

Common Security Module

CSM GAARDS Migration Module

Version No: 1.0

Last Modified: 04/29/2009

Author : Vijay Parmar

Team : Common Security Module (CSM)

Purchase Order# 3455

Client : National Cancer Institute - Center for Bioinformatics,

National Institutes of Health,

US Department of Health and Human Services

**Credits and Resources**

|  |  |  |  |
| --- | --- | --- | --- |
| CSM Contributors | | | |
| CSM Development Team |  | Guide | Program Management |
| Vijay Parmar 1 |  | Vijay Parmar 1 | Dave Hau 3 |
|  |  |  | Charles Griffin 1 |
|  |  | Charles Griffin 1 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| 1 Ekagra Software Technologies | 2 Science Applications International Corporation (SAIC) | 3 National Cancer Institute Center for Bioinformatics |  |
|  |  |  |  |

**Submitting a Support Issue**

A GForge Support tracker group, which is actively monitored by CSM developers, has been created to track any support requests. If you believe there is a bug/issue in the CSM software itself, or have a technical issue that cannot be resolved by contacting the [NCICB Application Support](#_Contacting_Technical_Support) group, please submit a new support tracker using the following link: <https://gforge.nci.nih.gov/tracker/?atid=131&group_id=12&func=browse> . Make sure to review any existing support request trackers prior to submitting a new one in order to help avoid duplicate submissions.

**Contacting Technical Support**

Technical support is available by contacting the **NCICB Application Support** group. There contact information is provided below:

|  |  |
| --- | --- |
| **NCICB Application Support** | <http://ncicb.nci.nih.gov/NCICB/support>  Telephone: 301-451-4384  Toll free: 888-478-4423 |

Document History

Document Location

The most current version of this document is located on the CSM website: <http://ncicb.nci.nih.gov/core/CSM>

Revision History

| Version Number | Revision Date | Author | Summary of Changes |
| --- | --- | --- | --- |
| 0.1 | 10/31/08 | Vijay Parmar | Initial Table of Contents |
| 0.2 | 10/05/08 | Vijay Parmar | Added new chapters |
| 0.3 | 04/13/2009 | Vijay Parmar | Added sections for new features such as Alternate Behavior, Standalone mode and misc configurability. |
|  |  |  |  |

Review

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Team/Role | Version | Date Reviewed | Reviewer Comments |
| Charles Griffin | Technical Manager | 1.0 |  |  |
| Bronwyn Gagne | Technical Writers |  |  |  |
|  |  |  |  |  |

Related Documents

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

|  |
| --- |
| Document Name |
| CSM 4.1 Technical Guide |
| CSM GAARDS User Migration Design Document |

These and other documents can be found on the CSM website: [NCICB CSM](http://ncicb.nci.nih.gov/NCICB/infrastructure/cacore_overview/csm). Additional information and FAQ regarding the CGMM are available at the [NCICB Wiki - CSM](https://wiki.nci.nih.gov/display/caCORE/Common+Security+Module+%28CSM%29).

Table of Contents

[1. Introduction 5](#_Toc228774614)

[2. Scope 5](#_Toc228774615)

[3. Using this Guide 6](#_Toc228774616)

[4. CGMM Overview 7](#_Toc228774617)

[4.1 Explanation 7](#_Toc228774618)

[4.2 Components involved with CGMM 9](#_Toc228774619)

[4.3 Security Concepts 10](#_Toc228774620)

[4.4 Minimal System Requirements 10](#_Toc228774621)

[5. CGMM API User Guide 11](#_Toc228774622)

[5.1 Workflow 11](#_Toc228774623)

[5.2 CGMM API Services 12](#_Toc228774624)

[5.3 Integrating with CGMM API 16](#_Toc228774625)

[5.4 Configurations for CGMM API 18](#_Toc228774626)

[6. Audit Logging 20](#_Toc228774627)

[6.1.1 Introduction 20](#_Toc228774628)

[6.1.2 Purpose 20](#_Toc228774629)

[6.1.3 Jar Placement 20](#_Toc228774630)

[6.1.4 Enabling CLM APIs in Integration with CGMM APIs 20](#_Toc228774631)

[6.1.5 Deployment Steps 22](#_Toc228774632)

[7. CGMM Tool User Guide 23](#_Toc228774633)

[7.1 Default Behavior 23](#_Toc228774634)

[7.1.1 Workflows 23](#_Toc228774635)

[7.1.1.1 Workflow 1: User logs in with CSM account 24](#_Toc228774636)

[7.1.1.2 Workflow 1-a: User has caGrid account 26](#_Toc228774637)

[7.1.1.3 Workflow 1-b: User does not have a caGrid account 28](#_Toc228774638)

[7.1.1.4 Workflow 2: User logs in with caGrid account 30](#_Toc228774639)

[7.1.1.5 Workflow 2-a: User is already migrated 31](#_Toc228774640)

[7.1.1.6 Workflow 2-b: User has CSM account 31](#_Toc228774641)

[7.1.1.7 Workflow 2-c: User doesn’t have a CSM account 33](#_Toc228774642)

[7.2 Alternate Behavior 33](#_Toc228774643)

[7.2.1 Workflows 34](#_Toc228774644)

[7.2.1.1 Workflow 35](#_Toc228774645)

[7.2.1.2 Workflow 1-a: User has caGrid account 35](#_Toc228774646)

[7.2.1.3 Workflow 1-b: User does not have a caGrid account 37](#_Toc228774647)

[7.3 CGMM Tool configuration 40](#_Toc228774648)

[8. Integrating CGMM with Container Managed Security 42](#_Toc228774649)

[8.1 Details 42](#_Toc228774650)

[8.2 Integration Steps 43](#_Toc228774651)

[9. CGMM Installation and Deployment 43](#_Toc228774652)

[9.1 Release Contents 44](#_Toc228774653)

[9.2 Installation Pre-requisites 45](#_Toc228774654)

[9.3 Deployment Checklist 48](#_Toc228774655)

[9.4 Deployment Steps 48](#_Toc228774656)

[Appendix A: CGMM Properties XSD 52](#_Toc228774657)

[Appendix B: Sample CGMM Properties File 56](#_Toc228774658)

[Appendix C: Sample Sync Description File 57](#_Toc228774659)

[Appendix D: Sample steps to install CGMM with reference implementation 58](#_Toc228774660)

[Appendix E: Sample Software setup steps for the reference implementation to test Container Managed Security Integration with CGMM. 61](#_Toc228774661)

[Appendix F: Sample Software setup steps for caArray – CGMM Container Managed Security Integration (Alternate Behavior) 63](#_Toc228774662)

[Glossary 66](#_Toc228774663)

**CSM GAARDS Migration Module Guide**

# Introduction

This document provides all the information application developers need to successfully integrate with the CSM GAARDS Migration Module (CGMM). The CGMM was chartered to provide a comprehensive solution to migrate existing web applications from CSM based authentication to GAARDS based authentication. caGrid is the underlying service oriented infrastructure that supports caBIG. The Grid Authentication and Authorization with Reliably Distributed Services (GAARDS) provides services and tools for the administration and enforcement of security policy in an enterprise Grid. GAARDS was developed on top of the Globus Toolkit and extends the Grid Security Infrastructure (GSI) to provide enterprise services and administrative.

# Scope

This document covers the CGMM API and CGMM Web application. It covers the workflows/scenarios handled by the CGMM. This document also briefly addresses the host application enhancements that are required to adopt the CGMM based authentication and migration features. The The caGrid information pertaining to the CGMM will be provided however, the caGrid, GAARDS, SyncGTS, Dorian etc details are out of scope for this document. For more information about caGrid and related technologies refer [caGrid Wiki](http://www.cagrid.org/mwiki/index.php?title=CaGrid).

# Using this Guide

Begin by reading the [CGMM Overview](#_CGMM_Overview) followed by [CGMM API User Guide](#_CSM_API_User) sections. It will give detailed knowledge and workflow for a developer on how to successfully integrate CGMM API into their applications. The [CGMM Tool User Guide](#_CGMM_API_User) section gives the default and alternate behavior workflows and details about the Authentication, Migration and/or New caGrid User creation. The section for Integration of CGMM with Container Managed Security provides details on achieving the same. Once the CGMM details and pertaining concepts are understood, read the [CGMM Installation and Deployment](#_CSM_Web_Services) section to know how to install and deploy CGMM Tool with an already working installation of a host application. The Appendix A, B and C provide the CGMM properties XSD, Sample CGMM properties configuration file and sample Sync Description configuration file. The Glossary is available to clarify abbreviations that are used in this document.

# CGMM Overview

The CSM GAARDS Migration Module provides two-tiered solution for existing web applications – namely a) migrate existing CSM accounts to caGrid accounts, b) act as the authentication ‘module’ for the host application. By doing so, the existing web applications gradually avail a single set of credentials (caGrid credentials) for authentication purpose. CGMM is created to address the following business/policy requirements:

* Avoid duplication of accounts for existing and new users. The application needs to provide a single set of credentials to access various application components.
* Use GAARDS based authentication
* Provisioning of new Users with Grid Identities
* To use caBIG approved Identity Providers and thus allowing federation of identity
* Configurable Look and Feel
* Configurable caGrid Identity Providers for authentication.

As in the Figure 4.1 overall architecture shows how existing host applications can integrate with CGMM and sort of off-load their authentication functionality to CGMM. The CGMM is expected to take over intercept and migrate CSM (local) accounts and enforce use of the caGrid account offered by various Identity providers in caBIG.

In this section, the overall architecture explanation, components involved in CGMM, Security Concepts and minimum system requirements are addressed.

# Explanation

The CGMM provides the host application the following solutions:

1. **Authentication** - validating and verifying a user’s CSM (local) credentials to initiate migration and validating and verifying a user’s caGrid Login ID and password against an Authentication Service. Once an already migrated user is authenticated, the CGMM passes the control to the host application by providing the users information and Grid Proxy.
2. **Migration** – allows migrating CSM user to caGrid user. The migration involves updating the CSM account (Login ID) information with the caGrid account (Login ID) in the CSM schema of the host application.
3. **New Grid User creation –** Creating of a new caGrid (Dorian) account for a new or existing User. Once the user has a caGrid account the CGMM can migrate the user for the host application.
4. **Configurable CGMM Tool –** that allows enabling or disabling the new user caGrid creation feature of the CGMM Tool. Also other information such as host application information and Authentication Service and Dorian Service information can be configured.
5. **CGMM API –** allows programmatic access and integration of the CGMM features.

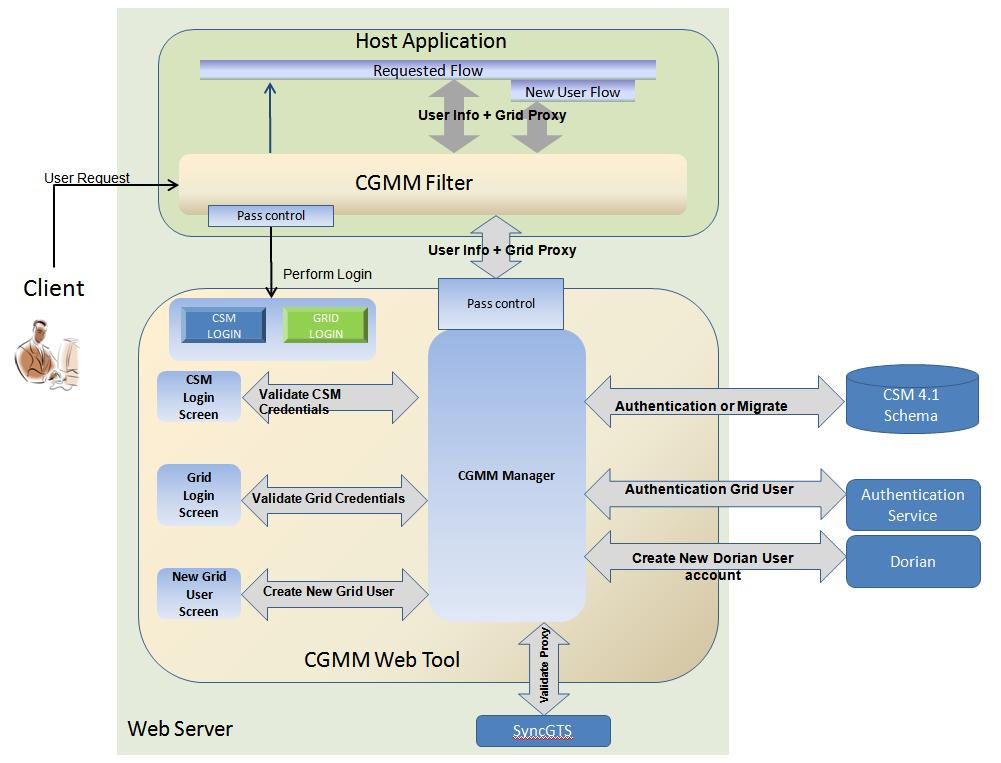


Figure 4.1 CGMM Architecture

The above diagram demonstrates overall architecture of the CGMM, the components involved and their interactions at a high level. As shown in the figure 4.1, CGMM is a web application that is hosted on the same application server as the Host web application. The Host application uses a migration Filter ‘CGMMMigrationFilter’, that is provided by the CGMM, to forward all un-authenticated user requests. The GAARDS components used are Authentication Service, Dorian Service and SyncGTS.

The overall flow for CGMM is as following

1. A user accesses Host applications secured page.
2. An Http filter intercepts the user’s request. The filter checks the session for user information attributes to verify if a user is logged in or not. If user is not logged in then the filter routes the user to CGMM.
3. CGMM module authenticates the user, migrates the user and obtains Grid proxy.
4. CGMM passes control back to the Host application and provides the Grid proxy and user information attributes. If the authenticated user did not have CSM credentials, then the control is passed to the new user creation workflow of the Host application. Otherwise the control is passed back to the user’s home page.
5. The filter intercepts the request and verifies user is logged in. Filter gets the Grid proxy and user information attributes. The filter sets this information in Session.
6. The filter gives up control to forward the request to the host application. The Host application uses the user information from session for authorization.

# Components involved with CGMM

The following are the minimum set of components involved in the Migration Module Framework. This section describes the components shown in the CGMM Architecture.

1. **CGMM Filter (in the host application)**

A new HTTP filter (provided as part of the CGMM) is configured by the host application to intercept and forward the user requests to the CGMM to either migrate the user account or to login the user into the Host application. Depending on whether the user is an existing application user or not, the control is passed back either to the login workflow or the new user creation workflow respectively.

1. **CGMM Tool**

CGMM tool is provided to assist in the migration of local CSM accounts to caGrid accounts and hence allow GAARDS based authentication via single set of credentials. CGMM is a separate web application that resides in the same container as the Host web application. CGMM also provides the Servlet Filter that is placed in front of the Host application intercepting and routing each user request for login or migration purpose. Detailed work flow of the migration module and the considered scenarios are provided in the next section.

1. **Authentication Service**

The IdPs registered on NCICB Production Grid will be used as the Identity Provider to validate user’s credentials. They will authenticate the user and provide a SAML token.

1. **Dorian**

The NCICB Production Dorian is used as a Federation Service to generate user’s grid identity. This Dorian instance will also host all the users migrated from individual local host application instances which are not associated to any other Identity Providers (IdPs).

1. **SyncGTS**

Sync GTS will be installed in CGMM for the Host application. Sync GTS daemon will keep the Host application in sync with the Grid Trust Fabric and update the CRL’s accordingly. Once the CGMM obtains Grid proxy from Dorian it will validate the proxy against the GTS to make sure the certificate is still valid and has not been revoked.

# Security Concepts

In order to successfully integrate CGMM with an existing host application, it is important to understand the definitions for components, systems and services involved as defined in . Application Developers should understand these concepts and begin to understand how they apply to their particular application.

| Concept | Definition |
| --- | --- |
| Host Application | The web application integrating with the CGMM Tool. The host web implements the CGMM Filter and all unsecured access to the web application will be forwarded to the CGMM Tool. |
| CGMM API | The CGMM API provides a CGMMManager interface to programmatically access all features of the CGMM Tool such as – Authentication of CSM users, authentication of Grid users, New Grid User creation etc. |
| CGMM Tool | The CGMM Tool is a web application that is deployed in the same container as the host application. The CGMM Tool does all the authentication, migration and new Grid user creation activities for the host application. |
| CSM User | Any user that has been provisioned in the CSM 4.1 Schema of the Host application. This user indicates the existence of the Host Application User with appropriate User Provisioning (assignment/association of Groups/ Protection Element/ Protection Groups to Role/Privilege). The user may or may not have a caGrid account or caGrid identity. |
| caGrid User | Any user that has already created an account or registered to caGrid. The registration provides the login credentials for the user. Once a user has registered with caGrid and obtained an account, that user can be authenticated using the valid credentials via the GAARDS security framework or via Authentication Service or Dorian Service. |
| Migration of CSM Account to Grid Account | The act of updating the CSM Login Name, in the CSM 4.1 Schema’s CSM\_USER table, with the caGrid User identity and marking the particular user as migrated is known as migration of CSM account to caGrid account. An already migrated user can be authenticated using caGrid Login ID and password. |

Table 4.1 Security concept definitions

# Minimal System Requirements

The following software is required and not included with CGMM Software as listed in Table 4-3. The software name, version, description, and URL hyperlinks are indicated in the table.

|  |  |  |  |
| --- | --- | --- | --- |
| Software | Description | Version | URL |
| JDK | The J2SE Software Development Kit (SDK) supports creating J2SE applications | 1.5.0\_11  or higher | <http://java.sun.com/j2se/1.5.0/download.html> |
| Oracle | Database Server† | 9i | <http://www.oracle.com/technology/products/oracle9i/index.html> |
| MySQL | 5.0.27 | <http://dev.mysql.com/downloads/mysql/5.0.html> |
| JBoss | Application Server† | 4.0.5 | <http://labs.jboss.com/jbossas/downloads> |
| Tomcat | 5.5.20 | <http://tomcat.apache.org/download-55.cgi> |
| Ant | Build Tool | 1.6.5  or higher | <http://ant.apache.org/bindownload.cgi> |
| caGrid | caGrid software | 1.2 | <https://cabig.nci.nih.gov/workspaces/Architecture/caGrid/> |
| Globus | Globus ToolKit | 4.0.3 | [Globus WS-Core with WS-Enum Support](http://gforge.nci.nih.gov/frs/download.php/1334/ws-core-enum-4.0.3.zip) |

Table 4‑3 Minimal software requirements

† Only one is required.

# CGMM API User Guide

The CGMM features are available as API’s. The CGMM API primarily consists of the CGMMManager interface. The CGMM API was created for host applications that wish to incorporate the CGMM features in their code base. Integration of CGMM API is not a requirement and is completely up to the development team to either adopt the CGMM tool (least changes to host application way) or integrate the CGMM functionality via API (more changes to host application authentication and migration logic).

Alternatively, the CGMM API can be used in different ways to suit the host applications requirement or also in standard java applications that can be run via automated scripts.

The CGMM API User guide starts with a recommended workflow for integration. It is followed by the detailed description of the CGMM API services available. The following sections demonstrate how to integrate with CGMM API and configuration requirements.

# Workflow

This workflow section outlines the basic steps, both strategic and technical, for successful CGMM API integration.

1. Read the *CSM GAARDS Migration Module Guide* (this document).  It provides an overview, workflow, and specific deployment and integration steps and CGMM Tool user guide.
2. Decide which services you would like to integrate with your host application.  If the application should authenticate CSM (local) users against an LDAP or other directory, select CSM Authentication.  If the application should authenticate caGrid users against Authentication Service(s), select caGrid Authentication. If the host application would like to create new caGrid users, select new caGrid user creation feature. The migration feature should be used to migrate the CSM (local) user ID to the caGrid ID of the user.  See the CSM API Services section more details.
3. Add the StartSyncGTSServlet servlet to your host web application. See section Adding Auto SyncGTS Servlet for more details.
4. Integrate the application code using the integration as shown in the following sections
5. Test and refine CGMM integration with your application.  Confirm that your CGMM API integration meets requirements.

# CGMM API Services

The CGMM API’s consist primarily of the following features – Authentication, Migration and new caGrid User creation, Synching with the caGrid Trust Fabric.

**CGMMManager**

The CGMM Manager is an interface that provides the following functionality. It is implemented by the CGMMManagerImpl class available in the CGMM API’s. Table 5.1 shows all the CGMMManager API methods that does the following.

* caGrid User Authentication and CSM Authentication
* Migration of CSM Account to caGrid Account
* New caGrid User Creation
* Miscellaneous – get CSM User details, get caGrid User Attributes and Attribute Map and get Authentication Service URL Map.

| Class/Method | Description |
| --- | --- |
| public interface CGMMManager | This CGMM Manager provides all the CSM GAARDS user migration related services offered by Common Security Module. This interface defines the contract for any class that wants to act as CGMMManager. It defines the methods required for authenticating CSM users, authenticating users with caGrid based accounts and creating accounts on the configured Dorian. The CGMMManager is implemented by CGMMManagerImpl. CGMMManager can be configured using the cgmm-properties.xml configuration file. |
| public boolean **performCSMLogin**(  String userIDCSM,  String password) throws CGMMInputException, CGMMConfigurationException, CGMMCSMAuthenticationException; | Authenticates user against the configured CSM credential provider. The CSM credential provider configuration can be done via CGMM configuration file.  **Parameters:**  **userIDCSM** The CSM User Login ID of the User.  **password** The Password of the CSM User.  **Returns:**  true if login is successful.  **Throws:**  CGMMCSMAuthenticationException is thrown when the credentials are invalid or other errors occur during validation  CGMMConfigurationException is thrown when there is a CGMM configuration exception  CGMMInputException is thrown when there is an error in specifying User Id/password. |
| public CGMMUser **getUserDetails**(  String loginID) throws CGMMInputException, CGMMConfigurationException, CGMMCSMUserException ; | Updates the CGMMUser object with CSM User Details. Retrieves CSM user information from CSM schema using the CSM API's AuthorizationManager and populates teh CGMMUser  **Parameters:**  **loginID** The Login ID of the User available in CSM. This ID can be a Grid ID or CSM Local User ID.  **Returns:**  CGMMUser  **Throws:**  CGMMCSMUserException is thrown when there is an error obtaining the CSM User from the CSM schema.  CGMMConfigurationException is thrown when there is a CGMM configuration exception  CGMMInputException is thrown when there is an error in specifying User Id/password |
| public boolean **isUserMigrated**(  String userIDCSM) throws CGMMInputException, CGMMConfigurationException, CGMMMigrationException ; | Checks if the user is migrated or not. If the user is migrated then the Grid ID of the user is available in the CSM schema and the user is marked as migrated. if the user is not migrated, the CSM ID of the user is available in the CSM schema and hence the user isnt marked as migrated.  **Parameters:**  **userIDCSM** The CSM User Login ID of the User.  **Returns:**  false if the user is not migrated.  **Throws:**  CGMMMigrationException is thrown when there is an error in migrating a CSM User to Grid User  CGMMConfigurationException is thrown when there is a CGMM configuration exception  CGMMInputException is thrown when there is an error in specifying User Id/password. |
| public boolean **migrateCSMUserIDToGridID**(  String userIDCSM,  String userIDGrid) throws CGMMMigrationException, CGMMConfigurationException ; | Updates the users CSM ID with the user's Grid ID and also marks the user as migrated in the CSM Schema.  **Parameters:**  **userIDCSM** The CSM User Login ID of the User.  **userIDGrid** The login ID for users Grid account.  **Returns:**  false if migration failure.  **Throws:**  CGMMConfigurationException is thrown when there is a CGMM configuration exception  CGMMMigrationException is thrown when there is an error in migrating a CSM User to Grid User |
| public GlobusCredential **performGridLogin**(  String loginIDGrid,  String password,  String authenticationServiceURL) throws CGMMInputException, CGMMConfigurationException, CGMMGridDorianException, CGMMGridAuthenticationServiceException, CGMMAuthenticationURLException ; | Authenticates the Grid credentials of the user against the provided Authentication Service URL.  **Parameters:**  **loginIDGrid** The login ID for users Grid account.  **password** The password for user Grid account.  **authenticationServiceURL** The URL for authentication service.  **Returns:**  GlobusCredential  **Throws:**  CGMMGridAuthenticationServiceException is thrown when there is an exception in caGrid's Authentication Service.  CGMMGridDorianException is thrown when there is a Dorian exception  CGMMConfigurationException is thrown when there is a CGMM configuration exception  CGMMInputException is thrown when there is an error in specifying User Id/password.  CGMMAuthenticationURLException is thrown when there is a Authentication Service URL specification exception. |
| public String **createDorianAccount**(  CGMMUser cgmmUser,  String dorianURL) throws CGMMAuthenticationURLException, CGMMGridDorianException, CGMMGridDorianUserPropertiesException; | Creates a caGrid (Dorian) account.  **Parameters:**  **cgmmUser** The CGMMUser object populated with required fields for Dorian account creation.  **dorianURL** The URL for Dorian Service  **Returns:**  Confirmation Message with the status of the Dorian account creation.  **Throws:**  CGMMGridDorianUserPropertiesException is thrown when there is an error in specifying Dorian User properties.  CGMMGridDorianException is thrown when there is a Dorian exception  CGMMAuthenticationURLException is thrown when there is a Authentication Service URL specification exception. |
| public SortedMap **getAuthenticationServiceURLMap**() throws CGMMConfigurationException; | Provides the SortedMap of Authentication Service URLS.  **Returns:**  SortedMap of Authentication Service URLs. The Key is the Authentication Service Name and the value is Authentication Service URL  **Throws:**  CGMMConfigurationException is thrown when there is a CGMM configuration exception |
| public HashMap<String, String> **getUserAttributesMap**(  String loginIDGrid,  String password,  String authenticationServiceURL) throws CGMMInputException, CGMMConfigurationException, CGMMGridDorianException, CGMMGridAuthenticationServiceException, CGMMAuthenticationURLException; | Returns User Attributes Map based on the authenticated user.  **Parameters:**  **loginIDGrid** The login ID for users Grid account.  **password** The password for user Grid account.  **authenticationServiceURL** The URL for authentication service.  **Returns:**  userAttributeMap containing the Users Attributes such as First,Last Name and Email Id.  **Throws:**  CGMMGridAuthenticationServiceException is thrown when there is an exception in caGrid's Authentication Service.  CGMMInputException is thrown when there is an error in the input provided  CGMMConfigurationException is thrown when there is a CGMM configuration exception  CGMMGridDorianException is thrown when there is an exception in caGrid's Dorian  CGMMGridAuthenticationServiceException is thrown when there is an exception in caGrid's Authentication Service.  CGMMAuthenticationURLException is thrown when there is a Authentication Service URL specification exception. |

Table 5.1 CGMM API – CGMMManager

## 

The CGMM API provides a CGMMManager for user authentication for CSM, user authentication for caGrid, user migration, new caGrid user creation etc as shown in Table 5.1. The CGMMManagerImpl class implements the CGMMManager Interface. Developers can easily incorporate the service into their host applications with simple configuration and coding changes to their applications.

**Importing the CGMM Authentication API**

To use the CGMM API’s CGMMManager, add the import statements (last two) as shown in to the action classes.

**import** gov.nih.nci.security.cgmm.CGMMManager;

**import** gov.nih.nci.security.cgmm.CGMMManagerImpl;

**import** gov.nih.nci.security.cgmm.beans.CGMMUser;

**import** gov.nih.nci.security.cgmm.exceptions.CGMMException;

**import** gov.nih.nci.security.cgmm.exceptions.CGMMConfigurationException;

**import** gov.nih.nci.security.cgmm.exceptions.CGMMConfigurationException;

**import** gov.nih.nci.security.cgmm.exceptions.CGMMInputException;

Figure 5.1 Example ‘sampleHostApplication’ host application – Import statements in an action class

**Obtaining the CGMMManager**

illustrates an example of how to use the CGMMManager in the ‘sampleHostApplication’ host application.

CGMMManager cgmmManager = **null**;

**try** {

cgmmManager = **new** CGMMManagerImpl();

} **catch** (CGMMConfigurationException e) {

System.out.println("ERROR Unable to obtain CGMMManager");

}

Figure 5.2 Example code to use the CGMM API - CGMManager class in the ‘sampleHostApplication’ host application

**Authenticating Users**

illustrates an example of how to use the CGMMManager in the ‘sampleHostApplication’ host application for authenticating CSM users.

String username = Form.getUsername());

String password = Form.getPassword());

//perform CSM Login

**try**{

cgmmManager.performCSMLogin(username, password);

} **catch** (CGMMException e1) {

System.*out*.println("ERROR Unable to perform CSM login");

}

Figure 5.3 Example code to perform authentication for CSM users in the ‘sampleHostApplication’ host application.

**Migrating Users**

4 illustrate an example of how to use the CGMMManager in the ‘sampleHostApplication’ host application for migrating users.

String userIDCSM = Form.getUsername());

String userIDGrid = Form.getGridID());

//perform Migration

**try**{

**b**oolean isMigrated = cgmmManager.isUserMigrated(username);

if(!isMigrated)

cgmmManager.migrateCSMUserIDToGridID(userIDCSM, userIDGrid);

} **catch** (CGMMException e1) {

System.*out*.println("ERROR Unable to migrate the user.");

}

Figure 5.4 Example code to migrate users in the ‘sampleHostApplication’ host application.

**Integrating Auto Start SyncGTS servlet**

To integrate the StartSyncGTSServlet in the host application, add the configuration shown in Figure 5.5 to the web.xml of the host application. This configuration is required since it is the only way to ensure the server of the host application is in sync with the caGrid Trust fabric before any secured caGrid Services can be invoked.

<servlet>

<servlet-name>Start Auto Sync GTS </servlet-name>

<servlet-class>

gov.nih.nci.security.cgmm.util.StartSyncGTSServlet

</servlet-class>

<load-on-startup>2</load-on-startup>

</servlet>

Figure 5.5 the web.xml configuration to integrate StartSyncGTSServlet in the ‘sampleHostApplication’ host application.

# Configurations for CGMM API

For successful integration of CGMM API into a host web application, the following configuration files have to be configured appropriately and correctly. Section 5.5 shows the configuration files needed for CGMM.

| Configuration file | Description |
| --- | --- |
| Cgmm-properties.xml | * Required configuration file to specify the CGMM information, Host Application information and Authentication Service/Dorian information * Sample shown in Appendix B * Refer the cgmm-propertiex.xsd for more information. * The CGMMManager retrieves this file based on the System property   ' gov.nih.nci.security.cgmm.properties.file’. |
| Sync-description.xml | * Required configuration file for the StartSyncGTSServlet. * Sample shown in Appendix C. * The CGMMManager retrieves this file based on the System property   ' gov.nih.nci.security.cgmm.syncgts.file’. |
| Cgmm.login.config | * Required to configure the CSM Authentication part of the CGMMManager API. * Specifies the Login Module to be used by the CGMMManager (that internally uses CSM AuthenticationManager) to authenticate CSM users. * The CGMMManager retrieves this file based on the System property 'gov.nih.nci.security.cgmm.login.config.file’. * If the Jboss login-config.xml is configured with Login Module for the host application then the System property 'gov.nih.nci.security.cgmm.login.config.file’ is ignored. |
| ApplicationSecurityConfig.xml | * Required to configure the CSM Authorization part of the CGMMManager API that is used to migrate CSM users or obtain CSM User information. * This file points to a hibernate.cfg.xml file for the host application. * Refers to the <<name>>.hibernate.cfg.xml based on the specified path. * The CGMMManager retrieves this file based on the System property   ' gov.nih.nci.security.configFile’. |
| <<name>>.hibernate.cfg.xml | * Required along with ApplicationSecurityConfig.xml file. * It points to the CSM Schema for the host application. * Replace <<name>> with the host application context name. |

Figure 5.6 CGMM configuration files.

# Audit Logging

# Introduction

In an effort to make CGMM compliant with CRF 21/ part 11, CCGMM will provide auditing and logging functionality. Audit logging capability will be provided through the Common Logging API that is available from clm-\*.jar. Audit logging is configurable by the client application developer via an application property configuration file. By placing the clm.jar along with the application property configuration file in the same class path as the cgmmapi.jar file, the client application will be able to utilize the inbuilt audit logging functionality. The logging results will be saved into a database or a flat text file depending on the configuration.

# Purpose

This section serves as a guide to help developers integrate Audit Logging for the CGMM API or CGMMWeb. This section outlines a step-by-step process that addresses what developers need to know in order to successfully integrate Common Logging Module (CLM), including:

* Jar placement
* Configuring the JDBC Appender configuration file or the regular log4j configuration file

# Jar Placement

The Audit Logging Application is available as a JAR, called clm-4.1.jar. This jar along with the cgmmapi.jar needs to be placed in the classpath of the application. The clm-4.1.jar should be placed in the common lib directory of JBoss.

# Enabling CLM APIs in Integration with CGMM APIs

The CGMM Manager service exposed by CGMM has been enabled for the purpose of Audit and Logging using the CLM. If configured properly, client applications using the CGMM APIs can enable the internal CLM based Audit and Logging capabilities.

**Event Logging**

The CGMM Manager has been modified to enable the logging of every event that the user performs. For Authentication/Login, Migration, New User Creation etc Services, the CGMM APIs log the events of the user.

The CGMM Web can perform all the above audit and logging services because it uses the CGMM APIs (which use CLM APIs) to perform operations on the database.

Since the CLM APIs are based on log4j, the following logger name is used in the CGMM APIs to perform the event logging.

Logger Name:

CGMM.Audit.Logging

The log4j log level used for all the event logs is INFO

In order to enable these loggers, they should be configured in the log4j.xml config file of Jboss as shown in JDBC Appender section below.

**Common Logging Database**

This is the persistence storage that the JDBC appender uses to store the Audit Logs. The Log Locator application of CLM connects to this database to allow the user to browse the logs.

**JDBC Appender**

To persist these Audit logs the CLM provides an asynchronous JDBC Appender. Thus, an application that wants to enable the audit logging for CGMM APIs should also configure this Appender. A sample log4j entry is show below.

<?xml version="1.0" encoding="UTF-8" ?><!DOCTYPE log4j:configuration SYSTEM ".\log4j.dtd">

<log4j:configuration xmlns:log4j='http://jakarta.apache.org/log4j/'>

<appender name="CLM\_APPENDER" class="gov.nih.nci.logging.api.appender.jdbc.JDBCAppender"> <param name="application" value="<<APPLICATION\_NAME>>" /> <param name="maxBufferSize" value="1" /> <param name="dbDriverClass" value="org.gjt.mm.mysql.Driver" /> <param name="dbUrl" value="jdbc:mysql://<<SERVER\_NAME>>:<<PORT>>/<<CLM\_SCHEMA\_NAME>>" /> <param name="dbUser" value="<<DB\_USER>>" /> <param name="dbPwd" value="<<PASSWORD>>" /> <param name="useFilter" value="true" /> <layout class="org.apache.log4j.PatternLayout"> <param name="ConversionPattern" value=":: [%d{ISO8601}] %-5p %c{1}.%M() %x - %m%n" /> </layout>

</appender>

<category name=" CGMM.Audit.Logging">

<level value="info" /> <appender-ref ref="CLM\_APPENDER" /> </category>

</log4j:configuration>

Figure 5.7 Example log4j.xml file

**NOTE:** In order to CLM features without using CGMM, the client application can separately download and install CLM. In this case CLM can be used (even without using CCGMM) to provide event logging and automated object state logging capabilities using the special appender and schema. Also the log locator tool can be used for the purpose of viewing the logs.

# Deployment Steps

In order for a client application to enabling the Audit Logging capabilities provided by CGMM (via CLM), the following steps must be performed:

Step 1: Create and Prime MySQL Logging Database

* 1. A database has to be created which will persist the audit logs that are generated as a basis of usage of the CGMM APIs
  2. Refer to the CLM’s guide for application developers for creating and priming the database for storing the audit logs.

Step 2: Configure the log4j.xml file for JBoss

* 1. Use the sample log4j file provided in the CGMM’s release to configure the log4j.xml file for JBoss. (see figure above)
  2. Replace the <<APPLICATION\_NAME>>, <<SERVER\_NAME>>, <<PORT>> and the <<CLM\_SCHEMA\_NAME>> with corresponding values where the schema created in Step 1 is hosted.
  3. Replace the values for the <<DB\_USER>> with the user name that has access on the schema. Also replace the <<PASSWORD>> with the corresponding password for the user.
  4. Based on whether the application wants to enable the event audit logging for Authentication & Authorization or object state audit logging for the Authorization; the corresponding logger needs to be configured. **Note:** The names of loggers must not differ from the sample.
  5. In case of CGMM Web Tool, the same log4j config file can be used.

Step 3: View the Logs

* 1. CLM provides a web-based locator tool that can be used to browse audit logs.
  2. The configuration steps for setting up the browser are mentioned in the CLM’s guide for application developers.

# CGMM Tool User Guide

The CGMM Tool is a web application that, on behalf of the host application, allows authentication of CSM/caGrid users, migration of CSM user account to caGrid user account and/or creation of new caGrid accounts for users. The CGMM overview section and the CGMM API User Guide provide a very detailed description and information about CGMM and CGMM API.

CGMM Tool is configurable and created by considering customizations by/for the host applications. The CGMM tool involves low effort of modification and configurations that are required by the host applications. The CGMM API, on the other hand, allows full integration of CGMM features programmatically, thus not requiring the use of CGMM Tool.

The CGMM Tool User Guide demonstrates the implemented CGMM Default Behavior and Alternate Behavior workflows/scenarios followed by the configurable features of the CGMM Tool.

# Default Behavior

The Default behavior is the original behavior/workflows available with the CGMM 0.5 release. The default behavior is meant for existing web applications that would like to utilize CGMM Web application for the following a) Authentication, b) Migration, and c) New caGrid User Creation. The default behavior also assumes the use of Servlet filter (CGMFilter) by the host application to intercept and interpret logged in/migrated users forwarded by the CGMM Web application.

# Workflows

The CGMM Tool’s default behavior allows multiple scenarios/workflows based on the User. The user may or may not have CSM account. The user also may or may not have a caGrid Account. Hence based on that there are four different scenarios addressed by the CGMM Tool. The scenarios are as following:

1. User logs in with CSM account and User does not have a caGrid account.
2. User logs in with CSM account and User has a caGrid account
3. User logs in with caGrid account and User does not have a CSM account
4. User logs in with caGrid account and User has a CSM account.

**Note**: The CGMM tool DO NOT addresses the scenario where a User does not have a CSM (local) Account and does not have a caGrid account. To host application needs to address this scenario.

Let us look at the user interface workflow of the CGMM by going through each of the scenarios mentioned above. Shown below is the CGMM Tool Home page. The home page provides details to the user regarding how to proceed using the tool.

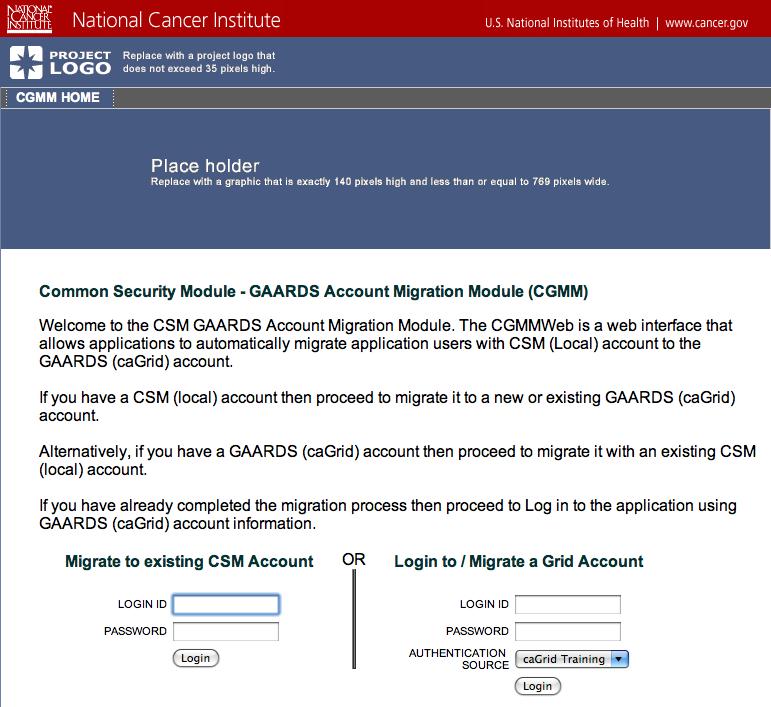


Figure 6.1 CGMM Home page.

# Workflow 1: User logs in with CSM account

In this scenario, the User has the CSM account. The user logs in by providing the username and password and clicking on the Login button.

If the Login Id or Password is invalid, the CGMM tool shows an error.

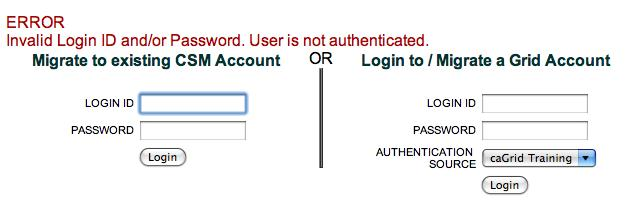


Figure 6.1 CGMM - CSM Login error.

If the Login Id and password are valid, the CGMM tool takes the user to CSM to GAARDS Account Migration page. In this page, the tool allows the user to either login using existing caGrid account or create a new caGrid account.

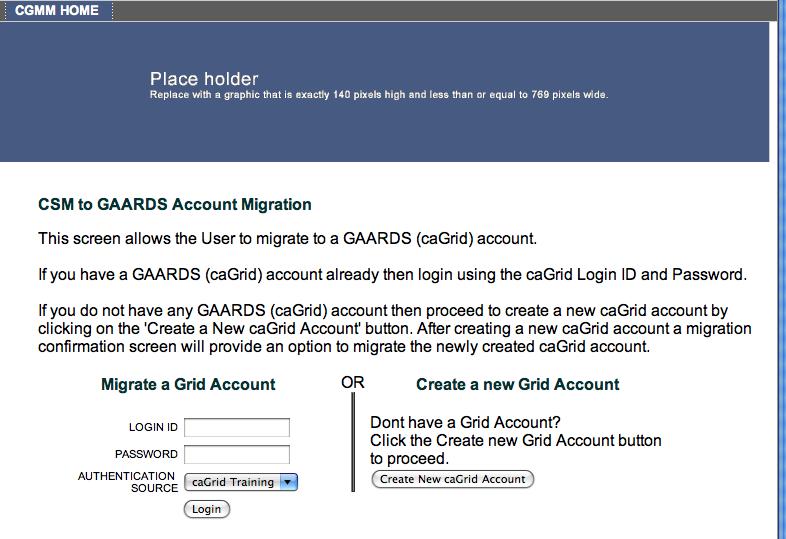


Figure 6.3 The CSM Login success page / Grid Login Page.

# Workflow 1-a: User has caGrid account

In this case the user has an existing caGrid account. Hence the user can proceed to migrate caGrid account by providing the Login ID, Password and also selecting the Authentication Source (Authentication Service).

**User logins in with caGrid Login ID and password**

Once the user clicks Login button, the CGMM Tool validates the caGrid account credentials provided by the user. If the credentials are valid then the CGMM Tool will show the ‘Confirm Migration’ screen to the user.



Figure 6.4 CSM to GAARDS Account migration page.

**User chooses to migrate his account**

On the migration confirmation page, the user has the option to cancel the migration or confirm it. Once the user proceeds to migrate by clicking on the ‘Yes, Migrate my CSM Account’ button, the CGMM will migrate the CSM account to caGrid account in the CSM Schema of the host application. CGMM will also mark the user as migrated. Once the migration process is complete, the CGMM Tool takes the user to the migration confirmation page. The user now has the only option of Log in to the host application.

When the user clicks the ‘Log in to <<Host Application Name>>’ button, the CGMM proceeds to log in the user using the caGrid account information. The CGMM tool then populates the HTTP Request with the caGrid User information and the Users Grid Proxy as request attributes and forwards the request to the Host application. The request is forwarded to the Host Applications User Home page, specified in the CGMM properties configuration, and the CGMM relieves the control to the Host application.



Figure 6.5 Migration complete page.

The User clicks on the ‘Log in to <<host application name>>’ button and is forwarded by the CGMM to the Host application User Home page. Figure 6.6 shows the User Home page for the ‘HostWeb’ web application, available as a reference implementation.



Figure 6.6 Migration complete page/ Host application User Home page.

# Workflow 1-b: User does not have a caGrid account

In this case the user has an does not have an existing caGrid account. Hence the user can proceed to obtain a new caGrid account . As shown in Figure 6.3, the User clicks on the ‘create new caGrid account’ button.

The User must provide all the information requested to proceed.

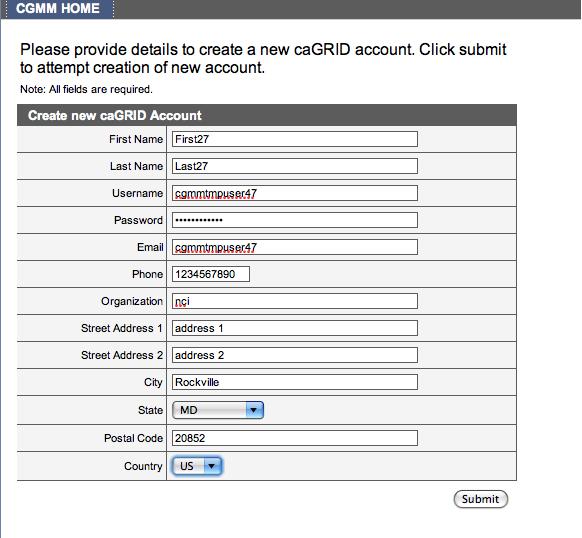


Figure 6.7 New caGrid Account page.

The user provides all information and clicks on the ‘submit’ button to attempt to create a new caGrid account.

The CGMM attempts to create a new caGrid (Dorian) account with user details provided by the user. The CGMM obtains the Dorian URL from the CGMM Properties configuration file.

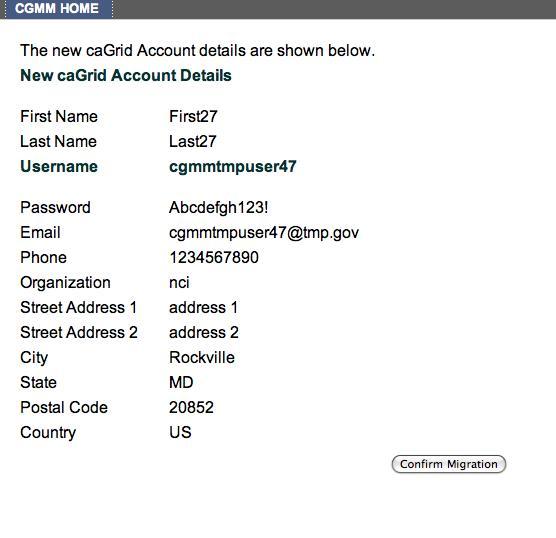


Figure 6.7 New caGrid Account page.

When the account is created, the CGMM asks the user to confirm the migration of the CSM account to this newly created caGrid account.

Once the user clicks the ‘Confirm Migration’ button, the CGMM proceeds to migrate the CSM account with the caGrid account. If the migration is successful, the CGMM tool shows the migration complete/success page as shown in Figure 6.5 .

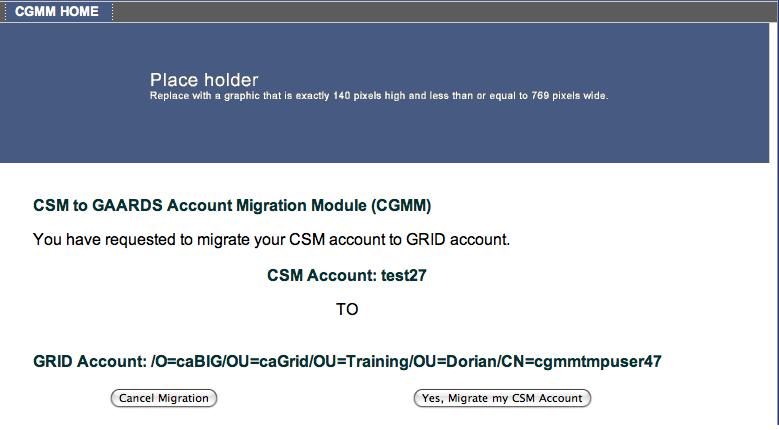


Figure 6.8 Migration complete page.

# Workflow 2: User logs in with caGrid account

In this scenario, the User has the caGrid account. The user logs in by providing the username, password and selecting the Authentication Source from the drop-down menu. The User then clicks the Login Button.

If the Login Id or Password is invalid, the CGMM tool shows an error.

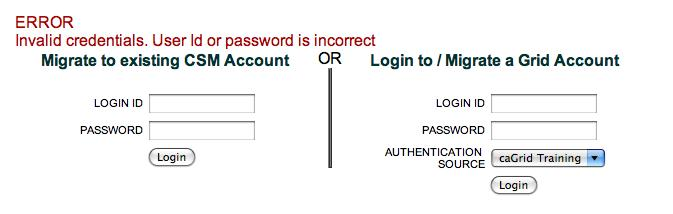


Figure 6.9 CGMM – caGrid Login error.

# Workflow 2-a: User is already migrated

The CGMM tool validates the user’s caGrid Login ID and password.. The CGMM Tool also verifies whether the caGrid User ID exists as a migrated user in the CSM Schema of the host application. If the user is already migrated, then the CGMM Tool populates the HTTP Request with user’s details, Grid Proxy and forwards the request to the host applications User Home page as shown in the Figure 6.6.

# Workflow 2-b: User has CSM account

If the Login Id and password are valid, the CGMM tool takes the user to GAARDS to CSM Account Migration page. In this page, the tool allows the user to login using existing CSM account or create a new CSM account.

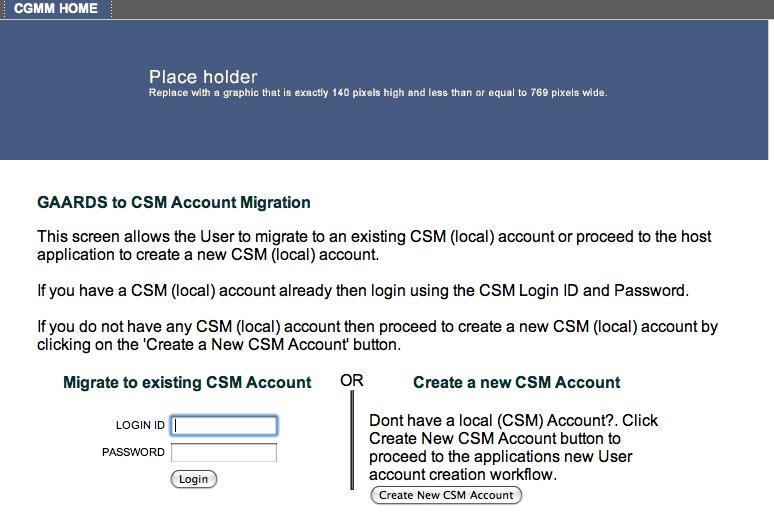


Figure 6.10 the caGrid Login success page / CSM Login Page.

In this case the user has an existing CSM account. Hence the user can proceed to migrate CSM account by providing the Login ID, Password and

**User logins in with CSM Login ID and password**

Once the user clicks Login button, the CGMM Tool validates the CSM account credentials provided by the user. If the credentials are valid then the CGMM Tool will show the ‘Confirm Migration’ screen to the user.



Figure 6.11 CSM to GAARDS Account migration page.

Once the user clicks the ‘Confirm Migration’ button, the CGMM proceeds to migrate the CSM account with the caGrid account. If the migration is successful, the CGMM tool shows the migration complete/success page as shown in Figure 6.5.

The User clicks on the ‘Log in to <<host application name>>’ button and is forwarded by the CGMM to the Host application User Home page. Figure 6.12 shows the User Home page for the ‘HostWeb’ web application, available as a reference implementation.



Figure 6.12 Migration complete page/ Host application User Home page.

# Workflow 2-c: User doesn’t have a CSM account

In this scenario, the User has already logged in with the caGrid account and user does not have a CSM account. Hence as shown in Figure 6.10, the User is left with the option to request creating a new CSM account for the host application.

If the User clicks the on the ‘Create New CSM Account’ button, the CGMM tool populates the HTTP request with caGrid User account, users Grid Proxy and forwards the request to the Host application to relieve control. The CGMM tool forwards the request to host applications New CSM User creation page. The CGMM obtains the context and URL for this page from the CGMM properties configuration file.

# Alternate Behavior

The Alternate behavior is the new feature of CGMM Web application. The alternate behavior is meant for existing web applications that would like to utilize CGMM Web application for the migration only. The alternate behavior assumes the host application does Authentication and New caGrid User Creation by itself. CGMM Web application notifies via Email about the new caGrid User Creation request sent by user. This behavior assumes the host application does not use Servlet Filter (CGMFilter) to intercept or interpret logged in/migrated users forwarded by the CGMM Web application. Hence in this behavior, CGMM Web redirects users to configured host application home/login page URLs.

# Workflows

The CGMM Tool’s alternate behavior allows multiple scenarios/workflows based on the User. The user must have Local (CSM) account. The user may or may not have a caGrid Account. Hence based on that there are four different scenarios addressed by the CGMM Tool. The scenarios are as following:

1. User logs in with CSM account and User does not have a caGrid account.
2. User logs in with CSM account and User has a caGrid account

**Note**: The CGMM tool DO NOT addresses the scenario where a User does not have a CSM (local) Account and does not have a caGrid account. The host application needs to address this scenario.

Let us look at the user interface workflow of the CGMM by going through each of the scenarios mentioned above. Shown below is the CGMM Tool Home page

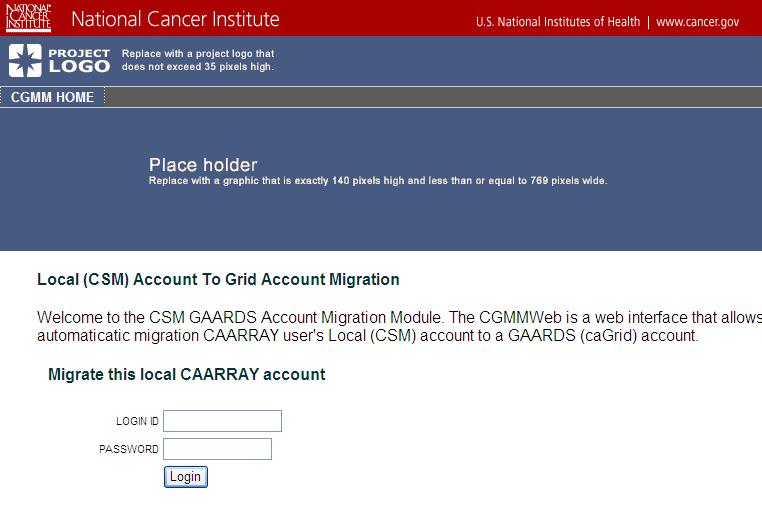


Figure 6.13 CGMM Home page (alternate behavior).

# Workflow

In this scenario, the User has the CSM account. The user logs in by providing the username and password and clicking on the Login button.

If the Login Id and password are valid, the CGMM tool takes the user to CSM to GAARDS Account Migration page. In this page, the tool allows the user to either login using existing caGrid account or create a new caGrid account.

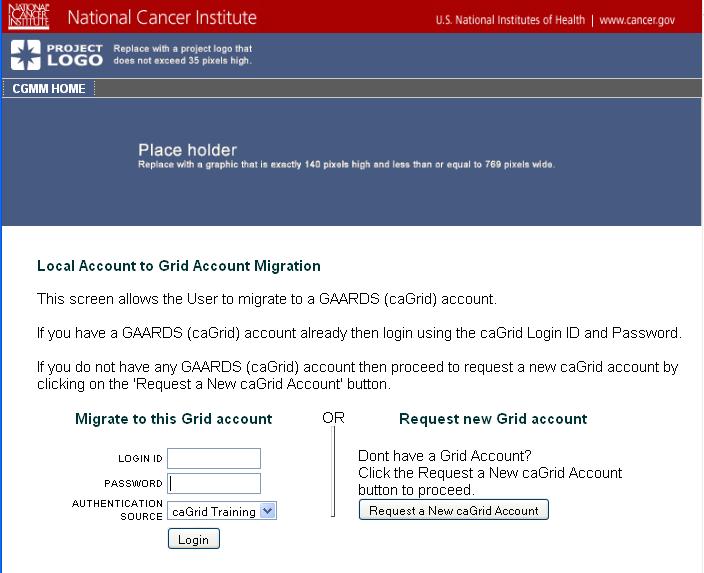


Figure 6.14 The CSM Login success page / Grid Login Page.

# Workflow 1-a: User has caGrid account

In this case the user has an existing caGrid account. Hence the user can proceed to migrate caGrid account by providing the Login ID, Password and also selecting the Authentication Source (Authentication Service).

**User logins in with caGrid Login ID and password**

Once the user clicks Login button, the CGMM Tool validates the caGrid account credentials provided by the user. If the credentials are valid then the CGMM Tool will show the ‘Confirm Migration’ screen to the user.

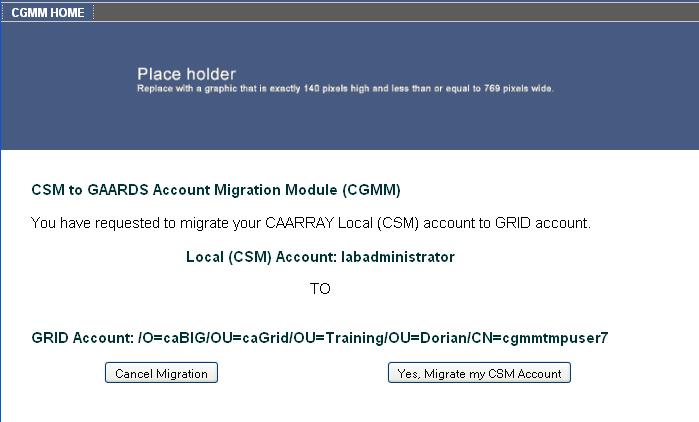


Figure 6.15 CSM to GAARDS Account migration page.

**User chooses to migrate his account**

On the migration confirmation page, the user has the option to cancel the migration or confirm it. Once the user proceeds to migrate by clicking on the ‘Yes, Migrate my CSM Account’ button, the CGMM will migrate the CSM account to caGrid account in the CSM Schema of the host application. CGMM will also mark the user as migrated. Once the migration process is complete, the CGMM Tool takes the user to the migration confirmation page. The user now has the only option of Log in to the host application.

When the user clicks the ‘Log in to <<Host Application Name>>’ button, the CGMM redirects the user to host application login page.



Figure 6.16 Migration complete page.

# Workflow 1-b: User does not have a caGrid account

In this case the user does not have an existing caGrid account. Hence the user can proceed to request a new account. As shown in Figure 6.14, the User clicks on the ‘Request a New caGrid Account’ button.

The User must provide all the information requested to proceed.

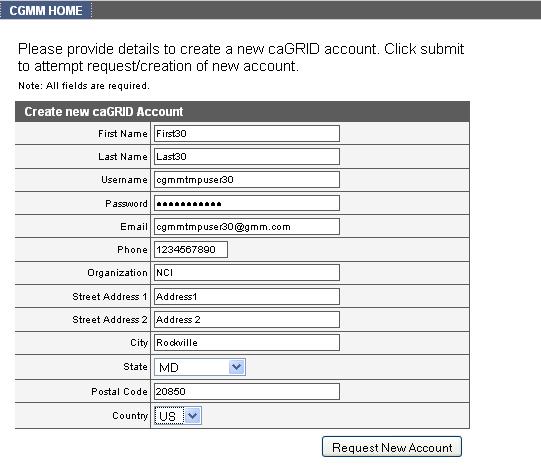


Figure 6.17 New caGrid Account Request page.

The user provides all information and clicks on the ‘Request New Account’ button to attempt to send an Email to the configured Administrator of the Host application.

The CGMM attempts to send an email to the ‘To’ Email ID provided in the CGMM configuration file. The configuration file also has the JNDI Name for the mail service, the ‘from’ Email id and the subject of the email. The email body consists of the information provided by the User in the above form.

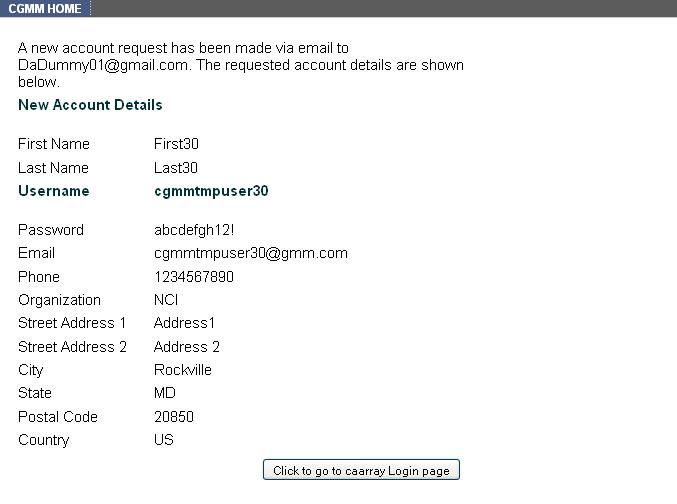


Figure 6.18 New caGrid Request submitted via email.

When the account is requested via email successfully, then CGMM shows the details to the user.

Once the user clicks the ‘Click to go to <<HostApplicationName>> Login page’ button, the CGMM proceeds to redirect the user to the host application login page.

The email sent to the host application administrator is shown below

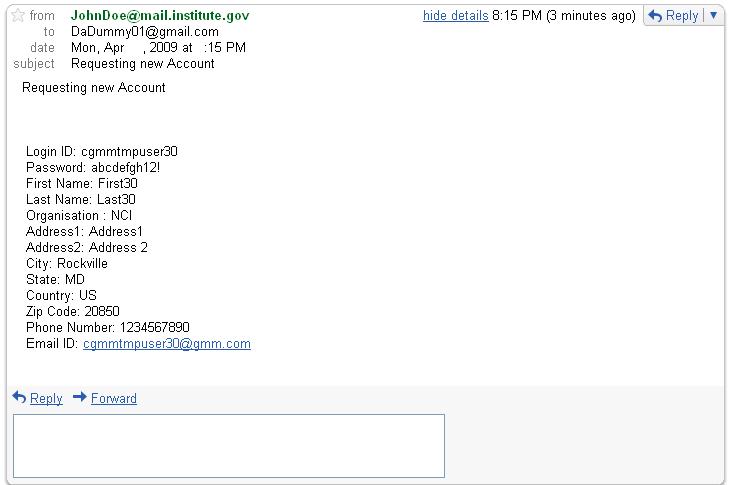


Figure 6.19 The email sent to the host application administrator.

# CGMM Tool configuration

CGMM Tool is designed to be customizable to allow host applications to implement the workflows however they decide to do so. The following are the customizations and configurations allowed for the CGMM tool

1. Configurable Look and Feel

The new caGrid User creation feature can be enabled or disabled based on the host application. This is achieved by configuring the cgmm-information section of cgmm-properties.xml with following:

* 1. Set the <cgmm-new-grid-user-creation-disabled> element to true
  2. Set the <cgmm-new-grid-user-creation-host-redirect-uri> element with host application context relative URI.

1. CGMM Information

The CGMM information configuration allows the following

* 1. Changing the CGMM tool’s context name.
  2. Enable/disable Auto Start SyncGTS Servlet
  3. Change the name of the cgmm.login.config file.
  4. Enable disable new caGrid User feature
  5. If disabled, the host application new caGrid user page URL.
  6. Enable/Disable Alternate Behavior of CGMM Web application.
  7. Enable/Disable Standalone Mode of CGMM Web application.

1. Configurable CaGrid Identity Providers for Authentication

The list of caGrid Identity providers is configurable via the cgmm-properties.xml

1. Host Information

The Host information customization allows the following

* 1. Configurable Host application web context name.
  2. Configurable name of the Host application.
  3. Configurable host applications Home page URL.
  4. Configurable host applications User Home Page URL.
  5. Configurable host applications User Login Page URL (for alternate behavior only).
  6. Configurable host applications new CSM user page URL.
  7. Configurable host applications Mail Service JNDI Name (for alternate behavior only).
  8. Configurable host applications Mail ‘To’ Email ID (for alternate behavior only).
  9. Configurable host applications Mail ‘From’ Email ID (for alternate behavior only).
  10. Configurable host applications Mail Subject text (for alternate behavior only).
  11. Configurable host applications Logo URL (for alternate behavior only).
  12. Configurable host application Logo Alt Text (for alternate behavior only).

1. Authentication Service/ Dorian information

The Authentication Service list allows specifying one or more Authentication Services to use for authentication purpose. The Dorian information, for each Authentication Service, can be used to create accounts, etc.

1. SyncGTS configuration

The sync-description.xml configuration file allows specifying the GTS Service URI, Trusted Authority filters, Excluded CA’s etc.

# Integrating CGMM with Container Managed Security

This section provides details regarding integration of CGMM API with applications existing container managed form-based security. This feature is available for those applications that utilize Form-based security in JBoss/Tomcat and the application would like to integrate CGMM API in their existing authentication workflow. This feature is made available as an alternative and for regular web applications utilizing non-container based JAAS authentication can use CGMM Web features.

# Details

For any web application that utilizes container managed security Form-based authentication, the integration of CGMM API to authenticate caGrid credentials requires modification to existing JBoss/Tomcat. To perform caGrid Authentication, the user input required is a) Login Name b) Password and c) caGrid Authentcation Source. The default Form Authenticator available (from Tomcat) allows two input parameters only. The caGrid Authenticator on the other hand requires three parameters instead of two. Hence a Custom Form Authenticator has been made available in CGMM APIs.

The JBoss application server recognizes only 5 types of Authenticators and FormAuthenticator is one of them. However there is no configurable alternative to specify a custom form authenticator.

The summarized steps for JBoss application server, to achieve the CGMM integration with custom Form-based container managed security is

1. The existing Web Application must utilize the custom Form Authenticator (CaGridFormAuthenticator) instead of the current FormAuthenticator. That is to use the custom auth-method ‘CAGRIDFORM’ instead of the default ‘FORM’ auth method in their security domain specified in the web-xml.
2. The catalina.jar in JBOSS\_HOME/server/default/deploy/jbossweb-tomcat55.sar folder should be modified
   1. In org/apache/catalina/startup/Authenticator.properties, add property CAGRIDFORM= gov.nih.nci.security.cgmm.authenticators.CaGridFormAuthenticator.
   2. In org/apache/catalina/authenticators/mbeans-descriptors.xml, add mbean ‘CaGridFormAuthenticator’ with type ‘gov.nih.nci.security.cgmm.authenticators.CaGridFormAuthenticator’

# Integration Steps

The overall steps required to integrated CGMM with an existing application using Form-based container managed security is as following. Appendix E: details the steps for a reference implementation ‘formsecurity.war’ application. Appendix F: provides steps for caArray – CGMM Container Managed Security Integration.

# CGMM Installation and Deployment

This section will provide details regarding the contents of the CGMM release. After the release content details, the minimum environment requirements are presented. In order for the CGMM Tool to function properly the environment setup mentioned in the installation prerequisites has to be made available.



Figure 7.0 Deployment Diagram

# Release Contents

The CGMM is released as a CGMM API Jar file and as a compressed web application in the form of a WAR (Web Archive) File. Along with the JAR and WAR files, the release includes sample configuration files that help developers configure the CGMM with their application(s). The CGMM Filter jar file is also made available

The CGMM Release contents can be found in theCGMM.zip file found on the NCICB Gforge website in the Security projects File Tab (<https://gforge.nci.nih.gov/frs/?group_id=12> ). The CGMM Release contents include the files in Table 7.1

| File | Description |
| --- | --- |
| cgmmweb.war | The CGMM Tool WAR file. |
| Cgmmapi.jar | The CGMM API Jar file. |
| Cgmmfilter.jar | The CGMM Filter jar file. |
| Cgmm-properties.xml | The CGMM properties configuration file. |
| ApplicationSecurityConfig.xml | The CSM Security Configuration file for various applications. For CGMM this file names and points the hibernate configuration file that will be used by the CGMMManager of CGMM for obtaining CSM AuthenticationManager / AuthorizationManager. |
| Cgmmweb.hibernate.cfg.xml | This is the hibernate configuration file pointed out by the ApplicationSecurityConfig.xml file for CSM. It is used to specify the Database connection properties or the Data Source name to be used for the Host Application Name. |
| cgmm.login.config | The login.config file to be used for obtaining the LoginModule for the Host application. The login.config file should be used to configure the login configuration for the Host application name. |
| sync-description.xml | The configuration file used by the SyncGTS servlet to sync the caGrid Trust fabric. This is required for caGrid Authentication purposes. |

Table 7.1 CGMM release contents

# Installation Pre-requisites

The following installation pre-requisites have to be done before CGMM Tool can be installed.

**Refactoring Host Application (Default Behavior)**

The Host application has to implement the following:

1. Add CGMM Filter to intercept all User requests

<filter>

<filter-name>CGMigrationFilter</filter-name>

<filter-class>

gov.nih.nci.security.cgmm.filters.CGMigrationFilter

</filter-class>

<init-param>

<param-name>CGMM\_APPLICATION\_CONTEXT</param-name>

<param-value>cgmmweb</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>CGMigrationFilter</filter-name>

<url-pattern>/secured/\*</url-pattern>

</filter-mapping>

Table 7.2 Web.xml configuration to add CGMM Filter

1. Identify the cgmm-properties.xml configuration details for Host information section.

The sample configuration is shown here and in Appendix B. Refer the cgmm-properties.xsd for more details about each configuration element.

Table 7.3 Sample configuration for host information

<host-application-information>

<host-context-name>cgmmhostweb</host-context-name>

<host-application-name-for-csm>sampleHostApplication</host-application-name-for-csm>

<host-public-home-page-url>/public/publicHome.jsp</host-public-home-page-url>

<host-user-home-page-url>/secured/userHomePage.jsp</host-user-home-page-url>

<host-new-local-user-creation-url> /public/newLocalUserCreation.jsp

</host-new-local-user-creation-url>

</host-application-information>

**Configure Container Managed Security (Alternate Behavior)**

The Host application has to implement the following:

1. Add Custom Form based Authentication configuration to web.xml

<security-constraint>

<web-resource-collection>

<web-resource-name>All resources</web-resource-name>

<description>Protects all resources</description>

<url-pattern>/protected/\*</url-pattern>

<http-method>GET</http-method>

<http-method>POST</http-method>

</web-resource-collection>

<auth-constraint>

<role-name>WebAppUser</role-name>

</auth-constraint>

</security-constraint>

<security-role>

<role-name>WebAppUser</role-name>

</security-role>

<login-config>

<auth-method>CAGRIDFORM</auth-method>

<realm-name>my-web</realm-name>

<form-login-config>

<form-login-page>/login.jsp</form-login-page>

<form-error-page>/error.html</form-error-page>

</form-login-config>

</login-config>

Table 7.4 Web.xml configuration to Custom Form based Authentication

1. Identify the cgmm-properties.xml configuration details for Host information section.

The sample configuration is shown in Appendix B. Refer the cgmm-properties.xsd for more details about each configuration element.

1. Add Mail Service configuration details for Request New User feature via email. For example:

<mbean code="org.jboss.mail.MailService" name="jboss:service=Mail">

<attribute name="JNDIName">java:/Mail</attribute>

<attribute name="User"><<user name>> </attribute>

<attribute name="Password"><<password>></attribute>

<attribute name="Configuration">

<configuration>

<property name="mail.transport.protocol" value="smtp"/>

<property name="mail.smtp.host" value="mailfwd.nih.gov"/>

<!-- <property name="mail.smtp.port" value="465"/>-->

<property name="mail.smtp.auth" value="false"/>

<property name="mail.smtp.starttls.enable" value="false"/>

<property name="mail.debug" value="false"/>

</configuration> </attribute> </mbean>

**caGrid Security infrastructure**

1. Identify the Authentication Service(s) that will be used for authenticationg caGrid users.
2. Identify the Dorian service that will be used to obtain grid proxy, create new caGrid user accounts etc.
3. Identify the sync-description.xml configuration information.

Refer the sample configuration file in Appendix C.

1. Identify the cgmm-properties.xml configuration details for Authentication Service and Dorian Service information.

The sample configuration is shown here. Refer the cgmm-properties.xsd for more details about each configuration element.

<authentication-service-list>

<authentication-service-information>

<service-name>caGrid Training</service-name>

<service-url> <https://dorian>.training.cagrid.org:8443/wsrf/services/cagrid/Dorian</service-url>

<dorian-information>

<service-url><https://dorian>.training.cagrid.org:8443/wsrf/services/cagrid/Dorian</service-url>

<proxy-lifetime-hours>12</proxy-lifetime-hours>

<proxy-lifetime-minutes>0</proxy-lifetime-minutes>

<proxy-lifetime-seconds>0</proxy-lifetime-seconds>

<proxy-delegation-path-length>3</proxy-delegation-path-length>

</dorian-information>

</authentication-service-information>

</authentication-service-list>

Table 7.4 Sample configuration for Authentication Service and Dorian information

**Identify Configuration parameters for CGMM**

1. Determine if the new caGrid User creation feature of the CGMM Tool is desired.
2. If the new caGrid user creation feature has to be disabled, then configure the cgmm-information section of cgmm-properties.xml with following:
   1. Set the <cgmm-new-grid-user-creation-disabled> element to true
   2. Set the <cgmm-new-grid-user-creation-host-redirect-uri> element with host application context relative URI.
3. If the Alternate behavior is enabled or set to true, then configure the host-information section of cgmm-properties.xml with the following:
   1. Set the <host-user-login-page-url> with host application login page context relative URL.
   2. Set the <host-mail-jndi-name> with JNDI name of the Jboss Mail Service.
   3. Set the <host-mail-email-id-to> with the ‘To’ Email ID.
   4. Set the <host-mail-email-id-from> with the ‘From’ Email ID.
   5. Set the <host-mail-email-subject> with the Email Subject text.
   6. Set the <host-application-logo-url> with the URL for application logo.
   7. Set the <host-application-logo-alt-text> with the Alt Text for the application logo.

# Deployment Checklist

Before deploying the CGMM, verify the following environment and configuration conditions are met. This software and access credentials/parameters are required.

* Host application Environment
  + Jboss 4.0 Application Server
  + MySQL 4.0 OR Oracle 9i Database Server (with an account that can create databases)
  + Host Application utilizing the CGMM Filter (optional in Standalone Mode).
  + CSM v4.1 Schema with existing Users.
* CGMM Release Components
  + CGMM Properties configuration file
  + Sync Description configuration file
  + ApplicationSecurityConfig.xml Security configuration for CGMM
  + JAAS Login Module Configuration for ‘sampleHostApplication’ Application.
  + Mail service configuration for alternate behavior.
* caGrid Environment
  + caGrid 1.2 software is installed
  + Dorian Service is available for creation of new Grid User accounts.
  + Authentication Service(s) available to authenticate Grid users.
  + SyncGTS to sync with Trust Fabric.
  + Host Certificate is available for the Server hosting the application server.

# Deployment Steps

**Verify the installation prerequisites are completed. Verify the deployment checklist is complete.**

**Step 1: Deploy cgmmweb.war file**

Copy the cgmmweb.war in the deployment directory of Jboss which can be found at {jboss-home}/server/default/deploy/

**Step 2: Deploy Host Application with CGMM Filter (optional in Alternate Behavior AND/OR Standalone Mode AND/OR Container Managed Security Integration)**

Copy the host application’s WAR file in the deployment directory of Jboss which can be found at {jboss-home}/server/default/deploy/

**Step 3: Configure System properties**

Set the System properties for the configuration files.

In Jboss, modify the Jboss\_home/server/default/deploy/properties-service.xml.

<attribute name="Properties">

gov.nih.nci.security.cgmm.syncgts.file =

<<path to>>/sync-description.xml

gov.nih.nci.security.cgmm.properties.file =

<<path to>>/cgmm-properties.xml

gov.nih.nci.security.configFile =

<<path to>>/ApplicationSecurityConfig.xml

gov.nih.nci.security.cgmm.login.config.file =

<<path to>>/cgmm.login.config

</attribute>

Table 7.5 Sample configuration for System properties in JBoss.

**Step 4: Configure SyncGTS**

* + Configure the URLs for Slave/Master GTS.
  + Refer the sample from Appendix C.

**Step 5: Configure CGMM Properties file**

* + For description of the elements see the cgmm-properties.xsd.
  + Example

<host-application-name-for-csm>sampleHostApplicationContextName</host-application-name-for-csm>

**Step 6: Configure CSM Application Security Configuration file**

* Configure ApplicationSecurityConfig.xml
  + Modify the ‘context-name’ to Host application context name. Example

<context-name>sampleHostApplicationContextName </context-name>

* + Modify the ‘hibernate-config-file’ Element to point to the hibernate configuration file. Example

<hibernate-config-file>/<<path to>>/cgmmweb.hibernate.cfg.xml</hibernate-config-file>

* Configure <<hostApplicationName>>.hibernate.cfg.xml
  + Configure the Database Connection Properties or DataSource for the application.

**Step 7: Configure the Jboss JAAS Login parameters**

In order to configure the CGMM to authenticate CSM users, create an entry in the login-config.xml of Jboss as shown in Figure 7.6. This entry configures a login-module against the host application context. The location of this file is {jboss-home}/server/default/conf/login-config.xml.

<application-policy name = "sampleHostApplication">

<authentication>

<login-module

code="gov.nih.nci.security.authentication.loginmodules.RDBMSLoginModule"

flag = "sufficient">

<module-option name="driver"><<Database Driver>></module-option>

<module-option name="url"><<Database URL>></module-option>

<module-option name="user"><<DB Username>></module-option>

<module-option name="passwd"><<DB Password>></module-option>

<module-option name="query">

SELECT \* FROM csm\_user WHERE login\_name=? and password=?</module-option>

<module-option name="encryption-enabled">YES</module-option>

</login-module></authentication>

</application-policy>

Figure 7.6 Sample login-config.xml entry for the host application

Alternatively, the JAAS configuration can be done via the cgmm.login.config configuration file.

* + Rename the cgmm.login.config file to value specified System property ‘gov.nih.nci.security.cgmm.login.config.file’
  + Modify the login.config name to the Host application Name.
  + Point to Host application Schema (CSM 4.1 Schema of the Host application)

**Step 8: Configure Jboss Mail Service (Only in case of Alternate Behavior AND/OR Standalone Mode)**

To configure JBoss Mail Service, add the following to the JBOSS\_HOME/server/default/deploy/mail—service.xml configuration. Sample configuration is shown in following figure.

<mbean code="org.jboss.mail.MailService" name="jboss:service=Mail">

<attribute name="JNDIName">java:/Mail</attribute>

<attribute name="User"><<user name>> </attribute>

<attribute name="Password"><<password>></attribute>

<attribute name="Configuration">

<configuration>

<property name="mail.transport.protocol" value="smtp"/>

<property name="mail.smtp.host" value="mailfwd.nih.gov"/>

<!-- <property name="mail.smtp.port" value="465"/>-->

<property name="mail.smtp.auth" value="false"/>

<property name="mail.smtp.starttls.enable" value="false"/>

<property name="mail.debug" value="false"/>

</configuration> </attribute> </mbean>

**Step 9: Configure CLM Audit Logging**

To enable audit logging, add the following Log4j appender and category to log4j.xml. Replace values for Application Name, Server Name, Port, Schema Name, DB User, Password information.

<appender name="CLM\_APPENDER" class="gov.nih.nci.logging.api.appender.jdbc.JDBCAppender"> <param name="application" value="<<APPLICATION\_NAME>>" /> <param name="maxBufferSize" value="1" /> <param name="dbDriverClass" value="org.gjt.mm.mysql.Driver" /> <param name="dbUrl" value="jdbc:mysql://<<SERVER\_NAME>>:<<PORT>>/<<CLM\_SCHEMA\_NAME>>" /> <param name="dbUser" value="<<DB\_USER>>" /> <param name="dbPwd" value="<<PASSWORD>>" /> <param name="useFilter" value="true" /> <layout class="org.apache.log4j.PatternLayout"> <param name="ConversionPattern" value=":: [%d{ISO8601}] %-5p %c{1}.%M() %x - %m%n" /> </layout>

</appender>

<category name=" CGMM.Audit.Logging">

<level value="info" /> <appender-ref ref="CLM\_APPENDER" />

</category>

**Step 10: Configure Log4j.xml**

To turn off the unnecessary log entries on the console, add the following to the log4j.xml configuration

<category name="COM.claymoresystems.ptls.SSLDebug">

<priority value="OFF" />

</category>

**Step 11: Start Jboss**

1. Once the deployment and configuration is completed, start JBoss. Check the logs to confirm there are no errors while the CGMM Web application and host application are deployed on the server.

2. Once the Jboss server has completed deployment, open a browser to access the host applications secured login page. The URL will be http://<<jboss-server>>/<<host\_application\_context>>, where the <<jboss-server>> is the IP or the DNS name of Jboss Server and <<host\_application\_context>> is the context name of the host application.

3. The Host application should forward the control to CGMM tool’s login screen.

**Note:** In case of any errors, follow a debugging and troubleshooting procedure to diagnose and solve the issues. For more information refer the CSM Wiki – CGMM FAQ page.

# Appendix A: CGMM Properties XSD

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">

<xs:element name="authentication-service-information">

<xs:annotation>

<xs:documentation>

This Element allows specifying required Authentication

Service Information. Please refer the caGrid Wiki for

more details regarding Authentication Service.

</xs:documentation>

</xs:annotation>

<xs:complexType>

<xs:sequence>

<xs:element ref="service-name"/>

<xs:element ref="service-url"/>

<xs:element ref="dorian-information"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="authentication-service-list">

<xs:annotation>

<xs:documentation>

This element allows specifying a list of Authentication

Services.

</xs:documentation>

</xs:annotation>

<xs:complexType>

<xs:sequence>

<xs:element ref="authentication-service-information" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="dorian-information">

<xs:annotation>

<xs:documentation>

This element allows specification of caGrid Dorian

related information. Please refer the caGrid Wiki for

more details regarding Dorian.

</xs:documentation>

</xs:annotation>

<xs:complexType>

<xs:sequence>

<xs:element ref="service-url"/>

<xs:element ref="proxy-lifetime-hours"/>

<xs:element ref="proxy-lifetime-minutes"/>

<xs:element ref="proxy-lifetime-seconds"/>

<xs:element ref="proxy-delegation-path-length"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="cgmm-information">

<xs:annotation>

<xs:documentation>

This element allows specification of CGMM related

information.

</xs:documentation>

</xs:annotation>

<xs:complexType>

<xs:sequence>

<xs:element ref="cgmm-context-name"/>

<xs:element ref="cgmm-login-config-file-name"/>

<xs:element ref="start-auto-syncgts"/>

<xs:element ref="cgmm-new-grid-user-creation-disabled"/>

<xs:element ref="cgmm-new-grid-user-creation-host-redirect-uri"/>

<xs:element ref="cgmm-alternate-behavior"/>

<xs:element ref="cgmm-standalone-mode"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="host-application-information">

<xs:annotation>

<xs:documentation>

This element allows specification of Host Application

related information.

</xs:documentation>

</xs:annotation>

<xs:complexType>

<xs:sequence>

<xs:element ref="host-context-name"/>

<xs:element ref="host-application-name-for-csm"/>

<xs:element ref="host-public-home-page-url"/>

<xs:element ref="host-user-home-page-url"/>

<xs:element ref="host-user-login-page-url"/>

<xs:element ref="host-new-local-user-creation-url"/>

<xs:element ref="host-mail-jndi-name" minOccurs="0" maxOccurs="1"/>

<xs:element ref="host-mail-email-id-to" minOccurs="0" maxOccurs="1"/>

<xs:element ref="host-mail-email-id-from" minOccurs="0" maxOccurs="1"/>

<xs:element ref="host-mail-email-subject" minOccurs="0" maxOccurs="1"/>

<xs:element ref="host-application-logo-url" minOccurs="0" maxOccurs="1"/>

<xs:element ref="host-application-logo-alt-text" minOccurs="0" maxOccurs="1"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="cgmm-new-grid-user-creation-disabled" type="xs:string">

<xs:annotation>

<xs:documentation>

This element indicates if the New Grid User Creation

workflow is disabled for this installation of CGMM. A

value of true indicates the particular workflow is

disabled. If disabled the

cgmm-new-grid-user-creation-host-redirect-url is

ignored. The value of false indicates that the workflow

is not disabled. The

cgmm-new-grid-user-creation-host-redirect-url is

expected to have valid content.

</xs:documentation>

</xs:annotation></xs:element>

<xs:element name="cgmm-new-grid-user-creation-host-redirect-uri" type="xs:string" nillable="true">

<xs:annotation>

<xs:documentation>

This element allows specifying the Hosts Redirect URL

once the New Grid User creation workflow is successfully

completed. If this workflow is disabled, then the this

element is ignored.

</xs:documentation>

</xs:annotation></xs:element>

<xs:element name="cgmm-alternate-behavior" type="xs:string">

<xs:annotation>

<xs:documentation>

This element allows specifying the CGMM Alternate Behavior. If value is set to 'true' then CGMM will redirect requests to Host application. If value is set to 'false' then CGMM will forward requests with User related parameters.

</xs:documentation>

</xs:annotation></xs:element>

<xs:element name="cgmm-standalone-mode" type="xs:string">

<xs:annotation>

<xs:documentation>

This element allows specifying the Stand Alone Mode for CGMM. In Stand Alone Mode the CGMM will not redirect or forward to the host application. Post Migration it will not provide any option to continue to the Host application pages.

</xs:documentation>

</xs:annotation></xs:element>

<xs:element name="cgmm-context-name" type="xs:string">

<xs:annotation>

<xs:documentation>

The Web application context name of CGMM Web Application. The default value is cgmmweb

</xs:documentation>

</xs:annotation></xs:element>

<xs:element name="cgmm-login-config-file-name" type="xs:string">

<xs:annotation>

<xs:documentation>

The JAAS Login Config file name. This file consists the

CSM Authentication configuration necessary for

authentication of CSM users. If the

java.security.auth.login.config JAAS property is set in

SystemProperties then this element is ignored and the

Login Module Configuration for cgmmweb is obtained from

the particular Login Configuration.

</xs:documentation>

</xs:annotation></xs:element>

<xs:element name="host-context-name" type="xs:string">

<xs:annotation>

<xs:documentation>

The Web Application Context name of the Host Web

Application.This string value must match the web context

name of the host application.

</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="host-application-name-for-csm" type="xs:string">

<xs:annotation>

<xs:documentation>

The Application Name of the Host Web Application that is to be used by CSM authentication and authorization. This string value must match the name of the host application available in the CSM Schema.

</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="host-public-home-page-url" type="xs:string"/>

<xs:element name="host-user-home-page-url" type="xs:string">

<xs:annotation>

<xs:documentation>

This element allows specifying the URL for User Home Page of the Host application. If kept blank, this element indicates CGMMWeb to use Alternate Behavior.

</xs:documentation>

</xs:annotation></xs:element>

<xs:element name="host-user-login-page-url" type="xs:string">

<xs:annotation>

<xs:documentation>

This element can be left blank if 'host-user-home-page-url' is specified and hence Default Behavior is desired. However if Alternate Behavior is desired, specify this element with the Login Page URL of the Host Application.

</xs:documentation>

</xs:annotation></xs:element>

<xs:element name="host-new-local-user-creation-url" type="xs:string">

<xs:annotation>

<xs:documentation>

This element OPTIONAL allows specifying the URL for New

Local User creation workflow of the Host application.

</xs:documentation>

</xs:annotation></xs:element>

<xs:element name="host-mail-jndi-name" type="xs:string">

<xs:annotation>

<xs:documentation>

This element OPTIONAL allows specifying the JNDI Name for the JBoss Mail Service setup.

</xs:documentation>

</xs:annotation></xs:element>

<xs:element name="host-mail-email-id-to" type="xs:string">

<xs:annotation>

<xs:documentation>

This element OPTIONAL allows specifying the 'To' Email Address

for emails sent by CGMM to request new accounts.

</xs:documentation>

</xs:annotation></xs:element>

<xs:element name="host-mail-email-id-from" type="xs:string">

<xs:annotation>

<xs:documentation>

This element OPTIONAL allows specifying the 'From' Email Address for emails sent by CGMM to request new accounts.

</xs:documentation>

</xs:annotation></xs:element>

<xs:element name="host-mail-email-subject" type="xs:string">

<xs:annotation>

<xs:documentation>

This element OPTIONAL allows specifying the Subject of the emails sent by CGMM to request new accounts.

</xs:documentation>

</xs:annotation></xs:element>

<xs:element name="host-application-logo-url" type="xs:string">

<xs:annotation>

<xs:documentation>

This element OPTIONAL allows specifying URL for the Application Header Logo.

</xs:documentation>

</xs:annotation></xs:element>

<xs:element name="host-application-logo-alt-text" type="xs:string">

<xs:annotation>

<xs:documentation>

This element OPTIONAL allows specifying Alt Text for the Application Header Logo.

</xs:documentation>

</xs:annotation></xs:element>

<xs:element name="start-auto-syncgts" type="xs:string"/>

<xs:element name="service-name" type="xs:string"/>

<xs:element name="service-url" type="xs:anyURI"/>

<xs:element name="proxy-lifetime-hours" type="xs:integer"/>

<xs:element name="proxy-lifetime-minutes" type="xs:integer"/>

<xs:element name="proxy-lifetime-seconds" type="xs:integer"/>

<xs:element name="proxy-delegation-path-length" type="xs:integer"/>

<xs:element name="cgmm-properties">

<xs:annotation>

<xs:documentation>

The Root Element of the CGMM Properties. This element

allows specifying the CGMM information, Host Application

Information and Authentication Service/Dorian

Information.

</xs:documentation>

</xs:annotation>

<xs:complexType>

<xs:sequence>

<xs:element ref="cgmm-information"/>

<xs:element ref="host-application-information"/>

<xs:element ref="authentication-service-list"/>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

# Appendix B: Sample CGMM Properties File

  <?xml version="1.0" encoding="UTF-8" ?>

<cgmm-properties xmlns:xsi="**http://www.w3.org/2001/XMLSchema-instance**" xsi:noNamespaceSchemaLocation="**cgmm-properties.xsd**">

<cgmm-information>

  <cgmm-context-name>cgmmweb</cgmm-context-name>

  <cgmm-login-config-file-name>cgmm.login.config</cgmm-login-config-file-name>

  <start-auto-syncgts>false</start-auto-syncgts>

  <cgmm-new-grid-user-creation-disabled>false</cgmm-new-grid-user-creation-disabled>

  <cgmm-new-grid-user-creation-host-redirect-uri>/public/newGridUserCreation.jsp</cgmm-new-grid-user-creation-host-redirect-uri>

  <cgmm-alternate-behavior>true</cgmm-alternate-behavior>

  <cgmm-standalone-mode>false</cgmm-standalone-mode>

  </cgmm-information>

<host-application-information>

  <host-context-name>caarray</host-context-name>

  <host-application-name-for-csm>caarray</host-application-name-for-csm>

  <host-public-home-page-url>/home.action</host-public-home-page-url>

  <host-user-home-page-url>/protected/project/workspace.action</host-user-home-page-url>

  <host-user-login-page-url>/protected/project/workspace.action</host-user-login-page-url>

  <host-new-local-user-creation-url>/registration/input.action</host-new-local-user-creation-url>

  <host-mail-jndi-name>java:/Mail</host-mail-jndi-name>

  <host-mail-email-id-to>DaDummy01@gmail.com</host-mail-email-id-to>

  <host-mail-email-id-from>JohnDoe@mail.institute.gov</host-mail-email-id-from>

  <host-mail-email-subject>Requesting new Account</host-mail-email-subject>

  <host-application-logo-url>images/appLogo.gif</host-application-logo-url>

  <host-application-logo-alt-text>caArray Host Application</host-application-logo-alt-text>

  </host-application-information>

<authentication-service-list>

<authentication-service-information>

  <service-name>caGrid Training</service-name>

  <service-url>https://dorian.training.cagrid.org:8443/wsrf/services/cagrid/Dorian</service-url>

<dorian-information>

  <service-url>https://dorian.training.cagrid.org:8443/wsrf/services/cagrid/Dorian</service-url>

  <proxy-lifetime-hours>12</proxy-lifetime-hours>

  <proxy-lifetime-minutes>0</proxy-lifetime-minutes>

  <proxy-lifetime-seconds>0</proxy-lifetime-seconds>

  <proxy-delegation-path-length>3</proxy-delegation-path-length>

 </dorian-information>

</authentication-service-information>

  </authentication-service-list>

  </cgmm-properties>

# Appendix C: Sample Sync Description File

<ns1:SyncDescription xmlns:ns1=”http://cagrid.nci.nih.gov/12/SyncGTS” xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”>

<ns1:SyncDescriptor>

<ns1:gtsServiceURI><https://slavegts>.training.cagrid.org:8443/wsrf/services/cagrid/GTS</ns1:gtsServiceURI>

<ns1:Expiration hours=”12” minutes=”0” seconds=”0”/>

<ns1:TrustedAuthorityFilter xsi:type=”ns2:TrustedAuthorityFilter” mlns:ns2=”http://cagrid.nci.nih.gov/8/gts”>

<ns2:Lifetime xsi:type=”ns2:Lifetime”>Valid</ns2:Lifetime>

<ns2:Status xsi:type=”ns2:Status”>Trusted</ns2:Status>

</ns1:TrustedAuthorityFilter>

<ns1:PerformAuthorization>true</ns1:PerformAuthorization>

<ns1:GTSIdentity>/O=caBIG/OU=caGrid/OU=Training Trust Fabric/CN=host/slavegts.training.cagrid.org</ns1:GTSIdentity>

</ns1:SyncDescriptor>

<ns1:ExcludedCAs>

<ns1:CASubject>O=caBIG,OU=caGrid,OU=Training Trust Fabric,CN=caGrid Training Trust Fabric CA</ns1:CASubject>

</ns1:ExcludedCAs>

<ns1:DeleteInvalidFiles>false</ns1:DeleteInvalidFiles>

<ns1:CacheSize>

<ns1:year>0</ns1:year>

<ns1:month>1</ns1:month>

<ns1:day>0</ns1:day>

</ns1:CacheSize>

<ns1:NextSync>600</ns1:NextSync>

</ns1:SyncDescription>

# Appendix D: Sample steps to install CGMM with reference implementation

The steps mentioned in this Appendix D will work as mentioned given the steps are followed correctly. These steps will install the reference implementation cgmmHostWeb web application along with the cgmmweb web application. Using these steps, a test environment can be setup to demonstrate how the CGMM Tool works with an existing Host application. The internal details of CGMM Tool are beyond the scope of this document. Refer the CGMM Design document for more details.

**Note: The paths, values are sample used in commands and configuration files are for example only.**

1. Verify caGrid 1.2 is installed. The CAGRID\_HOME variable should be set.

If caGrid 1.2 is not installed then install caGrid 1.2 using the caGrid Installer 1.2 (install the software only, no services needed).

1. Verify ANT\_HOME, JAVA\_HOME, CAGRID\_HOME, GLOBUS\_LOCATION are set as environment variables. If the variables are not set then set the variables as mentioned in this step.

At command prompt, Type the following and press Enter Key after each statement.

ANT\_HOME=/usr/local/apache-ant-1.6.5

export ANT\_HOME;

PATH=$PATH:/usr/local/apache-ant-1.6.5/bin

export PATH;

JAVA\_HOME=/usr/jdk1.5.0\_10

export JAVA\_HOME;

GLOBUS\_LOCATION=/usr/local/ws-core-4.0.3

export GLOBUS\_LOCATION;

CAGRID\_HOME=/h1/username/<<path where caGrid Software was installed>>

export CAGRID\_HOME;

1. Verify caGrid 1.2 is configured to point to the Training Grid 1.2

At command prompt, Type the following and press Enter Key after each statement.

Cd $CAGRID\_HOME

ant –Dtarget.grid=training-1.2 configure

1. Run SyncGTS

At command prompt, Type the following and press Enter Key after each statement.

Cd $CAGRID\_HOME/projects/syncgts

ant syncWithTrustFabric

1. Obtain Host Certificate for the machine.

This is a pre requisite and the instructions for obtaining the Host Credentials ( certificate) is available in the this link

<http://www.cagrid.org/mwiki/index.php?title=Dorian:1.1:Administrators_Guide:Requesting_Host_Credentials>.

1. Deploy the cgmmHostWeb.war by putting the war file in jboss deploy folder
2. Deploy the cgmmweb.war by putting the war file in jboss deploy folder
3. Configure CGMM and Host Application properties
4. Configure System Properties.

Modify the JBOSS\_HOME/server/default/deploy/properties-service.xml and add the following properties

gov.nih.nci.security.cgmm.syncgts.file = /usr/local/jboss-4.0.5.GA/server/default/cgmm\_config/sync-description.xml

gov.nih.nci.security.cgmm.properties.file = /usr/local/jboss-4.0.5.GA/server/default/cgmm\_config/cgmm-properties.xml

gov.nih.nci.security.configFile = /usr/local/jboss-4.0.5.GA/server/default/cgmm\_config/ApplicationSecurityConfig.xml

gov.nih.nci.security.cgmm.login.config.file = /usr/local/jboss-4.0.5.GA/server/default/cgmm\_config/cgmm.login.config

1. Configure JAAS Login Configuration Module.
   * Rename the cgmm.login.config file to value specified System property ‘gov.nih.nci.security.cgmm.login.config.file’
   * Modify the name of the cgmm.login.config file to ‘sampleHostApplication.login.config’
   * Point to CSM 4.1 Schema for the sampleHostApplication
2. Configure Sync GTS description configuration xml file.

This is required to sync the caGrid Trust Fabric with the Servers Keystore. The instructions on how to configure the sync-description.xml is available in this link.

* + <http://www.cagrid.org/wiki/GTS:1.2:Administrators_Guide:SyncGTS:Configuration>
  + However, the sample sync-description.xml provided in the Appendix C points to the caGrid Training 1.2

1. Configure CGMM Properties file.
   * For description of the elements see the cgmm-properties.xsd in Appendix A.
   * Use Appendix B contents to configure the cgmm-properties.xml file.
2. Configure ApplicationSecurityConfig.xm l file.
   * Modify the ‘context-name’ to Host application context name. Example

<context-name>sampleHostApplication</context-name>

* + Modify the ‘hibernate-config-file’ Element to point to hibernate configuration file. Example

<hibernate-config-file>

/usr/local/jboss-4.0.5.GA/server/default/cgmm\_config/cgmmweb.hibernate.cfg.xml

</hibernate-config-file>

1. Configure the Database Connection Properties or DataSource for the application.
   * Specify the Database connection properties in cgmmweb.hibernate.cfg.xml

<property name=”connection.username”>root</property>

<property name = “connection.url”> jdbc:mysql://localhost:3306/csmauthschema\_4\_1

</property>

<property name=”dialect”>org.hibernate.dialect.MySQLDialect</property>

<property name=”connection.password”>root</property>

<property name=”connection.driver\_class”>org.gjt.mm.mysql.Driver</property>

* + Or configure datasource. The sample JBOSS\_HOME/server/default/deploy/mysql-ds.xml configuration

<local-tx-datasource>

<jndi-name>cgmmweb</jndi-name>

<connection-url>jdbc:mysql://localhost:3306/csm41</connection-url>

<driver-class>org.gjt.mm.mysql.Driver</driver-class>

<user-name>root</user-name>

<password>root</password>

</local-tx-datasource>

# Appendix E: Sample Software setup steps for the reference implementation to test Container Managed Security Integration with CGMM.

**Software Setup Steps:**

**Note: Sample files and formsecurity.war are available in the Release Contents/reference\_implementation folder.**

**Note: Refer Appendix D Steps 1- 5 and perform them before continuing the following steps.**

1. Copy to JBOSS\_HOME\server\default\deploy\jbossweb-tomcat55.sar folder the following jars
   1. CGMM\_RELEASE\_FOLDER/cgmmapi.jar
   2. CGMM\_RELEASE\_FOLDER/catalina.jar (please note this is custom catalina.jar)
   3. CGMM\_RELEASE\_FOLDER/jbossweb-tomcat55-sar-jars/\*.jar
2. Deploy the CGMM\_RELEASE\_FOLDER/reference\_implementation/formsecurity.war to JBOSS\_HOME/server/default/deploy folder.
3. Modify JBOSS\_HOME/server/default/deploy/mysql-ds.xml and add the following datasource entry.

  <local-tx-datasource>

    <jndi-name>formsecurity</jndi-name>

    <driver-class>org.gjt.mm.mysql.Driver</driver-class>

    <connection-url>jdbc:mysql://localhost:3306/cgmm\_container\_managed\_security</connection-url>

    <user-name><<USERNAME>></user-name>

    <password><<PASSWORD>></password>

  </local-tx-datasource>

1. Modify JBOSS\_HOME/server/default/conf/login-config.xml
   1. Add following configuration inside the <policy> element

<application-policy name = "my-web">

       <authentication>

<login-module code="gov.nih.nci.security.cgmm.loginmodules.NullPasswordStackingLoginModule" flag="optional">

                    <module-option name="password-stacking">useFirstPass</module-option>

                </login-module>

                <login-module code="org.jboss.security.auth.spi.DatabaseServerLoginModule" flag="required">

                <module-option name="password-stacking">useFirstPass</module-option>

                              <module-option name="dsJndiName">java:formsecurity</module-option>

                              <module-option name="rolesQuery">SELECT cg.group\_name, 'Roles' FROM csm\_group cg, csm\_user\_group cug, csm\_user cu WHERE cg.group\_id = cug.group\_id AND cug.user\_id = cu.user\_id AND cu.login\_name = ?</module-option>

                </login-module>

    </authentication>

</application-policy>

1. Configure CGMM
   1. Verify the following properties are set in JBOSS\_HOME/server/default/deploy/properties-service.xml. Please ensure to specify correct path for each.
      1. gov.nih.nci.security.cgmm.syncgts.file = PATH\_TO\_jboss-4.0.5.GA/server/default/cgmm\_config/sync-description.xml
      2. gov.nih.nci.security.cgmm.properties.file = PATH\_TO\_jboss-4.0.5.GA/server/default/cgmm\_config/cgmm-properties.xml
      3. gov.nih.nci.security.configFile = PATH\_TO\_jboss-4.0.5.GA/server/default/cgmm\_config/ApplicationSecurityConfig.xml
      4. gov.nih.nci.security.cgmm.login.config.file = PATH\_TO\_jboss-4.0.5.GA/server/default/cgmm\_config/cgmm.login.config
   2. Modify the database connection properties in cgmmweb.hibernate.cfg.xml
   3. Modify the ApplicationSecurityConfig.xml to point to correct application name (sample has ‘sampleHostApplicationName’, this name should match in the CSM Schema).
   4. Modify cgmm.login.config and verify connection properties and the Application Policy name (sample has ‘sampleHostApplicationName’, this name should match in the CSM Schema).
2. Database Setup : Create and prime CSM 4.1 Schema.
   1. Modify the (attached ) sample script and change the following
      1. Search and replace ‘cgmmtmpuser2’ with the caGrid Login ID of your choice. Ensure the ID used is the one used to authenticate against caGrid Training Authentication Source.
      2. Search for ‘root’ and replace it with your database user name for MySQL.
      3. Search for ‘H/2qIBdj9TQ=’ and replace with encrypted value of MySQL password of the database user.
   2. Execute the db script.
3. Configure JBoss Mail Service
   1. Modify the JBOSS\_HOME/server/default/deploy/mail-service.xml and add following entry with valid attribute values

<mbean code="org.jboss.mail.MailService" name="jboss:service=Mail">

<attribute name="JNDIName">java:/Mail</attribute>

<attribute name="User">sample\_user name </attribute>

<attribute name="Password">sample\_password</attribute>

<attribute name="Configuration">

<configuration>

<property name="mail.transport.protocol" value="smtp"/>

<property name="mail.smtp.host" value="Sample\_ mailfwd.nih.gov"/>

<!-- <property name="mail.smtp.port" value="465"/>-->

<property name="mail.smtp.auth" value="false"/>

<property name="mail.smtp.starttls.enable" value="false"/>

<property name="mail.debug" value="false"/>

</configuration> </attribute> </mbean>

1. (Optional) Configure CLM Audit Logging.
2. **Start Test:** Start JBoss
3. Access url <http://localhost:8080/formsecurity/protected/>
4. It will prompt for caGrid credentials. Enter valid caGrid credentials. A successful login indicates configurations and setup was done correctly

# Appendix F: Sample Software setup steps for caArray – CGMM Container Managed Security Integration (Alternate Behavior)

**Note: The paths, values are sample used in commands and configuration files are for example only.**

**Note: Refer Appendix D Steps 1- 5 and perform them before continuing the following steps.**

1. Check out caArray Trunk source folder.
2. Modify the following
   1. caarray.war/WEB-INF/pages/login.jsp and add the following drop down list in the login form.

Authentication Source:

<select name="authenticationServiceURL" size="1">

<option value="https://dorian.training.cagrid.org:8443/wsrf/services/cagrid/Dorian">caGrid Training</option>"

<%-- <% // Use the following code to auto populate the Drop down list.

if (request.getAttribute("AUTHENTICATION\_SOURCE\_MAP") == null) {

out.println("AUTHENTICATION\_SOURCE\_MAP attribute is not available .");

}

Map sm = (Map)request.getAttribute("AUTHENTICATION\_SOURCE\_MAP");

Iterator it = sm.keySet().iterator();

while(it.hasNext()){

String key = (String)it.next();

String value = (String)sm.get(key);

out.println("<option value=\""+value+"\">"+key+"</option>");

}

%>

--%>

</select>

* 1. caarray.ear/META-INF/security-config.xml

*<policy>*

*<application-policy name="caarray">*

*<authentication>*

*<login-module code="gov.nih.nci.security.cgmm.loginmodules.NullPasswordStackingLoginModule" flag="optional">*

*<module-option name="password-stacking">useFirstPass</module-option>*

*</login-module>*

*<login-module code="org.jboss.security.auth.spi.DatabaseServerLoginModule" flag="required">*

*<module-option name="password-stacking">useFirstPass</module-option>*

*<module-option name="dsJndiName">java:jdbc/CaArrayDataSource</module-option>*

*<module-option name="rolesQuery">SELECT cg.group\_name, 'Roles' FROM csm\_group cg, csm\_user\_group cug, csm\_user cu WHERE cg.group\_id = cug.group\_id AND cug.user\_id = cu.user\_id AND cu.login\_name = ?</module-option>*

*</login-module>*

*</authentication>*

*</application-policy>*

*</policy>*

* 1. caarray.ear/META-INF/security-config.xml

1. Deploy caArray Application
2. Check out CGMM / Download CGMM Release
3. Build CGMM (only if source code was checked out).
   1. Go to ‘cgmmweb’ directory (aka CGMM\_FOLDER)
      1. At the command prompt: ant –f cgmm\_build.xml
      2. The Build contents are available in the CGMM\_RELEASE\_FOLDER folder.
4. Copy to JBOSS\_HOME\server\default\deploy\jbossweb-tomcat55.sar folder, the following jars
   * 1. CGMM\_RELEASE\_FOLDER/cgmmapi.jar
     2. CGMM\_RELEASE\_FOLDER/catalina.jar (Note: This is a custom jar file)
     3. CGMM\_RELEASE\_FOLDER/jbossweb-tomcat55-sar-jars/\*.jar
     4. Please make sure to update version of jars relevant to caArray to avoid any conflicts with caArray application.
   1. Copy to JBOSS\_HOME\server\default\lib folder, the following jars
      1. CGMM\_RELEASE\_FOLDER/jboss\_default\_libs/\*.jar
5. Configure CGMM
   1. Verify the following properties are set in JBOSS\_HOME/server/default/deploy/properties-service.xml. Please ensure to specify correct path for each.
      1. gov.nih.nci.security.cgmm.syncgts.file = PATH\_TO\_jboss-4.0.5.GA/server/default/cgmm\_config/sync-description.xml
      2. gov.nih.nci.security.cgmm.properties.file = PATH\_TO\_jboss-4.0.5.GA/server/default/cgmm\_config/cgmm-properties.xml
      3. gov.nih.nci.security.configFile = PATH\_TO\_jboss-4.0.5.GA/server/default/cgmm\_config/ApplicationSecurityConfig.xml
      4. gov.nih.nci.security.cgmm.login.config.file = PATH\_TO\_jboss-4.0.5.GA/server/default/cgmm\_config/cgmm.login.config
   2. See the attached sample files for the same.
   3. Modify the database connection properties in cgmmweb.hibernate.cfg.xml
   4. Modify the ApplicationSecurityConfig.xml to point to correct application name ( i.e. ‘caarray’, this name should match in the CSM Schema).
   5. Modify cgmm.login.config and verify connection properties and the Application Policy name (i.e. ‘caarray’, this name should match in the CSM Schema).
6. Database Setup
   1. Make sure there is atleast one migrated user with roles associated. For example: caarrayadmin
   2. Replace ‘caarrayadmin’ in Table csm\_user.login\_name column with a valid caGrid ID. Make sure there is atleast one migrated
7. Configure JBoss Mail Service
   1. Modify the JBOSS\_HOME/server/default/deploy/mail-service.xml and add following entry with valid attribute values

<mbean code="org.jboss.mail.MailService" name="jboss:service=Mail">

<attribute name="JNDIName">java:/Mail</attribute>

<attribute name="User">sample\_user name </attribute>

<attribute name="Password">sample\_password</attribute>

<attribute name="Configuration">

<configuration>

<property name="mail.transport.protocol" value="smtp"/>

<property name="mail.smtp.host" value="Sample\_ mailfwd.nih.gov"/>

<!-- <property name="mail.smtp.port" value="465"/>-->

<property name="mail.smtp.auth" value="false"/>

<property name="mail.smtp.starttls.enable" value="false"/>

<property name="mail.debug" value="false"/>

</configuration> </attribute> </mbean>

1. (Optional) Configure CLM Audit Logging
2. Start JBoss
3. Access URL <http://server:port/caarray>. Click on Login on the left side.
4. It will prompt for caGrid credentials. Enter valid caGrid credentials. A successful login indicates configurations and setup was done correctly.

# Glossary

The following table contains a list of terms used in this document, with accompanying definitions.

| **Term** | **Definition** |
| --- | --- |
|  |  |
| Ant | Apache Ant is a Java-based build tool used to perform various build related tasks. For more information on how Ant is used within the SDK. See <http://ant.apache.org/> for more information on Ant itself. |
| caGrid | The cancer Biomedical Informatics Grid, or caBIG™, is a voluntary virtual informatics infrastructure that connects data, research tools, scientists, and organizations to leverage their combined strengths and expertise in an open federated environment with widely accepted standards and shared tools. The underlying service oriented infrastructure that supports caBIG™ is referred to as caGrid. See <http://www>.cagrid.org |
| Ehcache | Ehcache is a simple, fast and thread safe cache for Java that provides memory and disk stores and distributed operation for clusters. CSM uses ehcache in conjunction with Hibernate. See <http://sourceforge.net/projects/ehcache> for more information. |
| Hibernate | Hibernate is an object-relational mapping (ORM) solution for the Java language, and provides an easy to use framework for mapping an object-oriented domain model to a traditional relational database. Its purpose is to relieve the developer from a significant amount of relational data persistence-related programming tasks. See <http://www.hibernate.org/> for more information. |
| JAR | JAR file is a file format based on the popular ZIP file format and is used for aggregating many files into one. A  JAR file is essentially a zip file that contains an optional META-INF directory. |
| JAAS | The JAAS 1.0 API consists of a set of Java packages designed for user authentication and authorization. It implements a Java version of the standard Pluggable Authentication Module (PAM) framework and compatibly extends the Java 2 Platform’s access control architecture to support user-based authorization. |
| SAML | Security Assertion Markup Language (SAML) is an XML standard for exchanging authentication and authorization data between security domains, that is, between an identity provider (a producer of assertions) and a service provider (a consumer of assertions). SAML is a product of the OASIS Security Services Technical Committee |
| Spring | Spring Framework is a leading full-stack Java/JEE application framework. Led and sustained by Interface21, Spring delivers significant benefits for many projects, increasing development productivity and runtime performance while improving test coverage and application quality. See <http://www.springframework.org/> for more information. |
| XSD | XML Schema Definition. |
| Globus Toolkit | The Globus® Toolkit is an open source software toolkit used for building grids. It is being developed by the Globus Alliance and many others all over the world |
| IDP | Identity Provider, a.k.a IdP. For more information <http://asc.gsa.gov/portal/template/faq08.vm> |