**Allegro Security Review**

M. Bret Blackford – August 2013

**Need**:

An automated process is desired to periodically provide an independent listing of Allegro user access and privilege changes. Listing should be emailed to an email distribution list including the Allegro administrator. The intent is to ensure that application users are not involved in the generation of the security report.

**Solution: High-Level Architecture:**

A Java application will be written that will access the Allegro Db (PRODACI1 using jdbc), write Allegro security setting to an Excel spreadsheet (using Apache POI), and then email to a distribution list.

The Java .jar file will (***AllegroSecurity-runnable.jar***) *-* be on an IT computer and IT will manage and control source and execution code as well as any associated files. CRON will be utilized to run the application at a predefined time (initially to be set at once a month).

**Specifics:**

The java application utilizes jdbc to call the production Allegro database (PRODACI1). Because jdbc is used the application needs to include the ojdbc6.jar file. The specific parameters required for the database call are obtained from the external ***allegro.security.properties.txt*** file (see **Exhibit A**). It has been requested that Information Technology (IT) obtain a unique ID and Password for use with this application.

The application obtains the following Allegro access/security information:

1. ***1.SecurityGroupAccessByMethod*** – lists the Allegro security access group in use and the associated table privileges within the system.
2. ***2.UserAccess*** – lists all Allegro security groups along with listing which user accounts are associated (and if user if active or inactive). Also included are the application methods available to the security group.
3. ***3.UserGroupAccess*** – Summary listing of which users and in which access group and if access is active or inactive.
4. ***4.AuditSecurity*** – Lists all changes to Allegro security for the past two months.
5. ***5.AdminAccess*** -- Lists who has “Admin” access to the Allegro application

The above information will then be exported to Excel (using the Apache POI API, which requires the poi-3.8-20121203.jar file). The name of the file created and the location where it is written is obtained from the ***allegro.security.properties.txt*** file.

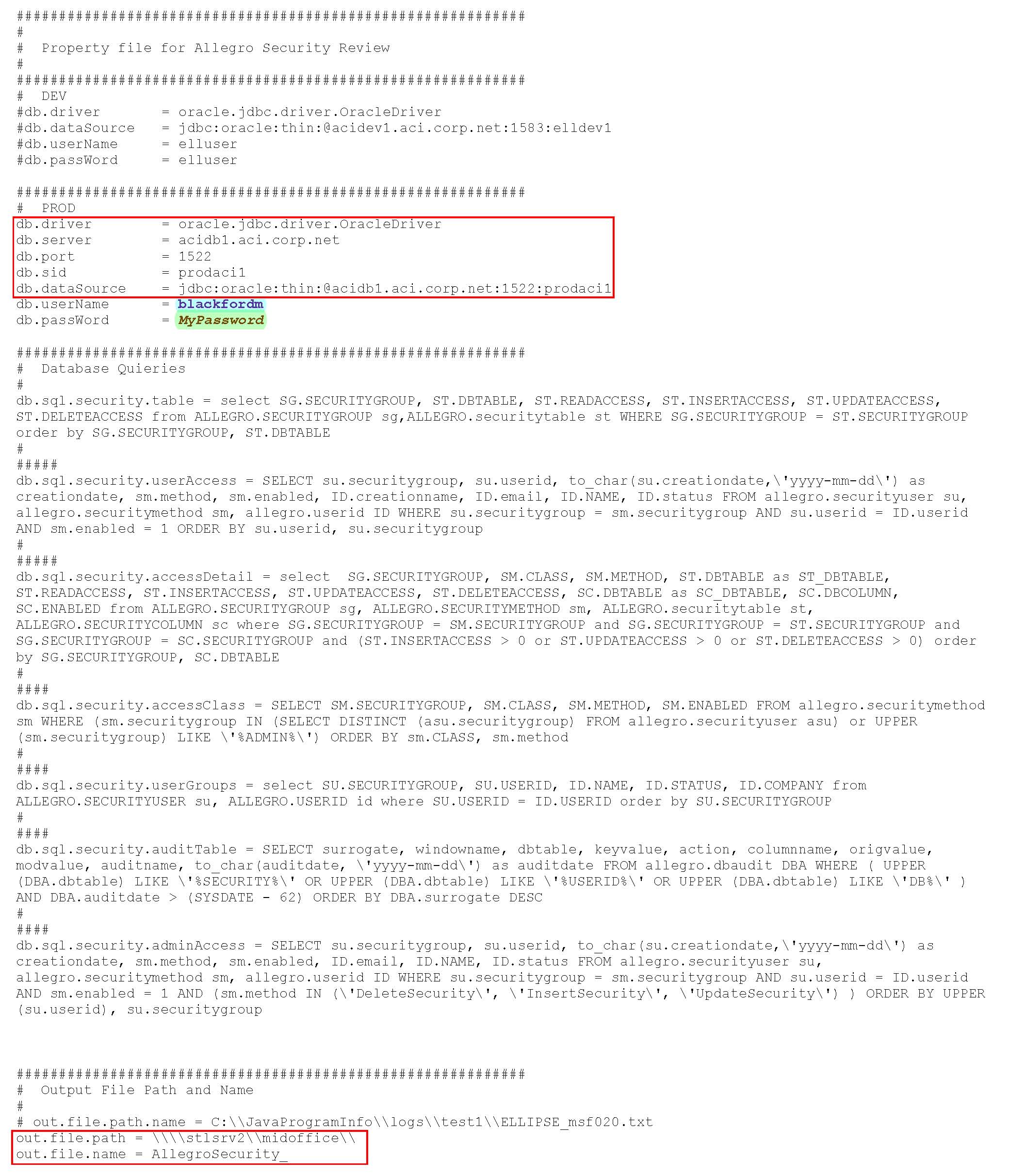
After the Excel file is created the file will be mailed to the [AllegroAdmins@ArchCoal.com](mailto:AllegroAdmins@ArchCoal.com) email group (see **Exhibit E**).

**Logging** – All key activity is logged using Apache’s log4j API. Log name and location is obtained from the ***log4j.properties*** file (see **Exhibit B**).

Application Dependencies are noted at **Exhibit F**.

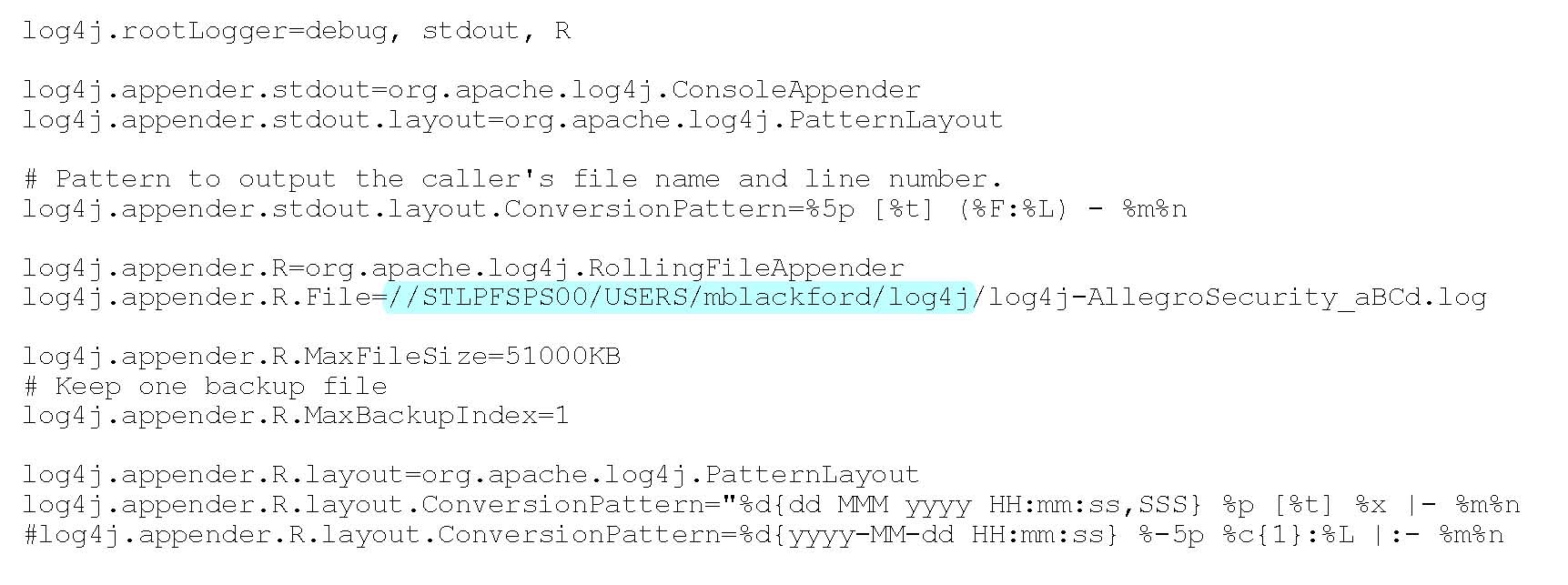
**Exhibit A**

**Allegro.security.properties.txt**



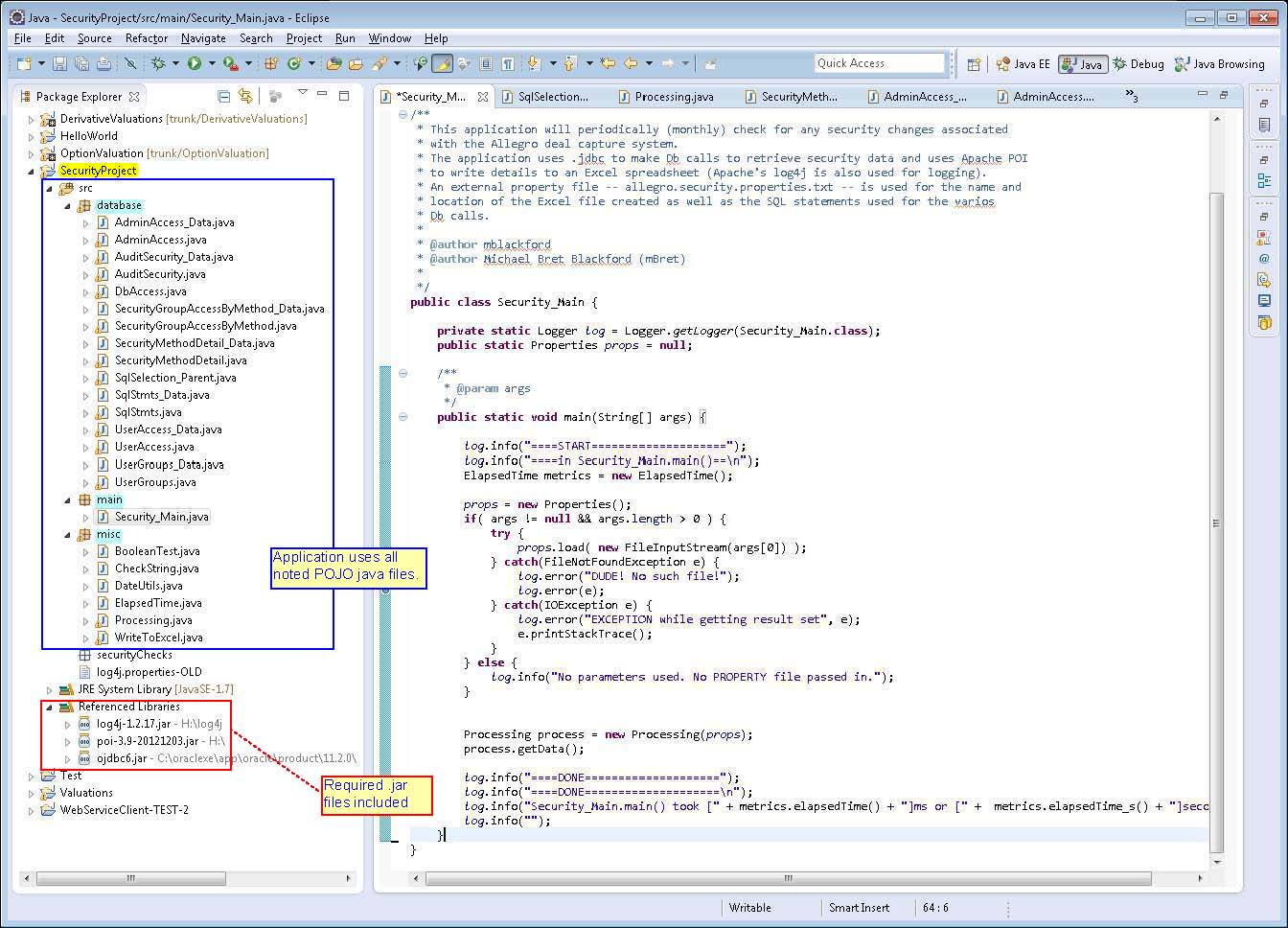
**Exhibit B**

**Log4j.properties**



**Exhibit C**

**Application Files**



**Exhibit D**

**Application Files (source code** *attached***)**



**Exhibit E**

**Application Output (automatic email** *attached***)**



**Exhibit F**

**Application Dependencies**

The following are required to run the application:

* It is preferable that the application is run in a Java 1.7 environment.
* The application’s .jar file (***AllegroSecurity-runnable.jar***) and the log4j.jar file (***log4j-1.2.17.jar***) need to be in the same directory.
* The property file (***allegro.security.properties.txt***) can be kept anywhere but the explicit path must be listed when the file is passed in as an application argument. An example of how to call the application is noted in the screenshot below.

***“Java –jar C:\\Bret\AllegroSecurity-runnable.jar \\stlpfsps00\users\mblackford\allegro.security.properties.txt”***

