Encrypting Passwords in Your LYNX-APP Config File (Windows)

As part of your LYNX-APP installation, you may wish add an additional layer of protection to the passwords in your *application.properties* configuration file by encrypting the password text—rather than leaving them in plain text.

These steps are for configuration on a Windows server. For Linux instructions, see LYNX-APP Installation for Linux.

Encryption When Moving LYNX to a New Server

If you're moving your LYNX installation to a new server, remember that encryption is done on a per-server basis, so your old encryption will not transfer.

In This Article:

Create an Environment Variable

- Right-click on Computer and navigate to Properties > Advanced System Settings > Environment Variables
- Create a new *system* variable called LYNX_ENCRYPT_PASSWORD.
  - *Note: Establishing a "user variable" is not a working alternative.*
- Enter a value for the variable. That will serve as the encryption password.

Due to issues with using the ^ and & characters, best practice is to make the encryption password longer and exclude all special characters.

Encrypt Your Passwords (LYNX-APP 1.2.3 and Newer)

- Open a new CMD prompt and navigate to the directory of the LYNX-APP
- Run the following command to encrypt the LYNX-WS password and the SIS DB LYNX schema password. You will need to run the command twice, once for each password.

```
java -jar lynx-app.jar --input "LYNX-WS or LYNX schema password" --algorithm "PBEWITHHMACSHA512ANDAES_256" --pas
scode %LYNX_ENCRYPT_PASSWORD%
```

- Copy the output and save somewhere.
- The LYNX-user password and the db-password can be the same, as long as you encrypt the value twice so that the encrypted value is unique for each set of credentials.
    - If you encrypt the same password once and use it for both sets of credentials, you will see an error like this in your logs:

```
Error creating bean with name 'appConfigurationImpl': Injection of autowired dependencies failed;
nested exception is java.lang.IllegalArgumentException: Password cannot be set empty
08:05:55.214 [main] ERROR org.springframework.boot.SpringApplication.reportFailure.815 - Application startup f
ailed.
```

- Modify the application.properties file

Encrypt Your Passwords (LYNX-APP 1.2.2 and Older)

Install Jasypt (or encryption tool of choice)

- Jasypt 1.9.3 is already packaged with the LYNX-APP-win.zip file within the ~\utility\ folder.
- You can also use another encryption tool that supports the algorithm PBEWITHSHA1ANDDESEDE.
- *Note: Jasypt password encryption, however, is not compatible with Java 8.*

Encrypt Passwords

- Open a new CMD prompt and navigate to the *bin* folder of the Jasypt directory.
- Run the following command to encrypt the LYNX-WS password and the SIS DB LYNX schema password. You will need to run the command twice, once for each password.

```
C:\jasypt-1.9.2\bin>encrypt.bat input=<enter LYNX-WS or LYNX schema password> password=%LYNX_ENCRYPT_PA
SSWORD% algorithm=PBEWITHSHA1ANDDESEDE
```

    - Copy the output and save somewhere.
    - The LYNX-user password and the db-password can be the same, as long as you encrypt the value twice so that the encrypted value is unique for each set of credentials.
        - If you encrypt the same password once and use it for both sets of credentials, you will see an error like this in your logs:

```
Error creating bean with name 'appConfigurationImpl': Injection of autowired dependencies failed;
nested exception is java.lang.IllegalArgumentException: Password cannot be set empty
08:05:55.214 [main]
ERROR org.springframework.boot.SpringApplication.reportFailure.815 - Application startup failed.
```

Modify the *application.properties* File

- Navigate to the *config* folder in the LYNX-APP directory.
- Open *application.properties* for editing.
- Uncomment the following line by removing the # sign. This is what allows Jasypt to decrypt encrypted passwords with a reference to the Environmental System Variable created earlier.

```
#jasypt.encryptor.password=${LYNX_ENCRYPT_PASSWORD:}
```

- Above that line, add **ONE** of the following algorithm lines, **based on your LYNX-APP version**
  - Version **1.2.3** and newer:

```
jasypt.encryptor.algorithm=PBEWITHHMACSHA512ANDAES_256
```

  - Version **1.2.2** and older:

```
jasypt.encryptor.algorithm=PBEWITHSHA1ANDDESEDE
```

- It should look like this when you are finished.

```
#################
# APPLICATION  #
#################
encoding=UTF-8

# Use this setting for encrypting passwords in the config file.
# The encryption password should be stored as system environment
# variable called LYNX_ENCRYPT_PASSWORD
jasypt.encryptor.algorithm=PBEWITHHMACSHA512ANDAES_256
jasypt.encryptor.password=${LYNX_ENCRYPT_PASSWORD:}
```

- Enter the encrypted passwords for LYNX-WS and the SIS DB LYNX Schema surrounded by ENC().

```
lynx-password=ENC(bzNkV2fEFX7iBchaiq8yUlt7lEeCsO8A)
```

```
db-password=ENC(KPpYiS5v7EYe2TXz1IEtuvZYejPWOo+G)
```

- Comment out the regular password assignment rows by adding a # sign. ****Be sure to remove the clear text credentials completely once you've tested and confirmed the encrypted credentials are working****
  - # lynx-password=
  - # db-password=

Test the Encrypted Password

- Restart the LYNX-APP service.
- Wait a few minutes to ensure that the process is running.
- Check the LYNX Dashboard and ensure that there are no errors. A green icon that says "APP OK" indicates that the LYNX-APP is running.
- If the LYNX-APP does not start correctly, check the *logs* folder within the LYNX-APP directory.
    - If you run into an error similar to the following, you may need to install the Java Cryptography Extension (JCE) on your server in order to use the PBEWITHSHA1ANDDESEDE algorithm.

```
Caused by: java.security.NoSuchAlgorithmException: no such algorithm: PBEWITHSHA1ANDDESED for provide
r SunJCE
    at sun.security.jca.GetInstance.getService(Unknown Source)
    at javax.crypto.JceSecurity.getInstance(JceSecurity.java:96)
    at javax.crypto.SecretKeyFactory.getInstance(SecretKeyFactory.java:204)
    at org.jasypt.encryption.pbe.StandardPBEByteEncryptor.initialize(StandardPBEByteEncryptor.java:689)
```