NCICB

Ekagra

High Impact – High Value – Business Results

Common Security Module

CSM v3.2 Schema Migration Procedure

Version No: 1.0

Last Modified: 12/05/2006

Author : Kunal Modi

Team : Common Security Module (CSM)

Purchase Order# 34552

Client : National Cancer Institute - Center for Bioinformatics,

National Institutes of Health,

US Department of Health and Human Services

Document History

Document Location

The most current version of this document is located in CVS under security/docs.

Revision History

Version Number	Revision Date	Author	Summary of Changes
1.0	12/01/2006	Kunal Modi	Initial Draft

Review

Name	Team/Role	Version	Date Reviewed	Reviewer Comments
Vijay Parmar	Reviewer	1.0	12/05/2006	

Related Documents

More information can be found in the following related CSM documents:

Document Name

Table of Contents

CSM Authorization Schema Migration Procedure

1. Introduction

Based on the requirement for CSM v3.2 there were certain requirements which translated into CSM database changes. Also the introduction of encryption for password requires all the existing database passwords that are unencrypted

1.1 Purpose

This document provides the procedure to modify the existing database to be able to be used with the latest CSM v3.2 APIs. Also this procedure doesn't disturb the existing data sitting inside the database during this migration. The document also provides a Java based routine to modify any unencrypted password in the CSM's Authorization Database using CSM's internal encryption logic.

1.2 Scope

This document provides the migration procedure to move data from an existing schema to a new schema. It does not provide instructions to create a new authorization schema (see the *CSM Guide for Application Programmers*). Since prior installations of the authorization schema were performed on MySQL and Oracle databases, this migration procedure pertains to the MySQL and Oracle databases only. There is no information for SQL Server database.

2. Migration Procedure

2.1 MySQL Migration

The following procedure defines in detail the steps needed to update the MySQL database from an existing 3.1 authorization schema to a new 3.2 authorization schema:

1. Obtain the CSM API v3.2 Release from NCICB Download Center [http://ncicb.nci.nih.gov/download]
2. In the `MigrationScript3.2MySQL.sql` from the CSM API v3.2 Release, change the `<<database_name>>` with the name of the database.
3. Go to the directory which contains the executables for MySQL and provide the following command.

   ```
   mysql --user=[user_name] --password=[password] –h [hostname] [auth_schema] < MigrationScript3.2MySQL.sql
   ```

   - `[user_name]` is the user name used to connect the MySQL database
   - `[password]` is the password for the user name
   - `[hostname]` is the host URL where the MySQL database is hosted. If you are running this command from the same machine where MySQL is hosted, you do not need to provide this parameter.
   - `[auth_schema]` is the name of the database created using the new authorization schema.
   - `[MigrationScript3.2MySQL.sql]` is the file containing the data exported from the old schema, which needs to be loaded into the new schema
4. Verify that there are no errors in the SQL Script executed. Also make sure that the database has been appropriately updated.

2.2 Oracle Migration

The following procedure defines in detail the steps needed to update the Oracle database from an existing 3.1 authorization schema to a new 3.2 authorization schema:

1. Obtain the CSM API v3.2 Release from NCICB Download Center [http://ncicb.nci.nih.gov/download]
2. Log onto Oracle Server into the Schema where the CSM Database is present using either SQL Plus or TOAD or any other tool.
3. Copy all the SQL commands from `MigrationScript3.2Oracle.sql` from the CSM API v3.2 Release, and paste them on the SQL Editor/Console. Now execute all these commands in a batch.
4. Verify that there are no errors in the SQL Script executed. Also make sure that the database has been appropriately updated.

2.3 Password Encryption

With CSM v3.2 all the passwords in the CSM database are stored encrypted. For existing data a password encrypting utility java class is available that will encrypt the existing unencrypted passwords and store them in the database.

The following procedure defines in detail the steps needed to encrypt passwords as part of the migration process.

1. Obtain the CSM API v3.2 Release from NCICB Download Center [http://ncicb.nci.nih.gov/download]
2. Locate the PasswordEncrypter.java file available in the 'resources' folder.
3. Modify the following statements and replace the values with your database information.

   static String DATABASE_SERVER_NAME = "localhost";

   static String DATABASE_SERVER_PORT_NUMBER = "3306";

   static String DATABASE_TYPE = "MySQL";

   static String DATABASE_NAME = "csmstage";

   static String DATABASE_USERNAME = "root";

   static String DATABASE_PASSWORD = "admin";

   static String DATABASE_DRIVER = "org.gjt.mm.mysql.Driver";

   static String DATABASE_TABLE_NAME = "csm_user";

   static String DATABASE_TABLE_FIELD_NAME = "password";

4. Compile and run the PasswordEncrypter java class.
5. If everything is configured properly, the field DATABASE_TABLE_FIELD_NAME contents are encrypted.

Troubleshooting Checklist:-

Depending on the database and database driver used it is possible that an exception occurs. In case an exception occurs modify the following statement

ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE)) {

With

ResultSet.TYPE_FORWARD_ONLY, ResultSet.CONCUR_UPDATABLE)) {