. export PATH=/bin:/home/aaa/java/jdk1.7.0_05/bin/:/usr/bin/ (to temporarily change the path)
 3. sh ./start.sh
-)

4.2 Fail to start using script.

A simple shell script “start.sh” is provided to help starting the server. However, for some reason it may fail with the following error:

```
Exception in thread "main" java.lang.NoClassDefFoundError: org/glassfish/embeddable/GlassFishException
    at java.lang.Class.getDeclaredMethods0(Native Method)
    at java.lang.Class.privateGetDeclaredMethods(Class.java:2442)
    at java.lang.Class.getMethod0(Class.java:2685)
    at java.lang.Class.getMethod(Class.java:1620)
    at sun.launcher.LauncherHelper.getMainMethod(LauncherHelper.java:484)
    at sun.launcher.LauncherHelper.checkAndLoadMain(LauncherHelper.java:476)
Caused by: java.lang.ClassNotFoundException: org.glassfish.embeddable.GlassFishException
    at java.net.URLClassLoader$1.run(URLClassLoader.java:366)
    at java.net.URLClassLoader$1.run(URLClassLoader.java:355)
    at java.security.AccessController.doPrivileged(Native Method)
    at java.net.URLClassLoader.findClass(URLClassLoader.java:354)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:423)
    at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:308)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:356)
    ... 6 more
```

If this happens, you could try to manually invoke the JAR file provided, as follows

- **java -Xms2g -Xmx4g -jar Start_EgenVar1.jar**

This will start the server with 2Gb memory and expand up to 4Gb if needed.

4.3. Mailing error:

```
Could not connect to SMTP host: smtp.gmail.com, port: 465;
nested exception is:
    javax.net.ssl.SSLHandshakeException: sun.security.validator.ValidatorException: PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested
target
    at com.sun.mail.smtp.SMTPTransport.openServer(SMTPTransport.java:1934)
    at com.sun.mail.smtp.SMTPTransport.protocolConnect(SMTPTransport.java:638)
    at javax.mail.Service.connect(Service.java:317)
    at javax.mail.Service.connect(Service.java:176)
    at javax.mail.Service.connect(Service.java:125)
    at javax.mail.Transport.send0(Transport.java:194)
```

OR

```
javax.net.ssl.SSLHandshakeException: sun.security.validator.ValidatorException: PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested
target
    at sun.security.ssl.Alerts.getSSLException(Unknown Source)
    at sun.security.ssl.SSLSocketImpl.fatal(Unknown Source)
```

```

at sun.security.ssl.Handshaker.fatalSE(Unknown Source)
at sun.security.ssl.Handshaker.fatalSE(Unknown Source)
at sun.security.ssl.ClientHandshaker.serverCertificate(Unknown Source)
at sun.security.ssl.ClientHandshaker.processMessage(Unknown Source)
at sun.security.ssl.Handshaker.processLoop(Unknown Source)
at sun.security.ssl.Handshaker.process_record(Unknown Source)
at sun.security.ssl.SSLSocketImpl.readRecord(Unknown Source)
at sun.security.ssl.SSLSocketImpl.performInitialHandshake(Unknown Source)
at sun.security.ssl.SSLSocketImpl.startHandshake(Unknown Source)
at sun.security.ssl.SSLSocketImpl.startHandshake(Unknown Source)
at no.ntnu.medisin.egenvar.server.Start_EgenVar1.getGmailCertificates(Start_EgenVar1.java:4210)
at no.ntnu.medisin.egenvar.server.Start_EgenVar1.createCertificate(Start_EgenVar1.java:4168)
at no.ntnu.medisin.egenvar.server.Start_EgenVar1.init(Start_EgenVar1.java:852)
at no.ntnu.medisin.egenvar.server.Start_EgenVar1.main(Start_EgenVar1.java:317)
Caused by: sun.security.validator.ValidatorException: PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested
target

```

These error messages occur for two reasons; one is that the mail server is facing temporary problem or the other is that there is a problem with the certificates.

First stop the server if it start despite the error. Wait 5 Minutes and try again. If you keep on getting the error, then take the steps described in obtaining new certificates section. to solve problems related to certificates.

5.4. Mailing error:

Error : 454 4.7.0 Cannot authenticate due to temporary system problem. Try again later. qq7sm19855682bkb.6 - gsmtip

This is a very rare but a possible error. This occurs if google faces some system issue preventing the authentication process. Do not make any changes, just wait a minute or two and retry.

5.5. Application server related errors

```

SEVERE: Exception while visiting com/sun/gjc/common/DataSourceSpec.class of size 3267
java.lang.NullPointerException
    at org.glassfish.hk2.classmodel.reflect.impl.TypesImpl.getType(TypesImpl.java:78)
    at org.glassfish.hk2.classmodel.reflect.impl.ModelClassVisitor.visit(ModelClassVisitor.java:119)
    at org.objectweb.asm.ClassReader.accept(Unknown Source)
    at org.objectweb.asm.ClassReader.accept(Unknown Source)
    at org.glassfish.hk2.classmodel.reflect.Parser$5.on(Parser.java:363)
    at org.glassfish.hk2.classmodel.reflect.util.JarArchive.onSelectedEntries(JarArchive.java:125)
    at org.glassfish.hk2.classmodel.reflect.Parser.doJob(Parser.java:348)
    at org.glassfish.hk2.classmodel.reflect.Parser.access$300(Parser.java:70)
    at org.glassfish.hk2.classmodel.reflect.Parser$3.call(Parser.java:307)
    at org.glassfish.hk2.classmodel.reflect.Parser$3.call(Parser.java:296)
    at java.util.concurrent.FutureTask$Sync.innerRun(FutureTask.java:334)
    at java.util.concurrent.FutureTask.run(FutureTask.java:166)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1110)

```

```
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:603)
at java.lang.Thread.run(Thread.java:722)
```

This error is application server related and will not interfere with the EGDMS functionality. This bug is known and has been reported.

4.6. INFO: Cannot find javadb client jar file, derby jdbc driver will not be available by default.

This error, if it occurs, will be handled by the system and no action is required by the user.

4.7. Out of memory errors

```
java.lang.OutOfMemoryError: PermGen space
  at java.lang.ClassLoader.defineClass1(Native Method)
  at java.lang.ClassLoader.defineClass(Unknown Source)
  at java.security.SecureClassLoader.defineClass(Unknown Source)
  at org.glassfish.web.loader.WebappClassLoader.findClass(WebappClassLoader.java:927)
  at org.glassfish.web.loader.WebappClassLoader.loadClass(WebappClassLoader.java:1476)
  at org.glassfish.web.loader.WebappClassLoader.loadClass(WebappClassLoader.java:1359)
  at com.sun.enterprise.deployment.annotation.impl.ModuleScanner.getElements(ModuleScanner.java:246)
  at com.sun.enterprise.deployment.archivist.Archivist.processAnnotations(Archivist.java:557)
  at org.glassfish.web.embed.impl.EmbeddedWebArchivist.processAnnotations(EmbeddedWebArchivist.java:133)
  at com.sun.enterprise.deployment.archivist.Archivist.readAnnotations(Archivist.java:442)
  at com.sun.enterprise.deployment.archivist.Archivist.readAnnotations(Archivist.java:429)
  at com.sun.enterprise.deployment.archivist.Archivist.readRestDeploymentDescriptors(Archivist.java:405)
  at com.sun.enterprise.deployment.archivist.Archivist.readDeploymentDescriptors(Archivist.java:380)
  at com.sun.enterprise.deployment.archivist.Archivist.open(Archivist.java:243)
  at com.sun.enterprise.deployment.archivist.Archivist.open(Archivist.java:252)
  at com.sun.enterprise.deployment.archivist.Archivist.open(Archivist.java:213)
  at com.sun.enterprise.deployment.archivist.ApplicationFactory.openArchive(ApplicationFactory.java:165)
  at org.glassfish.javaee.core.deployment.DolProvider.load(DolProvider.java:185)
  at org.glassfish.javaee.core.deployment.DolProvider.load(DolProvider.java:94)
  at com.sun.enterprise.v3.server.ApplicationLifecycle.loadDeployer(ApplicationLifecycle.java:827)
  at com.sun.enterprise.v3.server.ApplicationLifecycle.setupContainerInfos(ApplicationLifecycle.java:769)
  at com.sun.enterprise.v3.server.ApplicationLifecycle.deploy(ApplicationLifecycle.java:368)
  at com.sun.enterprise.v3.server.ApplicationLifecycle.deploy(ApplicationLifecycle.java:240)
  at org.glassfish.deployment.admin.DeployCommand.execute(DeployCommand.java:389)
  at com.sun.enterprise.v3.admin.CommandRunnerImpl$1.execute(CommandRunnerImpl.java:348)
  at com.sun.enterprise.v3.admin.CommandRunnerImpl.doCommand(CommandRunnerImpl.java:363)
  at com.sun.enterprise.v3.admin.CommandRunnerImpl.doCommand(CommandRunnerImpl.java:1085)
  at com.sun.enterprise.v3.admin.CommandRunnerImpl.access$1200(CommandRunnerImpl.java:95)
  at com.sun.enterprise.v3.admin.CommandRunnerImpl$ExecutionContext.execute(CommandRunnerImpl.java:1291)
  at com.sun.enterprise.v3.admin.CommandRunnerImpl$ExecutionContext.execute(CommandRunnerImpl.java:1259)
  at com.sun.enterprise.admin.cli.embeddable.DeployerImpl.deploy(DeployerImpl.java:129)
  at no.ntnu.medisin.egenvar.server.Start_EgenVar1.diploy_services(Start_EgenVar1.java:1323)
```

or

Nov 06, 2013 4:39:50 PM org.glassfish.webservices.WSServletContextListener contextInitialized

WARNING: Deployment failed

java.lang.OutOfMemoryError: PermGen space

```
    at sun.misc.Unsafe.defineClass(Native Method)
    at sun.reflect.ClassDefiner.defineClass(Unknown Source)
    at sun.reflect.MethodAccessorGenerator$1.run(Unknown Source)
    at sun.reflect.MethodAccessorGenerator$1.run(Unknown Source)
    at java.security.AccessController.doPrivileged(Native Method)
    at sun.reflect.MethodAccessorGenerator.generate(Unknown Source)
    at sun.reflect.MethodAccessorGenerator.generateConstructor(Unknown Source)
    at sun.reflect.NativeConstructorAccessorImpl.newInstance(Unknown Source)
    at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(Unknown Source)
    at java.lang.reflect.Constructor.newInstance(Unknown Source)
    at java.lang.reflect.Proxy.newProxyInstance(Unknown Source)
    at com.sun.xml.txw2.ContainerElement._cast(ContainerElement.java:368)
    at com.sun.xml.txw2.ContainerElement._element(ContainerElement.java:389)
    at com.sun.xml.txw2.ContainerElement.addElement(ContainerElement.java:233)
    at com.sun.xml.txw2.ContainerElement.invoke(ContainerElement.java:160)
    at $Proxy171.element(Unknown Source)
    at
com.sun.xml.bind.v2.schemagen.XmlSchemaGenerator$Namespace$ElementWithType.writeTo(XmlSchemaGenerator.java:1439)
    at com.sun.xml.bind.v2.schemagen.XmlSchemaGenerator$Namespace.writeTo(XmlSchemaGenerator.java:690)
    at com.sun.xml.bind.v2.schemagen.XmlSchemaGenerator$Namespace.access$800(XmlSchemaGenerator.java:505)
    at com.sun.xml.bind.v2.schemagen.XmlSchemaGenerator.write(XmlSchemaGenerator.java:487)
    at com.sun.xml.bind.v2.runtime.JAXBContextImpl.generateSchema(JAXBContextImpl.java:828)
    at com.sun.xml.ws.db.glassfish.JAXBRIContextWrapper.generateSchema(JAXBRIContextWrapper.java:103)
    at com.sun.xml.ws.wsdl.writer.WSDLGenerator.generateTypes(WSDLGenerator.java:434)
    at com.sun.xml.ws.wsdl.writer.WSDLGenerator.generateDocument(WSDLGenerator.java:410)
    at com.sun.xml.ws.wsdl.writer.WSDLGenerator.doGeneration(WSDLGenerator.java:307)
    at com.sun.xml.ws.db.DatabindingImpl.generateWSDL(DatabindingImpl.java:224)
    at com.sun.xml.ws.server.EndpointFactory.generateWSDL(EndpointFactory.java:550)
    at com.sun.xml.ws.server.EndpointFactory.create(EndpointFactory.java:276)
    at com.sun.xml.ws.server.EndpointFactory.createEndpoint(EndpointFactory.java:145)
    at com.sun.xml.ws.api.server.WSEndpoint.create(WSEndpoint.java:569)
    at com.sun.xml.ws.api.server.WSEndpoint.create(WSEndpoint.java:552)
    at com.sun.xml.ws.api.server.WSEndpoint.create(WSEndpoint.java:623)
```

or

Nov 06, 2013 4:39:50 PM org.apache.catalina.core.StandardContext start

SEVERE: PWC1306: Startup of context /eGenVar_WebService failed due to previous errors

Nov 06, 2013 4:39:50 PM org.apache.catalina.core.StandardContext start

SEVERE: PWC1305: Exception during cleanup after start failed

org.apache.catalina.LifecycleException: PWC2769: Manager has not yet been started

```
    at org.apache.catalina.session.StandardManager.stop(StandardManager.java:873)
    at org.apache.catalina.core.StandardContext.stop(StandardContext.java:5571)
    at com.sun.enterprise.web.WebModule.stop(WebModule.java:527)
    at org.apache.catalina.core.StandardContext.start(StandardContext.java:5384)
    at com.sun.enterprise.web.WebModule.start(WebModule.java:498)
    at org.apache.catalina.core.ContainerBase.addChildInternal(ContainerBase.java:917)
    at org.apache.catalina.core.ContainerBase.addChild(ContainerBase.java:901)
    at org.apache.catalina.core.StandardHost.addChild(StandardHost.java:733)
    at com.sun.enterprise.web.WebContainer.loadWebModule(WebContainer.java:2019)
    at com.sun.enterprise.web.WebContainer.loadWebModule(WebContainer.java:1669)
```

```

at com.sun.enterprise.web.WebApplication.start(WebApplication.java:109)
at org.glassfish.internal.data.EngineRef.start(EngineRef.java:130)
at org.glassfish.internal.data.ModuleInfo.start(ModuleInfo.java:269)
at org.glassfish.internal.data.ApplicationInfo.start(ApplicationInfo.java:301)
at com.sun.enterprise.v3.server.ApplicationLifecycle.deploy(ApplicationLifecycle.java:461)
at com.sun.enterprise.v3.server.ApplicationLifecycle.deploy(ApplicationLifecycle.java:240)
at org.glassfish.deployment.admin.DeployCommand.execute(DeployCommand.java:389)
at com.sun.enterprise.v3.admin.CommandRunnerImpl$1.execute(CommandRunnerImpl.java:348)
at com.sun.enterprise.v3.admin.CommandRunnerImpl.doCommand(CommandRunnerImpl.java:363)
at com.sun.enterprise.v3.admin.CommandRunnerImpl.doCommand(CommandRunnerImpl.java:1085)
at com.sun.enterprise.v3.admin.CommandRunnerImpl.access$1200(CommandRunnerImpl.java:95)
at com.sun.enterprise.v3.admin.CommandRunnerImpl$ExecutionContext.execute(CommandRunnerImpl.java:1291)
at com.sun.enterprise.v3.admin.CommandRunnerImpl$ExecutionContext.execute(CommandRunnerImpl.java:1259)
at com.sun.enterprise.admin.cli.embeddable.DeployerImpl.deploy(DeployerImpl.java:129)
at no.ntnu.medisin.egenvar.server.Start_EgenVar1.diploy_services(Start_EgenVar1.java:1323)
at no.ntnu.medisin.egenvar.server.Start_EgenVar1.status(Start_EgenVar1.java:490)
at no.ntnu.medisin.egenvar.server.Start_EgenVar1.commit(Start_EgenVar1.java:1227)
at no.ntnu.medisin.egenvar.server.Start_EgenVar1.main(Start_EgenVar1.java:270)

```

WARNING: GRIZZLY0023: Interrupting idle Thread: http-thread-pool-8185(2).
Nov 07, 2013 3:58:42 PM com.sun.grizzly.http.KeepAliveThreadAttachment timedOut
WARNING: GRIZZLY0023: Interrupting idle Thread: http-thread-pool-8185(2).

or

Deploying /home/sabryr/tmp/server/egenvar/war/eGenVar_WebService.war

^C

Java HotSpot(TM) 64-Bit Server VM warning: Exception java.lang.OutOfMemoryError occurred dispatching signal SIGINT to handler- the VM may need to be forcibly terminated

A restart of the server is required for these errors. The reason for these errors are likely too many redeployment attempts. Specially, if you have introduced any third party web services to the EGDMS.

4.8. Email server authentication errors

javax.mail.AuthenticationFailedException: 535-5.7.8 Username and Password not accepted. Learn more at 535 5.7.8 <http://support.google.com/mail/bin/answer.py?answer=14257> a4sm14232477bko.11 - gsmtpt

```

at com.sun.mail.smtp.SMTPTransport$Authenticator.authenticate(SMTPTransport.java:809)
at com.sun.mail.smtp.SMTPTransport.authenticate(SMTPTransport.java:752)
at com.sun.mail.smtp.SMTPTransport.protocolConnect(SMTPTransport.java:669)
at javax.mail.Service.connect(Service.java:295)
at no.ntnu.medisin.egenvar.server.Start_EgenVar1.sendEmail(Start_EgenVar1.java:3976)
at no.ntnu.medisin.egenvar.server.Start_EgenVar1.test_mail(Start_EgenVar1.java:1341)
at no.ntnu.medisin.egenvar.server.Start_EgenVar1.commit(Start_EgenVar1.java:1100)
at no.ntnu.medisin.egenvar.server.Start_EgenVar1.main(Start_EgenVar1.java:316)

```

Sending mail failed. Host=smtp.gmail.com|port=465|User=egenavar

Error 1e: mail set up failed. Check the user name and password is correct.

The above error may occur during installation and is due to providing the wrong login

information for the email account. Stop the server, delete the database directories (EGEN_DATAENTRY and EGEN_USERS) before reattempting to install the server.

4.9 Database server communication errors.

```
java.lang.Exception: DRDA_NoIO.S:Could not connect to Derby Network Server on host 127.0.0.1, port 1557:
Connection refused
    at org.apache.derby.impl.drda.NetworkServerControlImpl.consolePropertyMessageWork(Unknown Source)
    at org.apache.derby.impl.drda.NetworkServerControlImpl.consolePropertyMessage(Unknown Source)
    at org.apache.derby.impl.drda.NetworkServerControlImpl.setUpSocket(Unknown Source)
    at org.apache.derby.impl.drda.NetworkServerControlImpl.getCurrentProperties(Unknown Source)
    at org.apache.derby.drda.NetworkServerControl.getCurrentProperties(Unknown Source)
    at no.ntnu.medisin.egenvar.server.Start_EgenVar1.startDerby(Start_EgenVar1.java:1400)
    at no.ntnu.medisin.egenvar.server.Start_EgenVar1.commit(Start_EgenVar1.java:1101)
    at no.ntnu.medisin.egenvar.server.Start_EgenVar1.main(Start_EgenVar1.java:316)
Warning setting trace failed
Database server started @2013-11-1
```

The main reason for this error is trying to connect to the database using wrong user names and/or passwords. After the first time installation + start, the same configuration file (with the passwords used to create the databases) or **-fromlast** option should be used. If the passwords are not available and the password recovery failed, then the only option is to use a different installation location or to delete the two database folders. One example where the above situation can occur is if you had started the server for the first time without specifying a configuration file, then passwords will be generated and a copy of these will be emailed to the administrator. Subsequently, if the server is attempted to be started again without the **-fromlast** option then this error will occur.

4.6. Command line tool (egenv) related error 1

```
Error: webservices not available or the address of the registered server has changed, use egenv.sh -diagnose
```

This error can be due to the user providing the wrong WSDL address or a change in the IP of the server. If the server IP is dynamic or has a recycling policy, then the command line users may experience this error when the change occurs. Most likely, the users would say that “it was working fine until very recently and suddenly I get an error”. The solution is to authenticate using the diagnostic option.

- **egenv -diagnose**

This will ask user for the WSDL location, username and password and replace the old certificate, if the authentication was a success.

4.7. Derby related technical

java.sql.SQLException: A network protocol error was encountered and the connection has been terminated: The requested command encountered an unarchitected and implementation-specific condition for which there was no architected message (additional information may be available in the derby.log file on the server).

```
    at org.apache.derby.client.am.SQLExceptionFactory40.getSQLException(Unknown Source)
    at org.apache.derby.client.am.SqlException.getSQLException(Unknown Source)
    at org.apache.derby.client.am.Connection.createStatement(Unknown Source)
    at com.sun.gjc.spi.base.ConnectionHolder.createStatement(ConnectionHolder.java:256)
    at com.sun.gjc.spi.jdbc40.ConnectionWrapper40.createStatement(ConnectionWrapper40.java:75)
    at webservice.medisin.ntnu.no.Authenticate_service.getWebServerName(Authenticate_service.java:4278)
    at webservice.medisin.ntnu.no.Authenticate_service.advancedQueryHandler(Authenticate_service.java:6590)
    at webservice.medisin.ntnu.no.Authenticate_service.addRelationships(Authenticate_service.java:5577)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:601)
    at org.glassfish.webservices.InstanceResolverImpl$1.invoke(InstanceResolverImpl.java:143)
    at com.sun.xml.ws.server.InvokerTube$2.invoke(InvokerTube.java:149)
    at com.sun.xml.ws.server.sei.SEIInvokerTube.processRequest(SEIInvokerTube.java:94)
    at com.sun.xml.ws.api.pipe.Fiber.__doRun(Fiber.java:961)
    at com.sun.xml.ws.api.pipe.Fiber._doRun(Fiber.java:910)
    at com.sun.xml.ws.api.pipe.Fiber.doRun(Fiber.java:873)
    at com.sun.xml.ws.api.pipe.Fiber.runSync(Fiber.java:775)
    at com.sun.xml.ws.api.pipe.helper.AbstractTubeImpl.process(AbstractTubeImpl.java:116)
    at org.glassfish.webservices.MonitoringPipe.process(MonitoringPipe.java:142)
    at com.sun.xml.ws.api.pipe.helper.PipeAdapter.processRequest(PipeAdapter.java:119)
    at com.sun.xml.ws.api.pipe.Fiber.__doRun(Fiber.java:961)
    at com.sun.xml.ws.api.pipe.Fiber._doRun(Fiber.java:910)
    at com.sun.xml.ws.api.pipe.Fiber.doRun(Fiber.java:873)
    at com.sun.xml.ws.api.pipe.Fiber.runSync(Fiber.java:775)
    at com.sun.xml.ws.api.pipe.helper.AbstractTubeImpl.process(AbstractTubeImpl.java:116)
    at
com.sun.enterprise.security.webservices.CommonServerSecurityPipe.processRequest(CommonServerSecurityPipe.java:212
)
    at
com.sun.enterprise.security.webservices.CommonServerSecurityPipe.process(CommonServerSecurityPipe.java:144)
    at com.sun.xml.ws.api.pipe.helper.PipeAdapter.processRequest(PipeAdapter.java:119)
    at com.sun.xml.ws.api.pipe.Fiber.__doRun(Fiber.java:961)
    at com.sun.xml.ws.api.pipe.Fiber._doRun(Fiber.java:910)
    at com.sun.xml.ws.api.pipe.Fiber.doRun(Fiber.java:873)
    at com.sun.xml.ws.api.pipe.Fiber.runSync(Fiber.java:775)
    at com.sun.xml.ws.server.WSEndpointImpl$2.process(WSEndpointImpl.java:386)
    at com.sun.xml.ws.transport.http.HttpAdapter$HttpToolkit.handle(HttpAdapter.java:640)
    at com.sun.xml.ws.transport.http.HttpAdapter.handle(HttpAdapter.java:263)
    at com.sun.xml.ws.transport.http.servlet.ServletAdapter.handle(ServletAdapter.java:163)
    at org.glassfish.webservices.JAXWSServlet.doPost(JAXWSServlet.java:145)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:688)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:770)
    at org.apache.catalina.core.StandardWrapper.service(StandardWrapper.java:1550)
    at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:281)
    at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:175)
    at org.apache.catalina.core.StandardPipeline.doInvoke(StandardPipeline.java:655)
    at org.apache.catalina.core.StandardPipeline.invoke(StandardPipeline.java:595)
    at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:161)
    at org.apache.catalina.connector.CoyoteAdapter.doService(CoyoteAdapter.java:331)
    at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:231)
```

```

at com.sun.enterprise.v3.services.impl.ContainerMapper$AdapterCallable.call(ContainerMapper.java:317)
at com.sun.enterprise.v3.services.impl.ContainerMapper.service(ContainerMapper.java:195)
at com.sun.grizzly.http.ProcessorTask.invokeAdapter(ProcessorTask.java:860)
at com.sun.grizzly.http.ProcessorTask.doProcess(ProcessorTask.java:757)
at com.sun.grizzly.http.ProcessorTask.process(ProcessorTask.java:1056)
at com.sun.grizzly.http.DefaultProtocolFilter.execute(DefaultProtocolFilter.java:229)
at com.sun.grizzly.DefaultProtocolChain.executeProtocolFilter(DefaultProtocolChain.java:137)
at com.sun.grizzly.DefaultProtocolChain.execute(DefaultProtocolChain.java:104)
at com.sun.grizzly.DefaultProtocolChain.execute(DefaultProtocolChain.java:90)
at com.sun.grizzly.http.HttpProtocolChain.execute(HttpProtocolChain.java:79)
at com.sun.grizzly.ProtocolChainContextTask.doCall(ProtocolChainContextTask.java:54)
at com.sun.grizzly.SelectionKeyContextTask.call(SelectionKeyContextTask.java:59)
at com.sun.grizzly.ContextTask.run(ContextTask.java:71)
at com.sun.grizzly.util.AbstractThreadPool$Worker.doWork(AbstractThreadPool.java:532)
at com.sun.grizzly.util.AbstractThreadPool$Worker.run(AbstractThreadPool.java:513)
at java.lang.Thread.run(Thread.java:722)
Caused by: org.apache.derby.client.am.SqlException: A network protocol error was encountered and the connection
has been terminated: The requested command encountered an unarchitected and implementation-specific condition for
which there was no architected message (additional information may be available in the derby.log file on the
server).
at org.apache.derby.client.am.SqlException.copyAsUnchainedSQLException(Unknown Source)
at org.apache.derby.client.am.Sqlca.chainDeferredExceptionsToAgentOrAsConnectionWarnings(Unknown Source)
at org.apache.derby.client.am.Sqlca.getJDBCMessage(Unknown Source)
at org.apache.derby.client.am.SqlException.getMessage(Unknown Source)
at org.apache.derby.client.am.SqlException.getSQLException(Unknown Source)
at org.apache.derby.client.am.Statement.executeQuery(Unknown Source)
at com.sun.gjc.spi.jdbc40.StatementWrapper40.executeQuery(StatementWrapper40.java:234)
at webservice.medisin.ntnu.no.InsertLater.getID_forQuery(InsertLater.java:213)
at webservice.medisin.ntnu.no.InsertLater.commit(InsertLater.java:157)
at webservice.medisin.ntnu.no.InsertLater.run(InsertLater.java:62)
... 1 more

```

4.8 Mail certificate expired

```

Sep 11, 2014 9:43:26 AM com.sun.enterprise.security.ssl.impl.SecuritySupportImpl checkCertificateDates
SEVERE: SEC5054: Certificate has expired: [
[
Version: V3
Subject: CN=smtp.gmail.com, O=Google Inc, L=Mountain View, ST=California, C=US
Signature Algorithm: SHA1withRSA, OID = 1.2.840.113549.1.1.5

Key: Sun RSA public key, 2048 bits
modulus:.....
.....

```

This occurs when the certificate obtained from the mail server expires. The solution is it to obtain a new one. To do this use

- **sh ./start.sh -fromlast -recreate_mail_certificates**

6. Upgrade

When there is a new release of the software (server or one or more clients), this part of the system can be updated. This will not require the entered data or user accounts, but requires a restart of all services.

To check the latest version and to display the current and latest versions use

- **start.sh -version**

Or from the administrator menu, select **m** (more commands) and select **version**.

If a new version is available, follow the steps below to upgrade.

1. Stop the server.
2. **start.sh -upgrade**
3. Start the server

If the upgrade was successful, then you would see a message similar to one below.

```
ed4929:~/tmp/server/egenvar$ sh ./start.sh -update
Starting Update.. @ 2013-12-30 15:36:35
A newer version is available. Latest version=1.03. Installing updates.
Download OK, installing started...
New version of web services installed, please restart.
Download OK, installing started...
New version of egenv tool was installed, please restart.
Download OK, installing started...
New version of web interface installed, please restart.
Download OK, installing started...
New version of server installed, please restart.
```

If the upgrade failed, then you would see a message similar to one below. Specially the version 1.00 and 1.01 versions may fail with the following error. If this happens replace the **Start_EgenVar1.jar** file in the installation directory with the one downloaded from http://tang.medisin.ntnu.no/~sabryr/latest/Start_EgenVar1.jar

```
ed4929:~/tmp/server/egenvar$ sh ./start.sh -update
java version "1.7.0"
Java(TM) SE Runtime Environment (build 1.7.0-b147)
Java HotSpot(TM) 64-Bit Server VM (build 21.0-b17, mixed mode)
Java version check. 1.7.0 >= 1.7 -- OK
Error 2a: invalid config file (or read access denied) -update. To use default values invoke
without arguments
Error: Failed to load configuration settings
Source root=/home/sabryr/tmp/server/egenvar/
Server root= null
```

7 User management

7.1. Activating user accounts.

When there is a user account request with a valid email, a user account is automatically created with an access level of 9. With this access level the users can login to the system but can not access any information allowed for logged in users (still the public search is allowed). For users to be able to search an access level of 8 or less is required. This access level could be set using the admin menu of the command line.

Level	Access restrictions
0	Admin level, full access
1	User level full access
2 - 6 (This group will may further subdivided in future versions)	Can access all information, but with restricted display (can not see full file paths for instance)
7	Same as level 6 except can not access details about other users
8	Same as level 7 except can not access details about files locations
9	Only public information is accessible

To view activated users

```
Select a function
0) Stop server
1) Get status messages
2) User management
3) Server controls and backup management
4) Database operations
5) Version
6) Refresh
7) Get interface details (server address, WSDL etc..)
8) Refresh_connection_resources
9) Exit this menu
: 2
```

```
Select a function
0) Display user details
1) Add a new user
2) Set access level for users
3) Block a user
4) Delete a user
```

```
5) List activated users
6) List user accounts not yet activated
7) Exit menu
: 5
Activated users:_____
1      USERNAME=xxxx@gmail.com    VALIDATED=1    ALLOWED_GROUPS=1000, 2001, 2003, 2000    LEVEL=0
2      USERNAME=sxxx@gmail.com    VALIDATED=1    ALLOWED_GROUPS=300    LEVEL=6

(STOP -to stop server| M -more commands):
```

To activate all inactive users

```
Select a function
0) Stop server
1) Get status messages
2) User management
3) Server controls and backup management
4) Database operations
5) Version
6) Refresh
7) Get interface details (server address, WSDL etc..
8) Refresh_connection_resources
9) Exit this menu
: 2

_____
Select a function
0) Display user details
1) Add a new user
2) Set access level for users
3) Block a user
4) Delete a user
5) List acitivated users
6) List user accounts not yet activated
7) Activate all users
8) Activate selected users
9) Exit menu
: 7
Result=true
```