CSM GAARDS Migration Module Guide

Version 0.6 -- For CSM Version 4.2

|  |  |  |
| --- | --- | --- |
|  | Center for Biomedical Informatics and Information Technology |  |
|  |
|  |

This is a U.S. Government work. May 01, 2009

Revision History

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

#### Revision History

|  |  |  |
| --- | --- | --- |
| Revision Date | Author | Summary of Changes |
| 10/31/2008 | Vijay Parmar | Initial Table of Contents |
| 11/05/2008 | Vijay Parmar | Added new chapters |
| 11/10/2008 | Charles Griffin | Review of initial draft |
| 11/12/2008 | Bronwyn Gagne | Doc converted to current CBIIT template, and edited as necessary. |
| 11/14/2008 | Vijay Parmar,  Bronwyn Gagne | Final review and release of updated guide – version 0.5 of CGMM for CSM 4.1. |
| 4/16/2009 | Vijay Parmar Bronwyn Gagne | Updated draft of guide. Added sections for new features such as Alternate Behavior, Standalone mode and other misc. configurability. |
| 5/01/2009 | Vijay Parmar Bronwyn Gagne | Final review and release of updated guide for CGMM v0.6 for CSM 4.1.0.1. |
| 08/19/2009 | Vijay Parmar  Santhosh Garmilla | Updated Command line automated installation steps |

Table of Contents

[About This Guide 1](#_Toc238738117)

[Purpose 1](#_Toc238738118)

[Scope 1](#_Toc238738119)

[Topics Covered 1](#_Toc238738120)

[Related Documentation 2](#_Toc238738121)

[Text Conventions Used 3](#_Toc238738122)

[Credits and Resources 3](#_Toc238738123)

[Chapter 1 CGMM Overview 5](#_Toc238738124)

[CGMM Architecture 5](#_Toc238738125)

[CGMM Solutions 6](#_Toc238738126)

[CGMM Process Flow 7](#_Toc238738127)

[CGMM Components 7](#_Toc238738128)

**[CGMM Filter (in the host application)](#_Toc238738129)** [7](#_Toc238738129)

**[CGMM Tool](#_Toc238738130)** [7](#_Toc238738130)

**[Authentication Service](#_Toc238738131)** [8](#_Toc238738131)

**[Dorian](#_Toc238738132)** [8](#_Toc238738132)

**[SyncGTS](#_Toc238738133)** [8](#_Toc238738133)

[Security Concepts 8](#_Toc238738134)

[Minimum System Requirements 9](#_Toc238738135)

[Chapter 2 Using the CGMM API 10](#_Toc238738136)

[Workflow 10](#_Toc238738137)

[CGMM API Services 11](#_Toc238738138)

[CGMMManager 11](#_Toc238738139)

[Integrating with the CGMM API 15](#_Toc238738140)

[Importing the CGMM Authentication API 15](#_Toc238738141)

[Obtaining the CGMMManager 16](#_Toc238738142)

[Authenticating Users 16](#_Toc238738143)

[Migrating Users 16](#_Toc238738144)

[Integrating Auto Start SyncGTS servlet 17](#_Toc238738145)

[Configurations for CGMM API 17](#_Toc238738146)

[Chapter 3 Audit Logging 19](#_Toc238738147)

[Overview 19](#_Toc238738148)

[JAR Placement 19](#_Toc238738149)

[Enabling CLM APIs in Integration with CGMM APIs 19](#_Toc238738150)

[Event Logging 19](#_Toc238738151)

[Common Logging Database 20](#_Toc238738152)

[JDBC Appender 20](#_Toc238738153)

[Deployment Steps 21](#_Toc238738154)

[Step 1: Create and Prime MySQL Logging Database 21](#_Toc238738155)

[Step 2: Configure the log4j.xml file for JBoss 21](#_Toc238738156)

[Step 3: View the Logs 21](#_Toc238738157)

[Chapter 4 Using the CGMM Tool 22](#_Toc238738158)

[Overview 22](#_Toc238738159)

[Default Behavior 22](#_Toc238738160)

[Default Behavior Workflows/Scenarios 23](#_Toc238738161)

[Default Behavior Scenario 1: User Logs In with CSM Account 24](#_Toc238738162)

[Default Behavior Scenario 2: User Logs In with caGrid Account 28](#_Toc238738163)

[Alternate Behavior 31](#_Toc238738164)

[Alternate Behavior Workflows/Scenarios 32](#_Toc238738165)

[Alternate Behavior Scenario 1: User Logs In with CSM Account 32](#_Toc238738166)

[Standalone Mode 37](#_Toc238738167)

[Configuring the CGMM Tool 37](#_Toc238738168)

[Chapter 5 Integrating CGMM with Container Managed Security 39](#_Toc238738169)

[Overview 39](#_Toc238738170)

[Integration Steps 40](#_Toc238738171)

[Chapter 6 CGMM Installation and Deployment 41](#_Toc238738172)

[Release Contents 42](#_Toc238738173)

[Installation Pre-Requisites 43](#_Toc238738174)

[Refactoring Host Application (Default Behavior) 43](#_Toc238738175)

[Configure Container Managed Security (Alternate Behavior) 44](#_Toc238738176)

[caGrid Security Infrastructure 45](#_Toc238738177)

[Identify Configuration Parameters for CGMM 46](#_Toc238738178)

[Deployment Checklist 46](#_Toc238738179)

[Deployment Steps 47](#_Toc238738180)

[Appendix A CGMM Properties XSD File 51](#_Toc238738181)

[Appendix B Sample CGMM Properties File 57](#_Toc238738182)

[Appendix C Sample Sync Description File 59](#_Toc238738183)

[Appendix D CGMM with Reference Implementation 60](#_Toc238738184)

[Appendix E Testing CGMM Container Managed Security Integration 63](#_Toc238738185)

[Appendix F Integrating CGMM Container Managed Security with caArray 66](#_Toc238738186)

[Appendix G Installing CGMM in existing Container using Command line installer 70](#_Toc238738187)

[Glossary 71](#_Toc238738188)

[Index 73](#_Toc238738189)

About This Guide

This preface introduces you to the CSM GARRDS Migration Module (CGMM) Guide.

Topics in this section include:

* on this page
* on this page
* on page 1
* on page 2
* on page 3
* on page 3

## Purpose

This guide provides all the information application developers need to successfully use 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 caGrid information pertaining to the CGMM is 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 to the caGrid Knowledge Center Wiki located at: <http://www.cagrid.org/display/cagridhome/Home>.

## Topics Covered

In order to most effectively gain the information you need to use the CGMM, we strongly recommend you review all of the information provided in this guide. In particular, you should start with the first and second chapters of this guide, to gain proper background for using the CGMM.

Below you will find a brief description of what information resides in each chapter.

* *,*  provides an overview of CGMM and its capabilities.
* , provides the necessary information and workflow for a developer to successfully integrate the CGMM API into their application.
* , provides information on how to integrate Audit Logging for the CGMM API or CGMMWeb.
* , provides workflows scenarios for using both the Default and Alternate behavior of the CGMM Tool. This chapter includes information about using CGMM for authentication, migration, and/or new caGrid user creation.
* , provides information on integrating CGMM functionality for applications that use container managed security.
* *,*  provides the information and steps necessary to install and deploy the CGMM Tool with a working installation of a host application.
* , provides a sample CGMM properties XSD file.
* , provides a sample CGMM properties configuration file.
* , provides a sample Sync Description configuration file.
* , provides the steps necessary to install the reference implementation cgmmHostWeb web application along with the cgmmweb web application.
* , provides sample steps for testing CGMM integration with an application that uses container-managed security.
* , provides an example steps for integrating CGMM’s container-managed security with the caArray application.

The Glossary, located behind the appendices, is provided to clarify abbreviations and terms used in this document.

## Related Documentation

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

* Common Security Module (CSM) v4.1 Technical Guide
* CSM GAARDS User Migration Design Document.

These and other documents can be found on the CSM website: <http://ncicb.nci.nih.gov/NCICB/infrastructure/cacore_overview/csm>

Additional information and FAQ regarding the CGMM are available from the CSM Wiki page located at: <https://wiki.nci.nih.gov/x/4wBB>.

## Text Conventions Used

This section explains conventions used in this guide. The various typefaces represent interface components, keyboard shortcuts, toolbar buttons, dialog box options, and text that you type.

| Convention | Description | Example |
| --- | --- | --- |
| Bold | Highlights names of option buttons, check boxes, drop-down menus, menu commands, command buttons, or icons. | Click Search. |
| URL | Indicates a Web address. | http://domain.com |
| text in Small Caps | Indicates a keyboard shortcut. | Press Enter. |
| text in Small Caps + text in Small Caps | Indicates keys that are pressed simultaneously. | Press Shift + Ctrl. |
| Italics | Highlights references to other documents, sections, figures, and tables. | See Figure 4.5. |
| Italic boldface monospace type | Represents text that you type. | In the New Subset text box, enter Proprietary Proteins. |
| Note: | Highlights information of particular importance. | Note: This concept is used throughout this document. |
| { } | Surrounds replaceable items. | Replace {last name, first name} with the Principal Investigator’s name. |

## Credits and Resources

|  |  |  |
| --- | --- | --- |
| caCORE CSM Development and Management Teams | | |
| CSM Development and QA Teams | Documentation | Program Management |
| Vijay Parmar1 | Vijay Parmar1 | Sichen Liu3 |
| Aynur Abdurazik2 | Bronwyn Gagne4 | Satish Patel1 |
|  |  |  |
| 1 Ekagra Software Technologies | 2 Science Applications International Corp. (SAIC) | 3 National Cancer Institute Center for Biomedical Informatics and Information Technology |
| 4 Lockheed Martin |  |  |

|  |  |
| --- | --- |
| CSM Resources | |
| Name | URL |
| Mailing List | [security-csm-user@gforge.nci.nih.gov](mailto:security-csm-user@gforge.nci.nih.gov) |
| Mailing List Archive | <http://gforge.nci.nih.gov/pipermail/security-csm-user> |
| GForge Project Home | <http://gforge.nci.nih.gov/projects/security> |
| CSM Support Tracker | <http://gforge.nci.nih.gov/tracker/?atid=131&group_id=12&func=browse> |

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

#### 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.

#### Release Schedule

This guide was originally created to correspond with the 4.1 version of caCORE CSM and the CSM GAARDS Migration Module, which was released in November 2008 by the NCI Center for Biomedical Informatics and Information Technology (CBIIT), formerly the National Cancer Institute Center for Bioinformatics (NCICB).

The updates to this guide were added in April 2009, to coincide with release version 1.0 of the CGMM. This version contains a variety of features and enhancements to configurability and usability of the CGMM.

# CGMM Overview

The chapter provides an overview of the architecture, and discussions of the components involved in the CSM GAARDS Migration Module (CGMM), security concepts, and minimum system requirements.

Topics in this chapter include:

* on this page.
* on page 7.
* on page 8.
* on page 9.

## CGMM Architecture

The CGMM provides a two-tiered solution for existing web applications, namely to:

1. Migrate existing CSM accounts to caGrid accounts,
2. 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 has been 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.
* Ability to use GAARDS based authentication.
* Provisioning of new users with Grid identities.
* To use caBIG approved identity providers, thus allowing federation of identities.
* Provide a configurable “Look and Feel”
* Provide configurable caGrid identity providers for authentication.

As shown in Figure 1‑1 below, the CGMM architecture allows existing host applications to integrate with CGMM and sort of “off-load” their authentication functionality to CGMM. CGMM is expected to intercept and migrate CSM (local) accounts, and enforce the use of caGrid accounts offered by various Identity providers in caBIG.

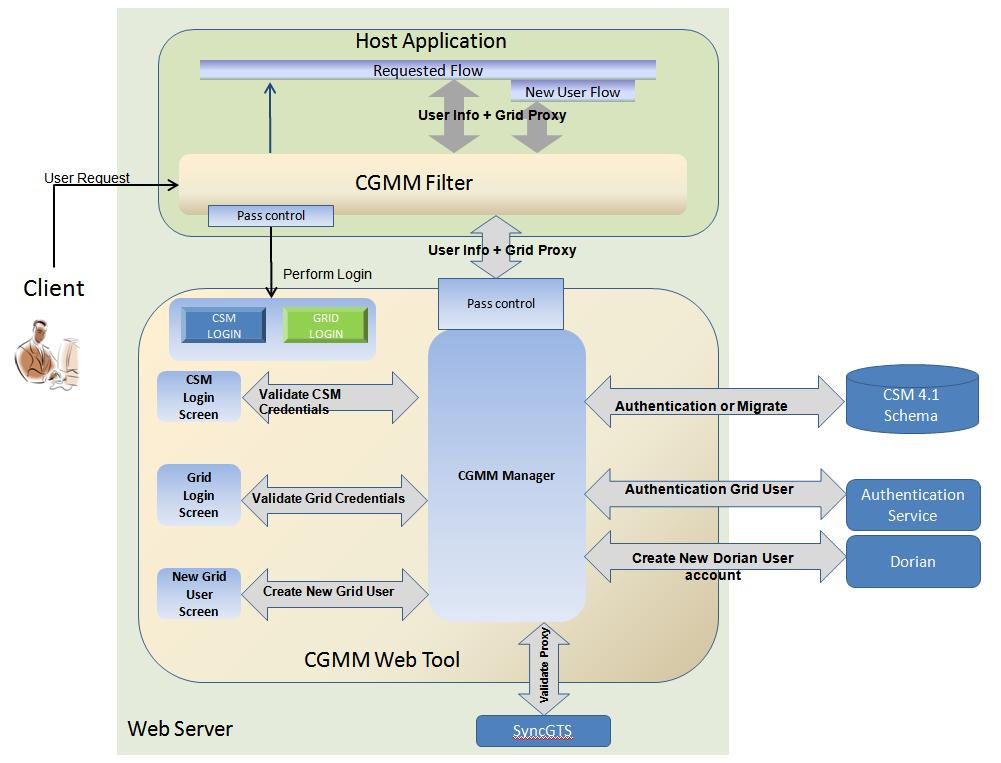


Figure ‑ CGMM Architecture

The above diagram demonstrates the overall architecture of CGMM, the components involved, and their interactions at a high level. As shown, 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, provided by the CGMM to forward all un-authenticated user requests. The GAARDS components used are Authentication Service, Dorian Service, and SyncGTS.

### CGMM Solutions

The CGMM provides the following solutions for the host application:

Authentication – CGMM validates and verifies a user’s CSM (local) credentials to initiate migration, and validates and verifies 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 user’s information and Grid Proxy.

Migration – CGMM migrates or transforms a CSM user to a 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.

New caGrid User Creation – CGMM creates 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.

Configurable CGMM Tool – CGMM allows for the enabling or disabling of the New caGrid User creation feature of the CGMM Tool. CGMM also allows for the configuration of other information, such as host application information and Authentication Service and Dorian Service information.

CGMM API – The CGMM API allows programmatic access and integration of the CGMM features.

### CGMM Process Flow

The overall flow for CGMM is as follows:

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.

## CGMM Components

The following are the minimum set of components involved in the CGMM Framework. This section describes the components shown in the CGMM Architecture diagram above (Figure 1‑1).

### 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 log the user into the Host application. Depending on whether the user is an existing application user or not, control is passed back to either the login workflow or the new user creation workflow respectively.

### CGMM Tool

The CGMM Tool is provided to assist in the migration of local CSM accounts to caGrid accounts. Performing this migration allows GAARDS-based authentication to the host application via single set of credentials. The CGMM Tool is a separate web application that resides in the same container as the Host web application. CGMM also provides the Servlet Filter that gets placed in front of the host application, intercepting and routing each user request for login or migration purpose. A detailed workflow of the migration module and the considered scenarios are provided in *,*  on page 22.

### Authentication Service

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

### Dorian

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

### SyncGTS

SyncGTS is installed in CGMM for the host application. The SyncGTS daemon keeps the host application in sync with the Grid Trust Fabric, and updates the CRL’s accordingly. Once the CGMM obtains Grid proxy from Dorian, it validates 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 the table below. 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 is 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 caGrid users, creation of new caGrid accounts, 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 ‑ Security concept definitions

## Minimum System Requirements

The software listed in the table below is required and is not included with CGMM. The product name, version, description, and URL hyperlinks are provided.

|  |  |  |  |
| --- | --- | --- | --- |
| 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  (Only one is required) | 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  (Only one is required) | 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 |

Table ‑ Minimum Software Requirements

# Using the CGMM API

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.

Topics in this chapter include:

* on this page.
* on page 11.
* on page 15.
* on page 17.

## 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 section more details.
3. Add the StartSyncGTSServlet servlet to your host web application. See on page 17 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, new caGrid User creation, and synching with the caGrid Trust Fabric.

### CGMMManager

The CGMM Manager is an interface that provides the functionality described in Table 2‑1 below. This functionality is implemented by the CGMMManagerImpl class, available in the CGMM APIs, and includes the following:

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

The following table lists and describes all of the CGMMManager API methods that perform these tasks:

| 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 the CGMMUser.  Parameters:  loginID The Login ID of the User available in CSM. This ID can be a caGrid 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 is not 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 caGrid 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 caGrid 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 caGrid 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 caGrid account.  password The password for user caGrid 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 an 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 an 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 Name, 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 an Authentication Service URL specification exception. |

Table ‑ CGMM API - CGMM Manager

## Integrating with the CGMM API

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 2‑1 above.

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 last two (highlighted) import statements to the action classes, as shown below:

**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;

### Obtaining the CGMMManager

The sample shown below provides example code to use the CGMM API - CGMMManager class in the ‘sampleHostApplication’ host application:

CGMMManager cgmmManager = **null**;

**try** {

cgmmManager = **new** CGMMManagerImpl();

} **catch** (CGMMConfigurationException e) {

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

}

### Authenticating Users

The sample shown below provides example code for authenticating CSM users in the ‘sampleHostApplication’ host application.

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");

### Migrating Users

The sample shown below provides example code for migrating users in the ‘sampleHostApplication’ host application.

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.");

}

### Integrating Auto Start SyncGTS servlet

To integrate the StartSyncGTSServlet in the host application, add the configuration shown in the example below to the web.xml file 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 invoking any secured caGrid Services.

<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>

## Configurations for CGMM API

For successful integration of CGMM API into a host web application, the following configuration files must be configured correctly. Table 2‑2 below shows the configuration files and changes needed for CGMM.

| Configuration File | Description |
| --- | --- |
| Cgmm-properties.xml | Required to specify the CGMM information, Host Application information and Authentication Service/Dorian information.  Sample provided in *,*  on page 57.  Refer the cgmm-propertiex.xsd shown in on page 51 for more information.  The CGMMManager retrieves this file based on the System property  gov.nih.nci.security.cgmm.properties.file. |
| Sync-description.xml | Required for the StartSyncGTSServlet.  Refer to the sample provided in on page 59 for more information.  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.  NOTE: 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 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 noted above.  It points to the CSM Schema for the host application.  Replace <<name>> with the host application context name. |

Table ‑ CGMM Configuration Files

# Audit Logging

This chapter 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.

## Overview

In an effort to make CGMM compliant with CRF 21/ part 11, the CGMM provides auditing and logging functionality. The CGMM audit logging capability is provided through the Common Logging API available from clm-\*.jar.

Client application developers can configure audit logging is configurable via an application property configuration file. By placing the clm.jar, along with this application property configuration file, in the same class path as the cgmmapi.jar file, the client application is able to utilize the built-in audit logging functionality. The logging results can be saved into a database or a flat text file, depending on the configuration.

## JAR Placement

The Audit Logging Application is available as a JAR file called clm-4.1.jar. This jar, along with the cgmmapi.jar must 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 allow for logging of every event that the user performs. For Authentication/Login, Migration, New User Creation, and other Services, the CGMM APIs log the events of the user.

The CGMM Web can perform all of 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 configuration file of JBoss, as shown in JDBC Appender section below.

### Common Logging Database

The Common Logging Database 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 the Audit logs, the CLM provides an asynchronous JDBC Appender. Therefore, an application that wants to enable the audit logging for CGMM APIs should also configure this Appender.

Shown below is a sample log4j file entry:

<?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 ‑: Example log4j.xml file

**NOTE:** In order to use CLM features without using CGMM, the client application can separately download and install CLM. In this case CLM can be used (even without using CGMM) 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

Use the steps outlined in this section to enable the Audit Logging capabilities provided by CGMM (via CLM).

### Step 1: Create and Prime MySQL Logging Database

1. Create a database that will persist the audit logs generated as a result of usage of the CGMM APIs
2. Refer to the CLM Programmer’s Guide 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 with the CGMM release to configure the log4j.xml file for JBoss. (see Figure 3‑1 above)
2. Replace the <<APPLICATION\_NAME>>, <<SERVER\_NAME>>, <<PORT>>, and <<CLM\_SCHEMA\_NAME>> entries with the appropriate corresponding values for the schema created in Step 1.
3. Replace the values for the <<DB\_USER>> entry with the user name that has access on the schema. Also replace the <<PASSWORD>> with the corresponding password for this user.
4. Configure the logger that corresponds with whether the application wants to enable the event audit logging for Authentication & Authorization, or object state audit logging for the Authorization. **NOTE:** The names of the loggers must not differ from the sample.
5. In the case of the 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 Programmer’s Guide.

# Using the CGMM Tool

This chapter demonstrates the implemented CGMM Default Behavior and Alternate Behavior workflows and scenarios followed by the configurable features of the CGMM Tool.

Topics in this chapter include:

* below.
* below.
* on page 23.
* on page 31.
* on page 32.
* on page 37.
* on page 37.

## Overview

The CGMM Tool is a web application that, on behalf of the host application, allows authentication of CSM/caGrid users, migration of a CSM user account to a caGrid user account, and/or creation of new caGrid accounts for users.

The CGMM tool is configurable and was created considering customizations by/for the host applications. The CGMM tool requires a low level of effort for modification and configuration 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. For more information, see *,*  beginning on page 10,

## Default Behavior

The phrase **Default Behavior** is the term being used to define the behavior and workflows available with the original version 0.5 release of CGMM.

CGMM default behavior is meant for existing web applications that would like to utilize the CGMM Web application for the following activities:

* Authentication,
* Migration,
* New caGrid user creation.

The default behavior also assumes that the host application will be using a Servlet filter (CGMMFilter) to intercept and interpret the information for the logged in/migrated users forwarded by the CGMM Web application. This information includes the user credentials, first/last name, and email address.

## Default Behavior Workflows/Scenarios

The CGMM Tool’s default behavior allows for 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. Based on that, there are two primary scenarios with underlying situations addressed by the CGMM Tool:

1. User logs in with CSM account and
2. User has a caGrid account.
3. User does not have a caGrid account.
4. User logs in with caGrid account and
5. User has already been migrated.
6. User has a CSM account.
7. User does not have a CSM account.

NOTE: The CGMM tool DOES NOT addresses the scenario where a user has neither a CSM (local) Account nor a caGrid account. In this case, the host application needs to address this scenario.

The sections that follow look at the user interface workflow of the CGMM by going through each of the scenarios mentioned above. Figure 4‑1 below shows the CGMM Tool Home page.

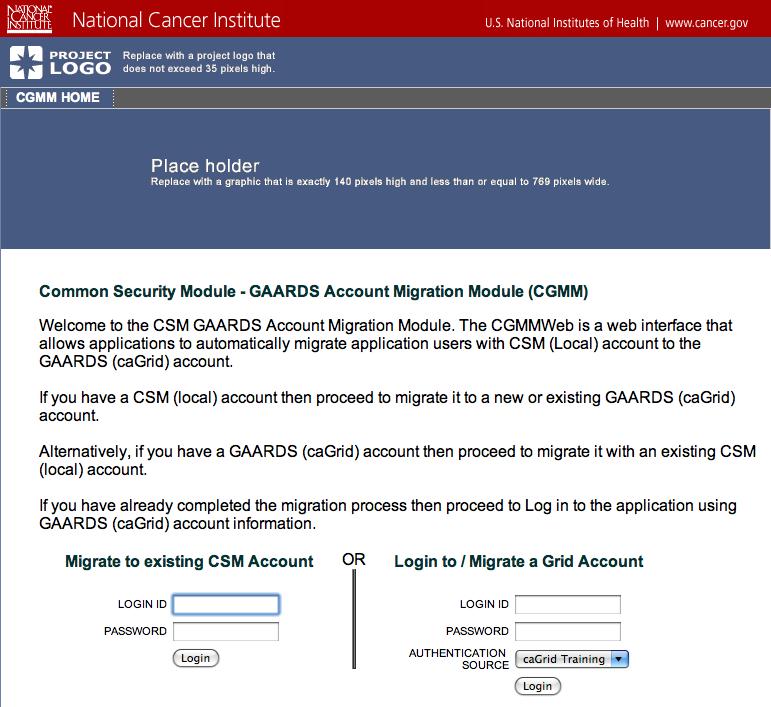


Figure ‑ CGMM Home Page

The home page provides details and basic instructions to the user regarding how to proceed using the tool, depending on their situation.

### Default Behavior Scenario 1: User Logs In with CSM Account

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

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

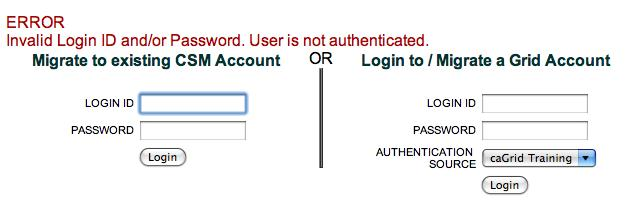


Figure ‑ CGMM - CSM Login Error

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

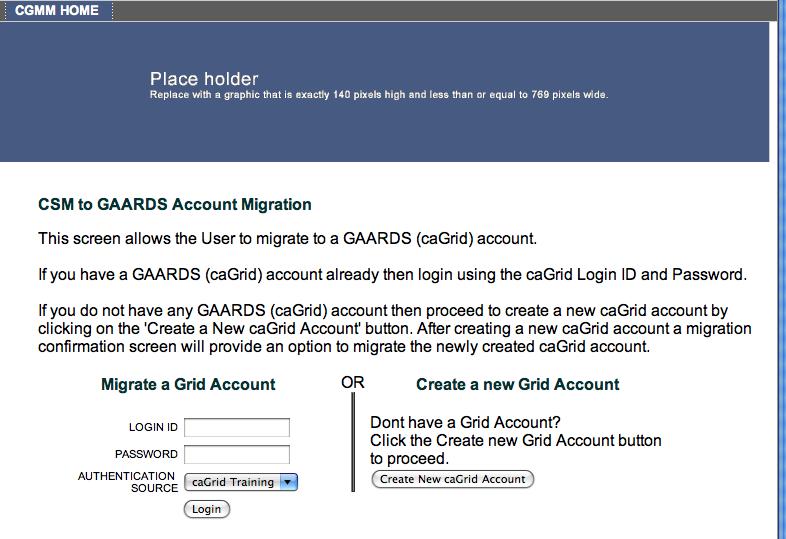


Figure ‑ CSM Login success page/Grid Login Page

#### Default Behavior Scenario 1-a: User Has caGrid Account

If the user already has an existing caGrid account, they can proceed to migrating to using their caGrid account by providing their caGrid Login ID and Password, and selecting the appropriate Authentication Source (Authentication Service).

##### User Logs In with caGrid Login ID and Password

After the user enters their caGrid login credentials and clicks Login, the CGMM Tool validates the caGrid account against the provided Authentication Source.

If the credentials are valid, the CGMM Tool displays the Confirm Migration screen to the user.



Figure ‑ CSM to GAARDS Account Migration Page

##### User Chooses to Migrate His/Her Account

On the migration confirmation page, the user has the option to cancel the migration or confirm it.

When the user selects to migrate by clicking the Yes, Migrate my CSM Account button, CGMM migrates the CSM account to the caGrid account in the CSM Schema of the host application. CGMM also marks the user as migrated.

Once the migration process is complete, the CGMM Tool takes the user to the migration confirmation page. From this page, the user can log into the host application.



Figure ‑ Migration Complete Page

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 user’s Grid Proxy as request attributes, and forwards the request to the Host application. This request is forwarded to the Host Applications User Home page, specified in the CGMM properties configuration. The CGMM then relinquishes control to the Host application.

If the request is accepted, the user is forwarded by the CGMM to the Host application User Home page.



Figure ‑ Host Application User Home Page (migration complete)

The above figure shows the User Home page for the “HostWeb” web application, shown here as a reference for implementation.

#### Default Behavior Scenario 1-b: User Does Not Have caGrid Account

If the user has a CSM login but does not have an existing caGrid account, the user can select to obtain a new caGrid account by clicking the Create New caGrid Account button. The Create new caGrid Account form appears.

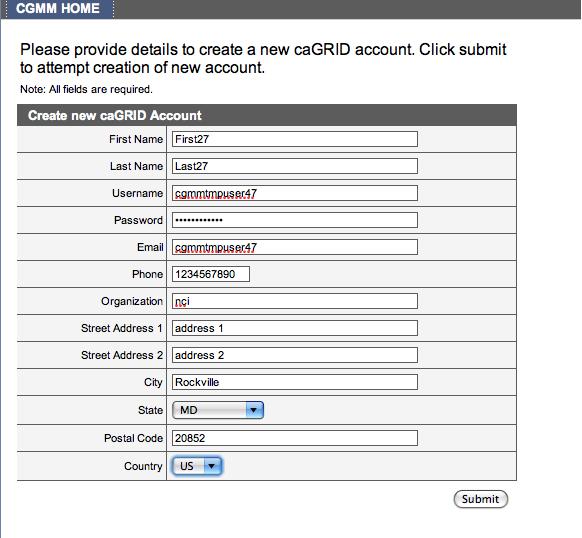


Figure ‑ New caGrid Account Form

The User must provide all of the requested information to proceed.

After completing all of the fields, the user must click Submit. An account details page appears, asking the user to review the details entered into the form for creating the new caGrid account.

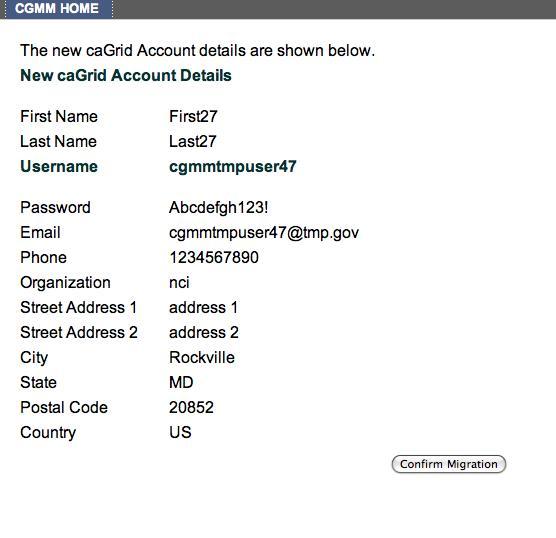


Figure ‑ New caGrid account information confirmation page

After confirming the details, the user must click Confirm Migration.

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

If the account creation is successful, the CGMM tool returns a complete/success page.

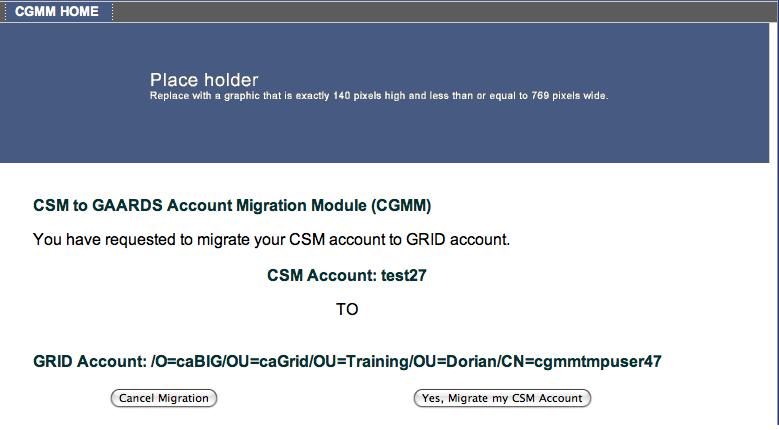


Figure ‑ Account creation complete/success page

At this point, the user has the option to cancel the migration or select to migrate their CSM account to their newly created caGrid account.

When the user selects to migrate by clicking the Yes, Migrate my CSM Account button, CGMM migrates the CSM account to the new caGrid account in the CSM Schema of the host application. CGMM also marks the user as migrated.

Once the migration process is complete, the CGMM Tool takes the user to the migration confirmation page. The user can now log into the host application.



Figure ‑ Migration complete page

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 user’s Grid Proxy as request attributes, and forwards the request to the Host application. This request is forwarded to the Host Applications User Home page that is specified in the CGMM properties configuration. The CGMM then relinquishes control to the Host application.

If the request is accepted, the user is forwarded by the CGMM to the Host application User Home page (as shown in Figure 4‑6 on page 26).

### Default Behavior Scenario 2: User Logs In with caGrid Account

If the User has a caGrid account, they can login by providing their caGrid username and password, and then selecting the appropriate Authentication Source from the drop-down list. The User then clicks Login.

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

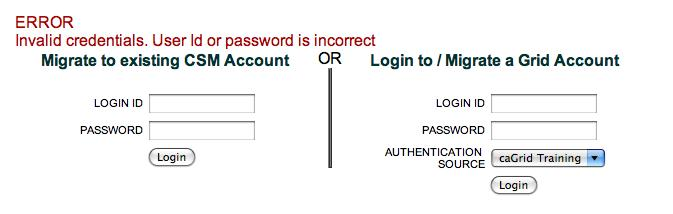


Figure ‑ CGMM - caGrid Login Error

#### Scenario 2-a: User Is Already Migrated

After entering their caGrid login credentials, 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, the CGMM Tool populates the HTTP Request with user’s details and Grid Proxy, and then forwards the request to the host application’s User Home page as shown in Figure 4‑6 on page 26.

#### Scenario 2-b: User Has CSM Account

After entering their caGrid login credentials, 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 has not been migrated, the tool presents the user with a CSM Login Page in which they can enter their CSM login credentials or create a new CSM account.

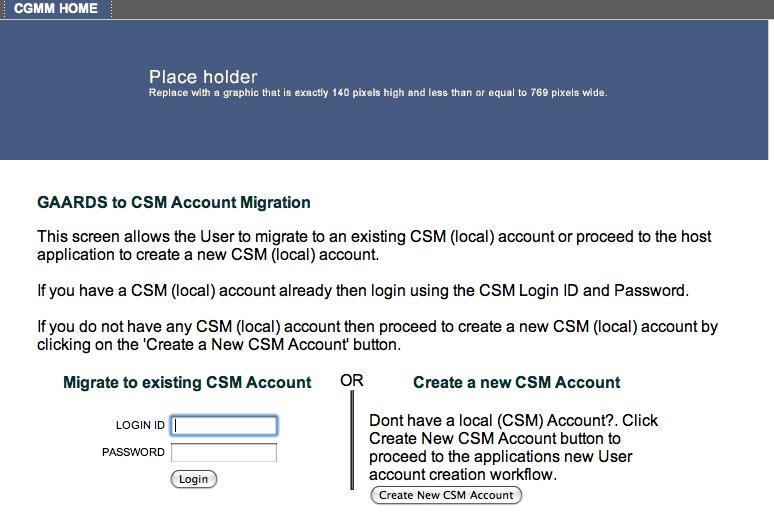


Figure ‑ caGrid Login Success - CSM Login Page

Since in this scenario the user has an existing CSM account, the user can proceed to migrate CSM account by providing their CSM Login ID, Password, and clicking Login.

##### User Logs In with CSM Login ID and Password

After the user provides their CSM login credentials, the CGMM Tool validates the credentials provided by the user. If the credentials are valid, the CGMM Tool displays the Confirm Migration screen.



Figure ‑ CSM to GAARDS Account Migration Page

If the user selects Yes, Migrate my CSM Account, 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.



Figure ‑ Migration Complete Page

When the user clicks the Log in to <<Host Application Name>> button, the user is logged in and is forwarded by the CGMM to the Host application User Home page.



Figure ‑ Migration Complete Page - Host Application User Home Page

The above figure shows the User Home page for the “HostWeb” web application, shown here as a reference for implementation.

#### Scenario 2-c: User Does Not Have a CSM Account

If the user has logged in with their caGrid account but does not have a CSM account, when they are presented with the CSM login page, they are left with the option to request the creation of a new CSM account for the host application.

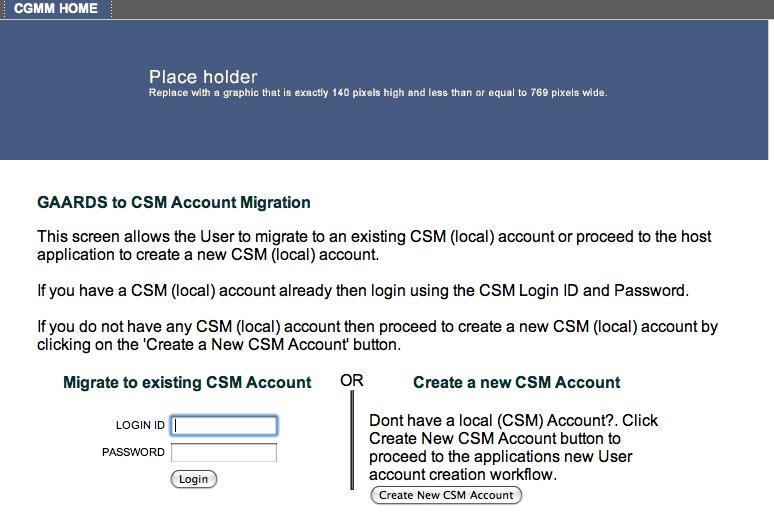


Figure ‑ caGrid Login Success - CSM Login Page

When the user selects Create New CSM Account, the CGMM tool populates the HTTP request with caGrid User account and the user’s Grid Proxy, and forwards the request to the Host application to relieve control. The CGMM tool then forwards the request to the host application’s New CSM User Creation page. The CGMM obtains the context and URL for this page from the CGMM properties configuration file.

## Alternate Behavior

Alternate Behavior is the term being used to define the new features of the CGMM Web application.

The alternate behavior is meant for existing web applications that want to utilize the CGMM Web application for account migration only.

The alternate behavior assumes that the host application will perform authentication and new caGrid user creation by itself. The CGMM Web application notifies the application administrator, via Email, of the new caGrid user creation request being sent by user.

The alternate behavior also assumes that the host application does not use a Servlet Filter (CGMMFilter) to intercept or interpret logged in/migrated users credentials forwarded by the CGMM Web application. Hence, using the alternate behavior, the CGMM Web application redirects users to configured host application home/login page URLs.

## Alternate Behavior Workflows/Scenarios

The CGMM Tool’s alternate behavior allows multiple scenarios/workflows based on the user. The user must have a Local (CSM) account. The user may or may not have a caGrid Account. Based on those criteria there are two different scenarios addressed by the CGMM Tool. The scenarios are as follows:

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

**NOTE**: The CGMM tool DOES NOT addresses the scenario where the user has neither a CSM (local) Account nor a caGrid account. The host application needs to address this scenario.

The sections that follow look at the user interface workflow of the CGMM by going through each of the scenarios mentioned above. Figure 4‑17 below shows the CGMM Tool Home page.

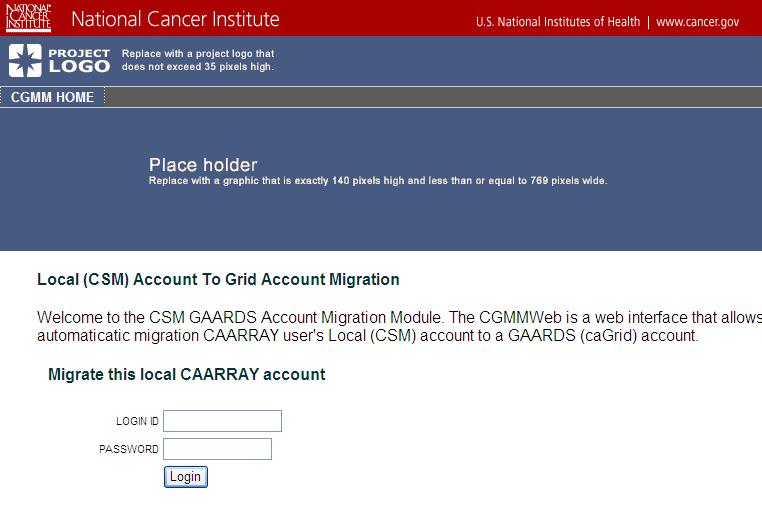


Figure ‑ CGMM Home page (alternate behavior)

### Alternate Behavior Scenario 1: User Logs In with CSM Account

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

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

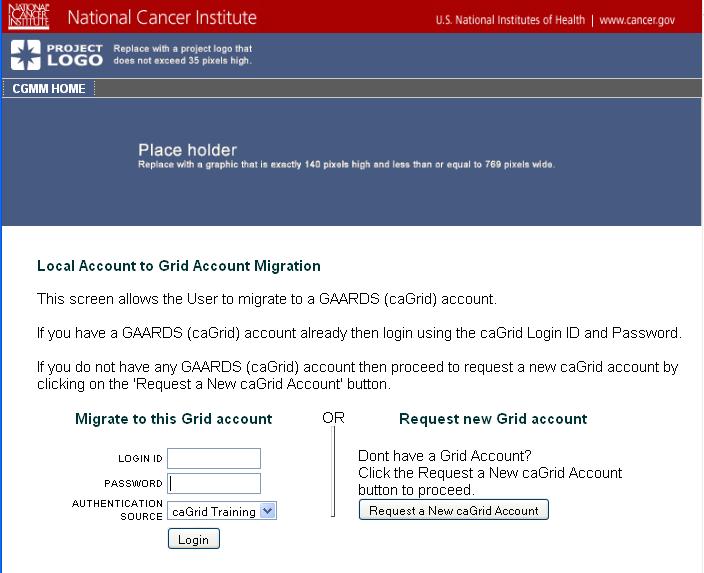


Figure ‑ CSM Login success page / Grid Login Page

#### Alternate Behavior Scenario 1-a: User Has caGrid Account

If the user already has an existing caGrid account, they can proceed to migrate to using their caGrid account by providing the Login ID and Password and then selecting the appropriate Authentication Source (Authentication Service).

##### User Logs In with caGrid Login ID and Password

After the user clicks Login, the CGMM Tool validates the caGrid account credentials provided. If the credentials are valid, the CGMM Tool displays the ‘Confirm Migration’ screen to the user.

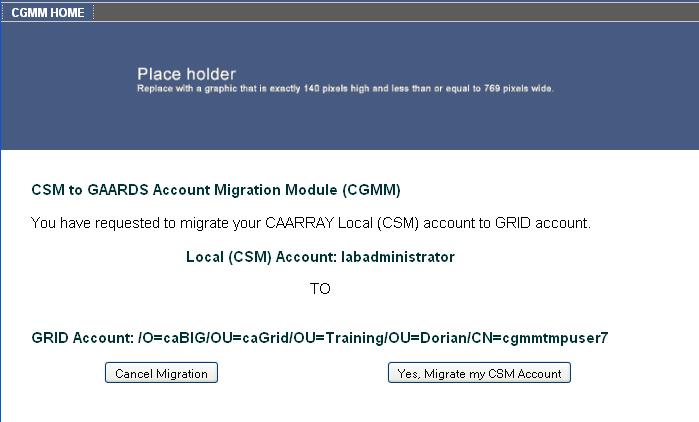


Figure ‑ CSM to GAARDS Account migration page

##### User Chooses to Migrate His/Her Account

On the migration confirmation screen, the user has the option to cancel the migration or confirm it. If the user selects to migrate by clicking the Yes, Migrate my CSM Account button, the CGMM migrate the CSM account to the caGrid account in the CSM Schema of the host application. CGMM also marks the user as migrated.

Once the migration process is complete, the CGMM Tool takes the user to the migration confirmation page. The user can now log into the host application.



Figure ‑ Migration complete page

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

#### Alternate Behavior Scenario 1-b: User Does Not Have a caGrid Account

If the user does not have an existing caGrid account, the user can request a new account. After logging in with their CSM account, the Grid account migration page provides the user with the option to Request a New caGrid Account, as shown in Figure 4‑22 below.

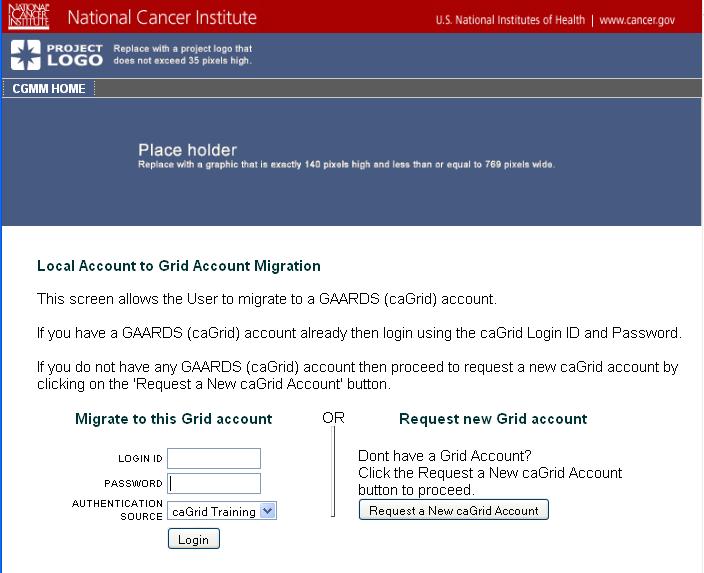


Figure ‑ CSM Login success page / Grid Login Page

When the user requests a new caGrid account, a form appears requesting information for creating the account. The User must provide all of the requested information to proceed.

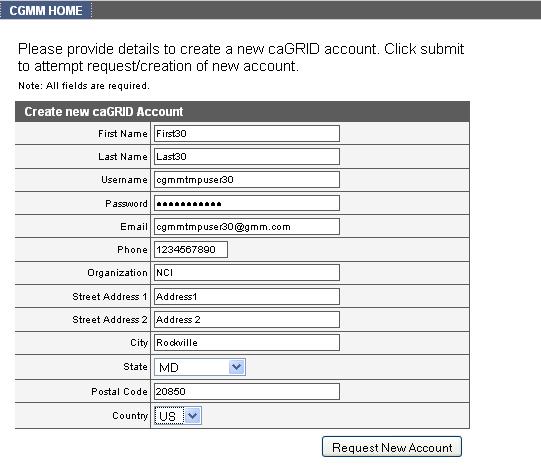


Figure ‑ New caGrid Account Request page

After completing the fields, the user must click Request New Account.

The CGMM attempts to send an email to the host application administrator Email ID provided in the CGMM configuration file. The configuration file should also contain the JNDI Name for the mail service.

The email request created will contain the administrator’s name in the **To** field, the requestor’s email address in the From field, and a subject line indicating that the message is a request for a new caGrid account. The body of the email contains the details provided by the user in the new account request form.

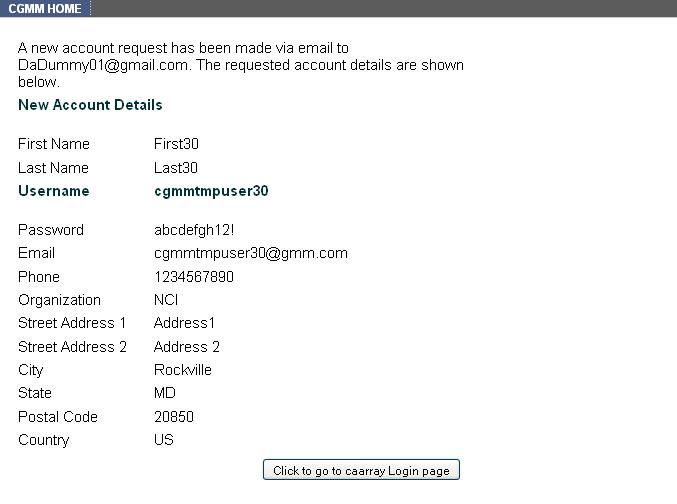


Figure ‑ New caGrid Request submitted via email

If the email is submitted successfully, the CGMM shows the details to the user.

At this point, the user can use the Click to go to <<HostApplicationName>> Login page to go to the host application. The CGMM redirects the user to the host application login page.

An example of the email sent to the host application administrator is shown below.

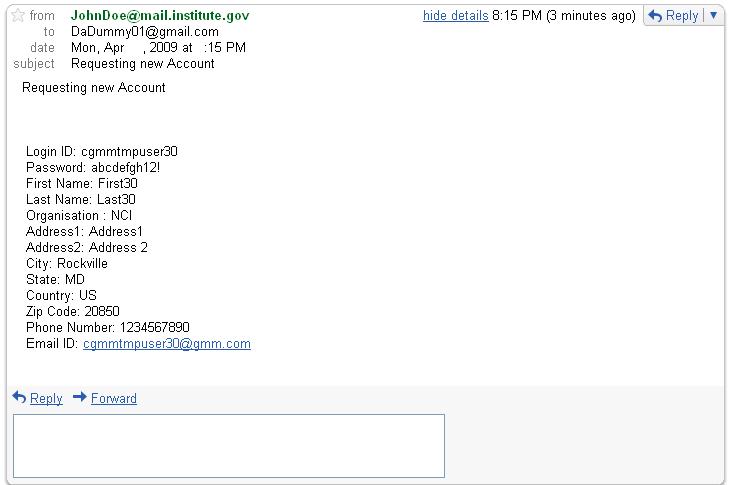


Figure ‑ Email sent to the host application administrator

## Standalone Mode

The Standalone mode is new feature provided for the CGMM Web application.

In Standalone Mode, the CGMM Web assumes there is no Host Web application that is co-hosted in the same container. In this mode, the CGMM Web does not forward or re-direct the user to any other application, after it is done with migration.

The details for configuring CGMM for standalone mode are indicated throughout the remaining sections of this document.

## Configuring the CGMM Tool

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 needs of the host application. This is achieved by configuring the cgmm-information section of the cgmm-properties.xml file 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 to the host application context relative URI.
3. CGMM Information

The CGMM information configuration allows the following:

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

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

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 application Home page URL.
4. Configurable host application User Home Page URL.
5. Configurable host application User Login Page URL (for alternate behavior only).
6. Configurable host application new CSM user page URL.
7. Configurable host application Mail Service JNDI Name (for alternate behavior only).
8. Configurable host application Mail ‘To’ Email ID (for alternate behavior only).
9. Configurable host application Mail ‘From’ Email ID (for alternate behavior only).
10. Configurable host application Mail ‘Subject’ text (for alternate behavior only).
11. Configurable host application Logo URL (for alternate behavior only).
12. Configurable host application Logo Alt Text (for alternate behavior only).
13. Authentication Service/Dorian Information

The Authentication Service list allows specifying one or more Authentication Services to use for authentication purposes. 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

The ability to integrate CGMM is now available for applications that utilize Form-based security using JBoss/Tomcat and that would like to integrate CGMM API into their existing authentication workflow.

This chapter provides details regarding the integration of CGMM API with applications that use existing container-managed form-based security.

## Overview

For web applications that utilize container-managed security with form-based authentication, the integration of CGMM API to authenticate caGrid credentials requires modification to the existing JBoss/Tomcat installation. caGrid Authentication requires three pieces of user input: Login Name, Password, and caGrid Authentication Source.

The default Form Authenticator available (from Tomcat) allows only two input parameters whereas the caGrid Authenticator requires three parameters. To accommodate this discrepancy, the CGMM API now contains a Custom Form Authenticator.

The JBoss application server recognizes only five types of Authenticators, one of which is the FormAuthenticator. However there is no configurable alternative to specify a custom form authenticator.

The summarized steps for completing CGMM integration of the JBoss application server with custom Form-based container managed security are as follows:

* The existing Web Application must utilize the custom Form Authenticator (CaGridFormAuthenticator) instead of the current Form Authenticator. Meaning the security domain specified in the web.xml file must use the custom authentication method CAGRIDFORM instead of the default FORM authentication method. See on page 44 for specifics on configuring the web.xml file.
* The catalina.jar file located in the folder JBOSS\_HOME/server/default/deploy/jbossweb-tomcat55.sar should be modified as follows:
* In the org/apache/catalina/startup/Authenticator.properties file, add the following property: CAGRIDFORM=gov.nih.nci.security.cgmm.authenticators.CaGridFormAuthenticator.
* In the org/apache/catalina/authenticators/mbeans-descriptors.xml file, add **mbean CaGridFormAuthenticator** with type gov.nih.nci.security.cgmm.authenticators.CaGridFormAuthenticator.

## Integration Steps

More information regarding integrating CGMM with an existing application that uses form-based container-managed security is available in the Appendices of this document.

on page 63 details the steps for a reference implementation of a formsecurity.war application.

on page 66 provides specific steps for caArray-CGMM container-managed security integration.

# CGMM Installation and Deployment

This chapter provides details regarding the contents of the CGMM release.

Topics in this chapter include:

* on page 42.
* on page 43.
* on page 46.
* on page 47.

Figure 6‑1 shows a diagram of a CGMM deployment and is provided as a reference for the information provided throughout the rest of this chapter.



Figure ‑ CGMM Deployment Diagram

NOTE: In order for the CGMM Tool to function properly, the environment setup detailed in the section of this chapter has to be made available.

## Release Contents

The CGMM is released both 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, designed to 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 the CGMM.zip file found on the NCICB GForge website in the Security projects File Tab: [https://gforge.nci.nih.gov/frs/?group\_id=12](https://gforge.nci.nih.gov/frs/?group_id=12%20) .

The CGMM Release contents include the files listed and described in the following table:

|  |  |
| --- | --- |
| 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 to 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 ‑ CGMM Release Contents

## Installation Pre-Requisites

The installation pre-requisites described in the sections that follow must be performed before the CGMM Tool can be installed.

### Refactoring Host Application (Default Behavior)

The Host application must implement the following:

1. Add CGMM Filter to intercept all User requests. Shown below is the Web.xml configuration needed to add CGMM Filter.

<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>

1. Identify the cgmm-properties.xml configuration details for Host information section. A sample configuration is shown below:

*<*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>

Refer to *,*  on page 57 for more information about this file. Refer also to the cgmm-properties.xsd shown in on page 51 for more details about each configuration element.

### Configure Container Managed Security (Alternate Behavior)

The Host application must implement the following:

1. Add Custom Form based Authentication configuration to Web.xml. Shown below is a configured sample web.xml file.

<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>

Figure ‑: Sample Web.xml configuration to Custom Form based Authentication

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

A sample configuration is shown in , on page 57. Refer also to the cgmm-properties.xsd shown in on page 51 for more details about each configuration element.

1. Add Mail Service configuration details for the 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.institute.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

Use the steps outlined below to configure the caGrid Security Infrastructure.

1. Identify the Authentication Service(s) that will be used for authenticating 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. For more details, see the sample configuration file provided in on page 59.
4. Identify the cgmm-properties.xml configuration details for Authentication Service and Dorian Service information.

A sample configuration is shown below. Refer also to the cgmm-properties.xsd shown in on page 51 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>

### Identify Configuration Parameters for CGMM

Determine if the new caGrid User Creation feature of the CGMM Tool is desired.

If the new caGrid User Creation feature is to be disabled, configure the cgmm-information section of the cgmm-properties.xml file 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 the host application context relative URI.

If the Alternate behavior is enabled or set to true, configure the host-information section of the cgmm-properties.xml file 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 that the following environment and configuration conditions are met. The software and access credentials/parameters are required.

Host Application Environment

* JBoss 4.0 Application Server.
* MySQL v4.0 or higher 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

Before deploying CGMM, verify that the installation prerequisites have been completed and that the deployment checklist is complete.

Step 1: Deploy cgmmweb.war file

Copy the cgmmweb.war file into the deployment directory of JBoss, located 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 into the deployment directory of JBoss, located 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. A sample configuration is shown below:

<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>

Step 4: Configure SyncGTS

Configure the URLs for Slave/Master GTS. Refer also to *,*  on page 59.

Step 5: Configure the CGMM Properties File

For a description of the elements, see *,*  on page 51.

Example:

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

Step 6: Configure the CSM Application Security Configuration File

Configure ApplicationSecurityConfig.xml as follows:

* Change the <context-name> element to the Host application context name. For example:

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

* Change the <hibernate-config-file> element to point to the Hibernate configuration file. For example:

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

In the <<hostApplicationName>>.hibernate.cfg.xml file, 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 file of Jboss as shown below. This entry configures a login-module against the host application context.

<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>

The location of this file is: JBOSS\_HOME/server/default/conf/login-config.xml.

Alternatively, the JAAS configuration can be done via the cgmm.login.config configuration file by performing the following:

* Rename the cgmm.login.config file to the value specified System property gov.nih.nci.security.cgmm.login.config.file.
* Modify the login.config name to the Host Application Name.
* Point to the 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 the JBoss Mail Service, add the configuration shown in the sample below to the JBOSS\_HOME/server/default/deploy/mail-service.xml file:

<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 the log4j.xml file. Be sure to replace the entries for Application Name, Server Name, Port, Schema Name, DB User, and Password with the appropriate values.

<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

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.

Once the Jboss server has completed deployment, open a browser to access the host applications secured login page. The URL is:

http://<<jboss-server>>/<<host\_application\_context>>

Where <<jboss-server>> is the IP or the DNS name of Jboss Server and <<host\_application\_context>> is the context name of the host application.

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 CGMM FAQ page of the CSM Wiki located at: <https://wiki.nci.nih.gov/x/4wBB>.

1. CGMM Properties XSD File

<?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 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>

1. 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>

1. 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>

1. CGMM with Reference Implementation

The steps provided in this appendix install the reference implementation cgmmHostWeb web application along with the cgmmweb web application. Using these steps you can set up a test environment to demonstrate how the CGMM Tool works with an existing Host application. The internal details of the CGMM Tool are beyond the scope of this guide. Refer the CGMM Design Document for more details.

The steps provided here have been tested and will work as long as the steps are followed correctly.

NOTE: The paths and values used in the commands and configuration files are for example only.

1. Verify that caGrid 1.2 is installed. If caGrid 1.2 is not installed, install caGrid 1.2 using the caGrid Installer 1.2 (install the software only; no services are needed).
2. Verify that the environment variables ANT\_HOME, JAVA\_HOME, CAGRID\_HOME, and GLOBUS\_LOCATION are set. You can do this by typing the following commands at the command prompt, pressing Enter 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 was installed>>

export CAGRID\_HOME;

1. Verify caGrid 1.2 is configured to point to the Training Grid 1.2 by typing the following commands at the command prompt, pressing Enter after each statement:

Cd $CAGRID\_HOME

ant –Dtarget.grid=training-1.2 configure

1. Run SyncGTS by typing the following commands at the command prompt, pressing Enter after each statement:

Cd $CAGRID\_HOME/projects/syncgts

ant syncWithTrustFabric

1. Obtain a Host Certificate for the machine. This is a pre requisite. Instructions for obtaining Host Credentials (certificate) are available at the following 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 the JBoss deployment folder: {jboss-home}/server/default/deploy/.
2. Deploy the cgmmweb.war by putting the war file in JBoss de deployment folder: {jboss-home}/server/default/deploy/.
3. Configure the CGMM and Host Application properties.
4. Configure the System Properties by modifying the {jboss-home}/server/default/deploy/properties-service.xml and adding 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 the JAAS Login Configuration Module as follows:

* Rename the cgmm.login.config file to the value specified in the 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 the CSM 4.1 Schema for the sampleHostApplication.

1. Configure the Sync GTS description configuration xml file. This step is required to sync the caGrid Trust Fabric with the Server’s Keystore. Instructions on how to configure the sync-description.xml file are available from the following link:

<http://www.cagrid.org/wiki/GTS:1.2:Administrators_Guide:SyncGTS:Configuration>

In addition, the sample sync-description.xml provided in on page 59 points to the caGrid Training 1.2

1. Configure CGMM Properties file. See on page 51 for a description of the elements in cgmm-properties.xsd. See on page 57 for details of the cgmm-properties.xml file.
2. Configure ApplicationSecurityConfig.xml file as follows:

* Modify the <context-name> to the Host application context name.   
  For example: <context-name>sampleHostApplication</context-name>
* Modify the <hibernate-config-file> element to point to the hibernate configuration file. For 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 as follows:

* Specify the database connection properties in cgmmweb.hibernate.cfg.xml as shown below:

<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 the datasource. The sample JBOSS\_HOME/server/default/deploy/mysql-ds.xml configuration is shown below:

<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>

1. Testing CGMM Container Managed Security Integration

The steps provided in this appendix are sample software setup steps for testing the integration of CGMM’s container-managed security for a reference implementation.

Because these steps test against a configured reference implementation with access to the caGrid 1.2 Training grid, you must refer to beginning on page 60 and perform Steps 1-5 before continuing with the steps provided below.

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

1. Copy the following jars to the JBOSS\_HOME\server\default\deploy\jbossweb-tomcat55.sar folder:

* CGMM\_RELEASE\_FOLDER/cgmmapi.jar
* CGMM\_RELEASE\_FOLDER/catalina.jar (NOTE: This is custom catalina.jar.)
* CGMM\_RELEASE\_FOLDER/jbossweb-tomcat55-sar-jars/\*.jar

1. Deploy the CGMM\_RELEASE\_FOLDER/reference\_implementation/formsecurity.war to the JBOSS\_HOME/server/default/deploy folder.
2. Modify the file 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 the file JBOSS\_HOME/server/default/conf/login-config.xml and 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 by performing the following steps:
2. In the JBOSS\_HOME/server/default/deploy/properties-service.xml file, verify that the following properties are set, being sure to specify the correct path for each:

* gov.nih.nci.security.cgmm.syncgts.file = PATH\_TO\_jboss-4.0.5.GA/server/default/cgmm\_config/sync-description.xml
* gov.nih.nci.security.cgmm.properties.file = PATH\_TO\_jboss-4.0.5.GA/server/default/cgmm\_config/cgmm-properties.xml
* gov.nih.nci.security.configFile = PATH\_TO\_jboss-4.0.5.GA/server/default/cgmm\_config/ApplicationSecurityConfig.xml
* gov.nih.nci.security.cgmm.login.config.file = PATH\_TO\_jboss-4.0.5.GA/server/default/cgmm\_config/cgmm.login.config

1. Modify the database connection properties in cgmmweb.hibernate.cfg.xml
2. Modify the ApplicationSecurityConfig.xml to point to the correct application name. Our sample uses ‘sampleHostApplicationName’; this name should match the one shown in the CSM Schema.
3. Modify cgmm.login.config and verify the connection properties and the Application Policy name. Our sample uses ‘sampleHostApplicationName’; this name should match the one shown in the CSM Schema.
4. Create and prime CSM 4.1 Schema by performing the following steps:
5. Modify the sample script and change the following

* 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.
* Search for root and replace it with your database user name for MySQL.
* Search for H/2qIBdj9TQ= and replace it with an encrypted value of the MySQL password of the database user.

1. Execute the db script.
2. Configure the JBoss Mail Service by modifying the JBOSS\_HOME/server/default/deploy/mail-service.xml file, and add the following entry, using 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. Test the configuration by performing the following steps:
3. Start JBoss.
4. Access the URL: <http://localhost:8080/formsecurity/protected/>.
5. When you are prompted for them, enter valid caGrid credentials.

A successful login indicates that your configurations and setup were done correctly.

1. Integrating CGMM Container Managed Security with caArray

The steps provided in this appendix are sample software steps for integrating the caArray application with CGMM container-managed security. As such, the paths and values used in the commands and configuration files are for example only.

Because these steps are performed against a configured reference implementation with access to the caGrid 1.2 Training grid, you must refer to beginning on page 60 and perform Steps 1-5 before continuing with the steps provided below.

1. Check out caArray Trunk source folder.
2. Modify the caarray.war/WEB-INF/pages/login.jsp file 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>

Figure ‑: caarray.war/WEB-INF/pages/login.jsp

1. Modify the <policy> section of the caarray.ear/META-INF/security-config.xml file so that it appears as shown below:

<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. Deploy the caArray Application.
2. Check out CGMM or Download the CGMM Release.
3. Build CGMM (only necessary if source code was checked out) by going to the cgmmweb directory (i.e., CGMM\_FOLDER) and typing the following command at the command prompt:

ant –f cgmm\_build.xml

The Build contents are now available in the CGMM\_RELEASE\_FOLDER folder.

1. Copy the following jars to the JBOSS\_HOME\server\default\deploy\jbossweb-tomcat55.sar folder:

* CGMM\_RELEASE\_FOLDER/cgmmapi.jar
* CGMM\_RELEASE\_FOLDER/catalina.jar (NOTE: This is a custom jar file.)
* CGMM\_RELEASE\_FOLDER/jbossweb-tomcat55-sar-jars/\*.jar

Please make sure to update any versions of jars relevant to caArray to avoid conflicts with the caArray application.

1. Copy all CGMM\_RELEASE\_FOLDER/jboss\_default\_libs/\*.jar files to the JBOSS\_HOME\server\default\lib folder.
2. Configure CGMM by performing the following steps:
3. In the JBOSS\_HOME/server/default/deploy/properties-service.xml file, verify that the following properties are set, being sure to specify the correct path for each:

* gov.nih.nci.security.cgmm.syncgts.file = PATH\_TO\_jboss-4.0.5.GA/server/default/cgmm\_config/sync-description.xml
* gov.nih.nci.security.cgmm.properties.file = PATH\_TO\_jboss-4.0.5.GA/server/default/cgmm\_config/cgmm-properties.xml
* gov.nih.nci.security.configFile = PATH\_TO\_jboss-4.0.5.GA/server/default/cgmm\_config/ApplicationSecurityConfig.xml
* gov.nih.nci.security.cgmm.login.config.file = PATH\_TO\_jboss-4.0.5.GA/server/default/cgmm\_config/cgmm.login.config

1. Modify the database connection properties in cgmmweb.hibernate.cfg.xml.
2. Modify the ApplicationSecurityConfig.xml file to point to the correct application name. For caArray, the application name is caarray and should match the name shown in the CSM Schema.
3. Modify cgmm.login.config and verify the connection properties and the Application Policy name. For caArray, this is caarray and should match the name shown in the CSM Schema.
4. Database Setup:
5. Make sure there is at least one migrated user with admin roles associated. For example: caarrayadmin.
6. Replace caarrayadmin in the csm\_user.login\_name column of the CSM User table with a valid caGrid ID. Make sure there is at least one migrated user with admin roles associated.
7. Configure the JBoss Mail Service by modifying the JBOSS\_HOME/server/default/deploy/mail-service.xml file, and add the following entry, using 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. Test the configuration by performing the following steps:
3. Start JBoss.
4. Access the URL: http://<server:port>/caarray.
5. When the page appears, click Login on the left side..
6. When you are prompted for them, enter valid caGrid credentials.

A successful login indicates that your configurations and setup were done correctly.

1. Installing CGMM in existing Container using Command line installer

The steps provided in this appendix are sample software steps for integrating the caArray application with CGMM container-managed security. As such, the paths and values used in the commands and configuration files are for example only.

Because

Glossary

The following table contains a list of terms used in this document along with their 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. |
| 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 |
| 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. |
| IDP | Identity Provider. Is also sometimes shown as “IdP”. For more information, see <http://asc.gsa.gov/portal/template/faq08.vm>. |
| 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. |
| WSDD | An acronym for Web Service Deployment Descriptor, which can be used to specify resources that should be exposed as Web Services. See http://ws.apache.org/axis/java/user-guide.html#CustomDeploymentIntroducingWSDD for more information. |
| WSDL | An acronym for Web Services Definition Language, which is an XML-based language that provides a model for describing Web services. See http://www.w3.org/TR/wsdl.html or http://en.wikipedia.org/wiki/WSDL for more information. |
| XSD | XML Schema Definition. |

Index

A

alternate CGMM scenarios, 34

API

authenticating users, 17

CGMM, 7, 11

CGMM Manager, 12

configuration files, 18

importing authentication, 16

importing CGMM Manager, 16

migrating users, 17

obtaining authentication, 17

obtaining CGMM Manager, 17

services, 12

workflow, 11

audit logging, 20

authenticating users, 17

authentication, 6

B

before you install, 45

C

caGrid account

create new, 7, 27, 36

caGrid security infrastructure, 47

CGMM

API, 7, 11

API configuration, 18

API services, 12

architecture, 6

components, 7

customization, 7

deployment, 43, 48, 49

filter, 7

installation, 43, 45

installation parameters, 48

overview, 5

process flow, 7

release contents, 44

security concepts, 8

CGMM Manager class, 12

CGMM Manager Service

audit logging, 20

CGMM Properties sample file, 59

CGMM Properties XSD file, 53

CGMM Tool, 23

alternate behavior, 33

alternate workflow, 34

customizing, 39

default behavior, 23

default workflow, 24

overview, 8

clm.jar file, 20

Common Logging API, 20

common logging database, 21

configuration files, 18

create caGrid account, 7, 27, 36

create CSM account, 33

customize CGMM Tool, 39

D

database

for logging, 21

default CGMM scenarios, 24

deploying CGMM, 43, 48, 49

deploying logging, 22

Dorian, 8

E

email caGrid account request, 37

event logging, 20

F

filter intercept, 7

G

GAARDS

authentication, 5

components used, 6

Glossary, 72

grid trust fabric

synching, 18

H

host application

authentication, 6

environment requirements, 48

installation pre-requisites, 45

integrating with API, 16

issues solved, 6

login after account request, 38

login after migration, 27, 30, 36

migration filter, 6

refactoring, 45

HTTP filter, 7

I

identity provider, 8

importing authentication API, 16

installing CGMM, 43, 45

integrating the API, 11, 16

J

JAAS deployment, 50

JBDC Appender, 21

JBoss

configure for logging, 22

JBoss deployment, 50, 52

L

log4j file entry, 21

loggers, 20

logging

events, 20

M

migrate

CSM account, 26, 29, 35

to existing Grid account, 31

to new Grid account, 29

without CSM account, 33

migrating users, 17

migration process, 7, 24, 34

minimum requirements, 9

O

obtaining authentication API, 17

overview

CGMM, 5

CGMM Tool, 8, 23

configuration files, 18

R

related documents, 2

release contents, 44

release schedule, 4

request new caGrid account, 37

S

sample Sync description file, 61

security caGrid infrastructure, 47

security concepts, 8

submit support issue, 4

SyncGTS, 8, 40

SyncGTS servlet, 11, 18

synching with trust fabric, 18

U

user login

CSM account, 34

after caGrid account request, 38

after migration, 27, 30, 36

caGrid account, 30

CSM account, 25, 27, 36

user migration process, 24, 34

user provisioning, 5

using caGrid login, 25, 30, 35

using CSM login, 25, 27, 34, 36

W

workflow for API integration, 11

workflow for CGMM tool, 24, 34