Glygen Array Development Notes **Date: 06/06/2018**

## Password Encryption Tool and Encrypted passwords in application.yml and/or application.properties

In order to protect the password that the system is using, all the passwords/secrets in application.yml and/or application.properties files should be encrypted and stored as ENC(……) in the above property files. An example entry:

From application.yml:

spring:

datasource:

url: jdbc:postgresql://localhost:5432/glygen

username: glygen

password: ENC(6fDqWRT97iuR7f1U4akTM918JIiZPKRT)

There is a tool under “util” package, named PasswordGeneratorUtil to generate the encrypted passwords to be put in above file. PasswordGeneraterUtil should be run with two arguments:

PasswordGeneratorUtil password DB

where password is the password that needs to be encypted.

Before running this tool, an environment variable JASYPT\_SECRET needs to be set. This secret is the one that the system will use to decrypt the password while running. Therefore the value used for JASYPT\_SECRET when generating the encrypted passwords should be the same one as the one that is set as environment variable when glygen array backend system is running.

## Password Encryption for the Glygen array application users

The system comes with an “admin” user that is imported into the database at initial setup. Its password should be encrypted before putting into data.sql file. The same tool “PasswordGeneratorUtil” can be used to generate such encryption as well. This time it needs to be called with only one argument

PasswordGeneratorUtil password

The resulting password, say xxxxx, should then be used in the data.sql as follows:

insert into users (userid, username, password, enabled, firstname, lastname, email, affiliation, affiliationwebsite, publicflag, logintype) values (1, admin, '{bcrypt}xxxxx’, true, 'Glygen', 'User', 'glycomic@uga.edu', 'UGA', 'uga.edu', true, 'LOCAL');