caBIG™ Clinical Trials Suite 1.0

Installation Guide

caBIG™ cancer Biomedical Informatics Grid™

an initiative of the National Cancer Institute

February 21, 2008

Copyright and License

Copyright Notice. Copyright 2008 National Cancer Institute ("caBIG™ Participant"). CCTS was created with NCI funding and is part of the caBIG™ initiative. The software subject to this notice and license includes both human readable source code form and machine readable, binary, object code form (the "caBIG™ Software").

This caBIG™ Software License (the "License") is between caBIG™ Participant and You. "You (or "Your") shall mean a person or an entity, and all other entities that control, are controlled by, or are under common control with the entity. "Control" for purposes of this definition means (i) the direct or indirect power to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

License. Provided that You agree to the conditions described below, caBIG™ Participant grants You a non-exclusive, worldwide, perpetual, fully-paid-up, no-charge, irrevocable, transferable and royalty-free right and license in its rights in the caBIG™ Software, including any copyright or patent rights therein that may be infringed by the making, using, selling, offering for sale, or importing of caBIG™ Software, to (i) use, install, access, operate, execute, reproduce, copy, modify, translate, market, publicly display, publicly perform, and prepare derivative works of the caBIG™ Software; (ii) make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of caBIG™ Software (or portions thereof); (iii) distribute and have distributed to and by third parties the caBIG™ Software and any modifications and derivative works thereof; and (iv) sublicense the foregoing rights set out in (i), (ii) and (iii) to third parties, including the right to license such rights to further third parties. For sake of clarity, and not by way of limitation, caBIG™ Participant shall have no right of accounting or right of payment from You or Your sublicensees for the rights granted under this License. This License is granted at no charge to You. Your downloading, copying, modifying, displaying, distributing or use of caBIG™ Software constitutes acceptance of all of the terms and conditions of this Agreement. If you do not agree to such terms and conditions, you have no right to download, copy, modify, display, distribute or use the caBIG™ Software.

1. Your redistributions of the source code for the caBIG™ Software must retain the above copyright notice, this list of conditions and the disclaimer and limitation of liability of Article 6 below. Your redistributions in object code form must reproduce the above copyright notice, this list of conditions and the disclaimer of Article 6 in the documentation and/or other materials provided with the distribution, if any.

2. Your end-user documentation included with the redistribution, if any, must include the following acknowledgment: “This product includes software developed by National Cancer Institute.” If You do not include such end-user documentation, You shall include this acknowledgment in the caBIG™ Software itself, wherever such third-party acknowledgments normally appear.
3. You may not use the names “The National Cancer Institute”, “NCI”, “Cancer Bioinformatics Grid” or “caBIG™” to endorse or promote products derived from this caBIG™ Software. This License does not authorize You to use any trademarks, service marks, trade names, logos or product names of either caBIG™ Participant, NCI or caBIG™, except as required to comply with the terms of this License.
4. For sake of clarity, and not by way of limitation, You may incorporate this caBIG™ Software into Your proprietary programs and into any third party proprietary programs. However, if You incorporate the caBIG™ Software into third party proprietary programs, You agree that You are solely responsible for obtaining any permission from such third parties required to incorporate the caBIG™ Software into such third party proprietary programs and for informing Your sublicensees, including without limitation Your end-users, of their obligation to secure any required permissions from such third parties before incorporating the caBIG™ Software into such third party proprietary software programs. In the event that You fail to obtain such permissions, You agree to indemnify caBIG™ Participant for any claims against caBIG™ Participant by such third parties, except to the extent prohibited by law, resulting from Your failure to obtain such permissions.
5. For sake of clarity, and not by way of limitation, You may add Your own copyright statement to Your modifications and to the derivative works, and You may provide additional or different license terms and conditions in Your sublicenses of modifications of the caBIG™ Software, or any derivative works of the caBIG™ Software as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
6. THIS caBIG™ SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES (INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE) ARE DISCLAIMED. IN NO EVENT SHALL THE NATIONAL CANCER INSTITUTE OR ITS AFFILIATES BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS caBIG™ SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Credits and Resources

caBIG Clinical Trials Suite (CCTS)		
Development	***Technical Documentation***	***Program Management***
	Paula Brown[7]	Edmund Mulaire[4]
	John Koisch[2]	Charlie Mead[2]
	Patrick McConnell[3]	Patrick McConnell[3]
	Manav Kher[4]	Juli Klemm[8]
	Edmund Mulaire[4]	
	Jill Hadfield[8]	
Key Contributors: Vijaya Chadaram[3], Sharon Elcombe[5], Jomol Mathew[6]		
[1] Scenpro	[2] Booz Allen Hamilton	[3] Duke Comprehensive Cancer Center
[4] SemanticBits	[5] Mayo Clinic	[6] Dana Farber Cancer Center
[7] Ekagra	[8]National Cancer Institute Center for Bioinformatics	

Cancer Adverse Events Reporting System (the AE Reporting System)		
Development	***Technical Documentation***	***Program Management***
Vinay Kumar[5]	Jennifer Reed[5]	Christo Andonyadis, D.Sc.[1]
Srini Akkala[5]		Lisamarie Schick[4]
Biju Joseph[5]		Edmund Mulaire[4]
Krikor Krumlian[6]		Ram Chilukuri[5]

Cancer Adverse Events Reporting System (the AE Reporting System)		
Development	***Technical Documentation***	***Program Management***
Rhett Sutphin[7]		
Jared Flatow[7]		
Sean Whitaker[7]		
Karthik Iyer[5]		
Key Contributors: Sharon Elcombe[8], Sonja Hamilton[8], Jean Hanson[8], Jennifer Frank[8], Larry Esser[8], Bob Morell[9], Ken Livengood[9], Steven Cheng[9], Warren Kibbe[7], Cal Collins[6], David Patton[7], Renee Webb[7], Ann Setser[2], Anne Tompkins[3]		
[1] National Cancer Institute Center for Bioinformatics (NCICB)		[2] NCI - Cancer Therapy Evaluation Program (CTEP)
[3]NCI - Department of Cancer Prevention (DCP)	[4]Booz Allen Hamilton	[5]SemanticBits
[6]Akaza Research	[7]Northwestern University	[8]Mayo Clinic
[8]Wake Forest University		

Cancer Central Clinical Participant Registry (C3PR)		
Development	***Technical Documentation***	***Program Management***
Kruttik Aggarwal[2]	Kruttik Aggarwal[2]	Robert Annechiarico[1]
Manav Kher[2]	Manav Kher[2]	Ram Chilukuri[2]
Ram Chilukuri[2]		Jamie Cuticchia[1]
Vinay Gongoli[2]		Patrick McConnell[1]
Ramakrishna Gundala[2]		
Key Contributors: Bob Morell[3], Steven Cheng[3], Robert Annechiarico[1], William Banks[1], Peter Le[1]		
[1] Duke Comprehensive Cancer Center		[2] SemanticBits
[3]Wake Forest University		

caXchange		
Development	***Technical Documentation***	***Program Management***
Kalpesh Patel[2]	Kalpesh Patel[2]	Michael Holck[1]
Amit Srivastava[2]	Amit Srivastava[2]	Kalpesh Patel[2]
Yogaraj Khanal[2]	Yogaraj Khanal[2]	Patrick McConnell[3]
Steve Chernyak[2]	Steve Chernyak[2]	
Key Contributors:		
[1] Scenpro	[2] Ekagra	

LabViewer		
Development	***Technical Documentation***	***Program Management***
Michael Holck[1]	Michael Holck[1]	Michael Holck[1]
Anupama Sharma[1]	Anupama Sharma[1]	
Joe Zhou[1]	Joe Zhou[1]	
Key Contributors: Becky Angeles[1], David Dorman[1], Bill McCurry[1]		
[1] Scenpro		

Patient Study Calendar		
Development	***Technical Documentation***	***Program Management***
Rhett Sutphin[1]	Rhett Sutphin[1]	Warren Kibbe[1]
Saurabh Agrawal[2]	Saurabh Agrawal[2]	Sean Whitaker[1]
John Dzak[1]	John Dzak[1]	Ram Chilukuri[2]
Nataliya Shurupova[1]	Nataliya Shurupova[1]	
Key Contributors: Renee Webb[1], David Patton[1], Maggie Patton[1], Sharon Elcombe[3], Jean Hanson[3], Nicole Knutson[3], Cindy Sleeper[3], Jack London[4], Bob Morell[5], Steve Cheng[5], Paul Courtney[6], Robert Dennis[7], Christopher Denny[7], Virginia Hetrick[8], Cal Collins[9], Cathy Reniere[10], Courtney Haverdink[11], Umit Topaloglu[12], Kristel Dobratz[13], Michael Schauer[14], Patrick McConnell[14], Lisamarie Schick[15], Niket Parikh[15], John Koish[15], David Wu[15]		
[1]Northwestern University	[2]SemanticBits	[3]Mayo Clinic
[4] Thomas Jefferson University	[5]Wake Forest University	[6]Dartmouth University
[7]University of California at Los Angeles	[8]Patient Advocate	[9]Akaza Research

Patient Study Calendar		
Development	***Technical Documentation***	***Program Management***
[10]University of Virginia	[11]Siouxland Hematology Oncology Associates	[12]University of Arkansas
[13]Vocabularies & Common Data Elements (VCDE) Mentor, Gulf Stream Bioinformatics	[14]Duke Comprehensive Cancer Center	[15]Booz Allen Hamilton

Cancer Central Clinical Database (C3D) Connector		
Development	***Technical Documentation***	***Program Management***
Patrick Conrad[1]	Patrick Conrad[1]	Charles Griffin[1]
Charles Griffin[1]	Charles Griffin[1]	
Key Contributors:		
[4] Ekagra		

Contacts and Support	
NCICB Application Support	http://ncicbsupport.nci.nih.gov/sw/ Telephone: 301-451-4384 Toll free: 888-478-4423

LISTSERV Facilities Pertinent to Software Teams		
LISTSERV	***URL***	***Name***
CCTS--General	ccts-general@gforge.nci.nih.gov	CCTS General Forum
CCTS--Support	ccts-support@gforge.nci.nih.gov	CCTS Support Forum
CCTS--Technical	ccts-technical@gforge.nci.nih.gov	CCTS Technical Forum

TABLE OF CONTENTS

Copyright and License i

Credits and Resources iii

caBIG™ Clinical Trials Suite 1.0 Installation Guide 1

1. Background 1

1.1 Scope 2

1.2 Deployment Architecture 2

1.3 Production Deployment 3

1.3.1 Production Consolidated HSM Deployment *4*

1.3.2 Production Consolidated VM Deployment *5*

1.3.3 Test Consolidated VM Deployment *6*

2. Prerequisites 6

2.1 Hardware Requirements 6

2.2 Software Requirements 7

2.3 caGrid Requirements 7

3. Pre-installation 8

3.1 Deployment Planning 8

3.2 Install Prerequisites 12

3.3 Setting Environment Variables 12

3.3.1 Setting Environment Variables on Linux Servers *12*

3.3.2 Setting Environment Variables on a Windows Server *13*

4. caGrid Installation 14

4.1 GUI Emulator for Linux Servers 14

4.2 Dorian 14

4.2.1 Launch the caGrid 1.1 Installer *14*

4.2.2 Install Dorian *15*

4.2.3 Start Dorian *18*

4.2.4 Bind Host Credentials to the New User *27*

4.3 Create Additional Host Certificates 29

4.4 GTS 31

4.4.1 Copy GTS Certificates *31*
4.4.2 Launch the caGrid 1.1 Installer *31*
4.4.3 Install GTS *32*
4.4.4 Copy the GTS Root CA File *34*
4.4.5 Copy the Dorian CA File *34*
4.4.6 Start GTS/SyncGTS *35*
4.5 Complete the Trust Level Setup 35
4.5.1 Create Trust Levels *35*
4.5.2 Add Dorian to the Trust Fabric *37*
4.5.3 Grant Dorian CRL Publication Rights *38*
4.6 CDS 38
4.6.1 Download CDS *39*
4.6.2 Build CDS *39*
4.6.3 Obtain CDS Certificates *39*
4.6.4 Setup CDS Trust *39*
4.6.5 Deploy Globus *39*
4.6.6 Secure Tomcat *40*
4.6.7 Create a Globus Security Descriptor *41*
4.6.8 Finalize Tomcat Security *42*
4.6.9 Configure CDS *42*
4.6.10 Start CDS *43*
4.7 IndexService and GME 43
4.7.1 Launch the caGrid 1.1 Installer *43*
4.7.2 Install the IndexService and GME *44*
4.8 Secure Application Containers 46
4.8.1 Secure Application Tomcats *46*
4.8.2 Deploy SyncGTS *46*
4.9 WebSSO 46
4.9.1 Build WebSSO *46*
4.10 Inject the Suite UI 46
4.10.1 Deploy WebSSO *46*
5. Suite Installation 47
5.1 Cancer Adverse Event Reporting System (caAERS) 47
5.1.1 Component Overview *47*
5.1.2 Component Installation *47*
5.2 Cancer Centralized Clinical Database (C3D) Grid Component 48
5.2.1 Component Overview *48*
5.2.2 Component Installation *48*
5.3 Cancer Central Clinical Participant Registry (C3PR) 48

5.3.1 Component Overview *48*
5.3.2 Component Installation *49*
5.4 caXchange 49
5.4.1 Component Overview *49*
5.4.2 Component Installation *49*
5.5 LabViewer 49
5.5.1 Component Overview *49*
5.5.2 Component Installation *50*
5.6 Patient Study Calendar (PSC) 50
5.6.1 Component Overview *50*
5.6.2 Component Installation *50*
6. Configuration 50
6.1 Advertisement (Optional) 50
6.1.1 Configuring Deployed Services *50*
6.1.2 Configuring Undeployed Services *51*
6.2 Schema Registration (Optional) 51
6.3 caXchange 53
6.4 Inter-Application Linking (Hotlinks) 53
7. Validating the Installation 53
8. Troubleshooting 54
8.1 GSS-API Bad Certificate Error 54
8.1.1 Error *54*
8.1.2 Solution *54*
9. Glossary 55
10. Supplementary Resources 56

caBIG™ Clinical Trials Suite 1.0 Installation Guide

1. Background

The caBIG™ Clinical Trial Suite (hereafter referred to as “the Suite”) project provides key software components that aid the distributed healthcare team in curing incidences of cancer in patients. It provides software components that support computable semantic interoperability between and among the various distributed systems involved in conducting clinical trials of cancer-fighting protocols. It is organized as part of the Cancer Bio-Informatics Grid (caBIG™) project sponsored by the National Cancer Institute (NCI).

Primarily, the Suite architecture integrates clinical trial software by providing a robust platform to share semantically annotated data sets. It leverages multiple technologies including caBIG™ infrastructural components (e.g., caGrid Security), NCICB technologies (for example, Common Security Module), as well as other CTMS

workspace implementations. *Figure 1* display a high-level view of the Suite architecture.

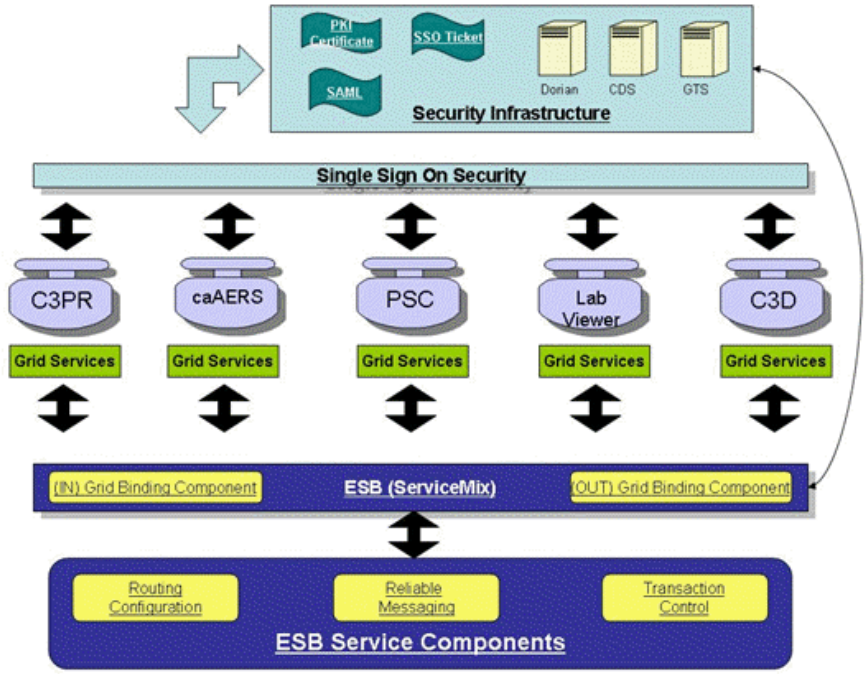

Figure 1 High-level view of the Suite architecture

1.1 Scope

This document is geared towards enterprise system administrators who will be deploying the various caGrid, Suite, and supporting software components. It is an extremely complex installation procedure, and thus a solid understanding of enterprise security (certificates, certificate authorities, certificate signing, certificate-based trust), web and grid applications (Tomcat, Globus, Tomcat logs), application building (Java, ant), databases (MySQL, Postgres, and Oracle), and hardware (operating systems, daemon processes, and server processes) is critical to a successful installation. The installer should budget between 8 and 40 hours for the installation procedure depending on expertise and environmental considerations.

1.2 Deployment Architecture

The Suite is built upon a componentized architecture and thus there are a number of deployment options depending on the components being deployed, levels of security being employed, load to the system, and access to the system. For this reason, four different deployment scenarios are provided that should meet most cancer center needs while providing a spectrum of options to help guide custom deployment architectures. The following sections describe the recommended options.

Notes:

- Wherever Virtual Machine (VM) is identified, physical machines can be used. Wherever physical machines are identified, VM will not suffice.
- All databases are consolidated on a single node in each deployment, which is not a requirement. Databases can be spread across any number of nodes.

- All web applications and application grid services are consolidated on a single node, which is not a requirement. The web applications and application grid services can be deployed to any number of nodes.
- Each deployment diagram has the C3D grid service deployed to a separate node. A separate node is not required as long as the Clinical Database grid service is deployed to a machine running the Windows operating system.

SyncGTS is required on each node except the GTS node in order to maintain trust. It is required in one Tomcat container per user account. For example, if you are running each container on a node under the same user account, it only needs to be deployed to one container. If you are running each container on a node under a different user account, SyncGTS will have to be deployed to each container.

1.3 Production Deployment

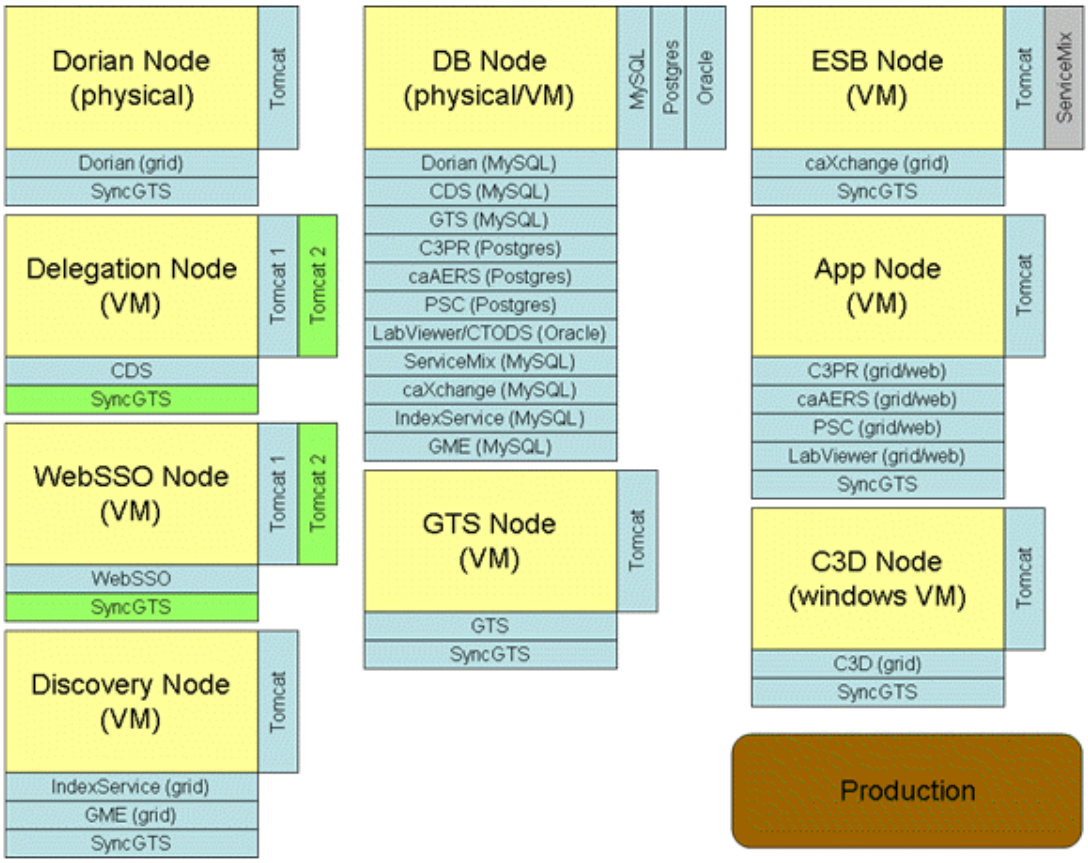

Figure 2 Example deployment diagram for a Production environment

Figure 2 is the most secure deployment and is suitable for an enterprise production environment. Each security component is insulated by being deployed on its own VM. Dorian is deployed on a physical machine for the hardware security module (HSM) to provide credentials to run Dorian. This deployment architecture leverages the most machines and thus may not be realistic for all users.

1.3.1 Production Consolidated HSM Deployment

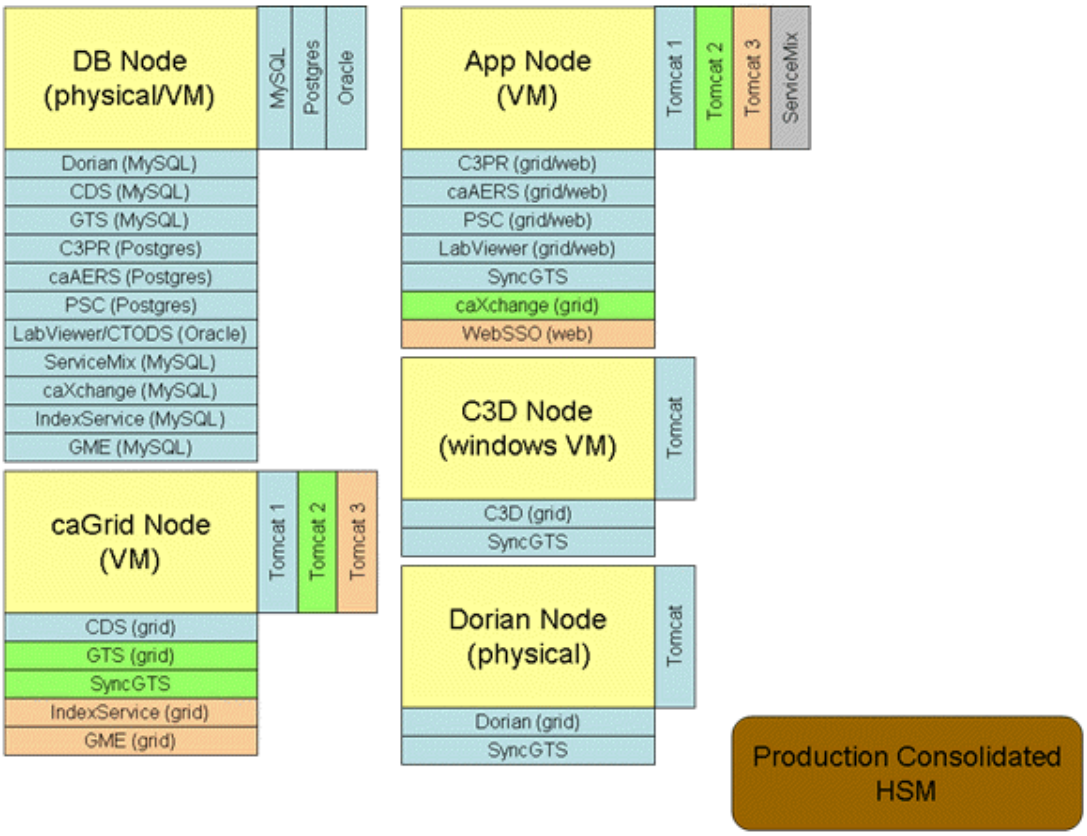

Figure 3 Example deployment diagram for a Production Suite environment where services are consolidated on fewer services but still maintain an appropriate level of security with an HSM

Figure 3 consolidates some of the security services into a single node. Dorian is still maintained on a physical machine in order to provide for HSM-provided credentials. This deployment is most likely to be realistic for production environment using an HSM.

1.3.2 Production Consolidated VM Deployment

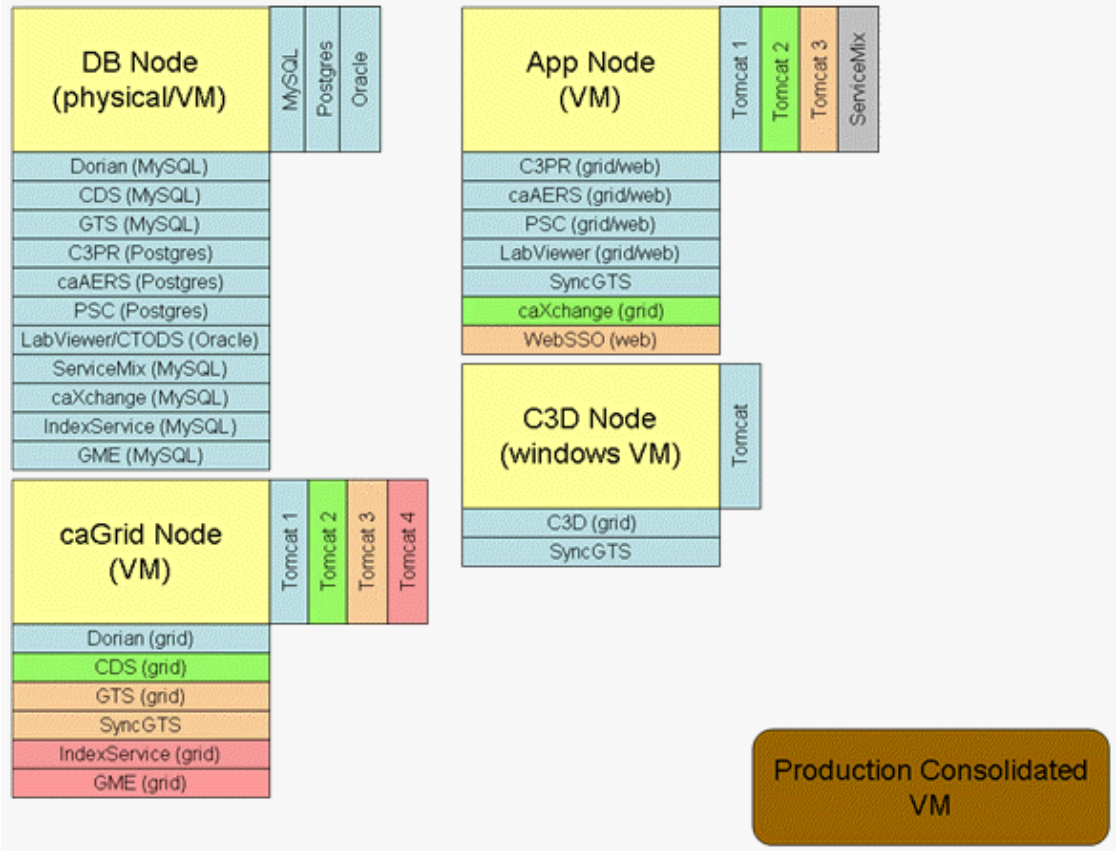

Figure 4 Example deployment diagram for a Production Suite environment where services are consolidated on fewer services but still maintain an appropriate level of security without an HSM.

Figure 4 consolidates some of the security services into a single node, including Dorain (HMS cannot be leveraged). This deployment is most likely to be realistic for production environment not using an HSM.

1.3.3 Test Consolidated VM Deployment

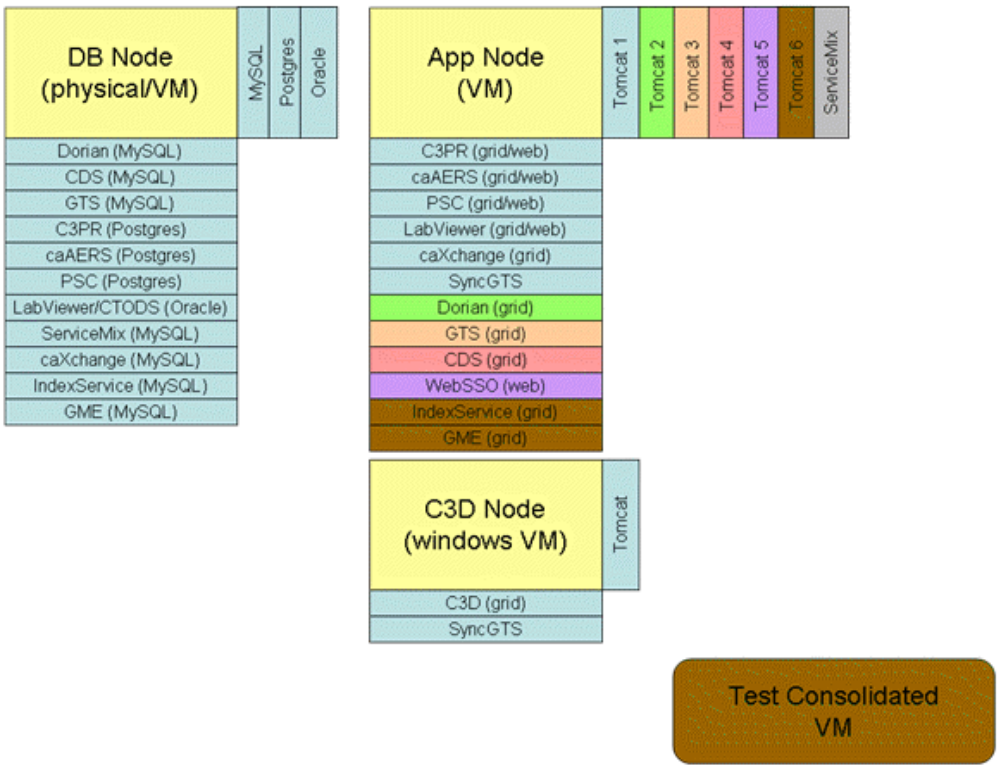

Figure 5 Example deployment diagram for a Test Suite environment. This deployment architecture uses the fewest servers but is not as secure as the other deployment options because core security components are not maintained on separate hardware instances.

Figure 5 consolidates all application and security services into a single node. This is not recommended for production use, but it is acceptable for testing.

2. Prerequisites

2.1 Hardware Requirements

Hardware requirements are variable depending on the deployment architecture, load on the systems, size of databases, etc. *Table 1.1* gives some rough guidelines to help determine hardware requirements. The more consolidated the nodes (that is, the more applications/services deployed per node), the more the hardware requirements should be increased.

Hardware Item	***Requirement***
Processor	2 GHz per node
Disk	20 GB per node
Memory	2 GB per node
Network	1 Mbps
Windows OS (for C3D Grid Service)	Windows Server 2003 Standard Addition

Table 1.1 Suite hardware requirements

Hardware Item	Requirement
OS for other nodes	Any that supports prerequisites

Table 1.1 Suite hardware requirements

2.2 Software Requirements

The following software is required to run the various Suite components. Some of the software is required for all components, and some is required for specific components. See individual installation guides for more details on specific requirements.

Requirement	Version	Components	Link
Java SE	1.5.x SDK	All	http://java.sun.com/j2se/1.5/index.jsp
Ant	1.6.5+	All	http://archive.apache.org/dist/ant/binaries/
MySQL (innodb)	4.1.x	Grid	http://dev.mysql.com/downloads/mysql/4.1.html
Oracle	10g	Databases	http://www.oracle.com/technology/software/products/database/oracle10g/index.html
Postgres	8.2	Databases	http://www.postgresql.org/ftp/binary/
Tomcat	5.0.28	All	http://archive.apache.org/dist/tomcat/tomcat-5/v5.0.28/bin/
ServiceMix	3.1.2	ESB	http://servicemix.apache.org/servicemix-312.html
C3D (Oracle Clinical)	4.5	C3D Grid Service	http://www.oracle.com/industries/life_sciences/clinical.html

Table 1.2 Suite software requirements

2.3 caGrid Requirements

The following caGrid components are required to run the full Suite.

Requirement	Version	Link
Dorian*	caGrid 1.1	https://cabig.nci.nih.gov/workspaces/Architecture/caGrid
GTS*		
SyncGTS		
IndexService**		
GME**		
Web SSO***	caGrid 1.2M1	http://www.cagrid.org/mwiki/index.php?title=WebSSO:1.2:Administrators_Guide
CDS***		

Table 1.3 caGrid requirements

Notes

- *For the Suite, Dorian and GTS must be deployed in separate Tomcat containers. Since both deployments use the CATALINA_HOME environment variable, use different operating system users for each deployment.
- **The IndexService and GME are optional components. They are not required to run the Suite, but may be leveraged in a grid environment for dynamic discovery.
- ***The CDS and Web Single Sign On grid services will be included in the forthcoming caGrid 1.2 release. To accommodate the Suite, a caGrid 1.1 compatible release candidate was developed and released. Web SSO and CDS cannot be deployed in the same Tomcat container as the caGrid 1.1 components.

3. Pre-installation

3.1 Deployment Planning

The first step in installing the Suite is planning the deployment, including which servers will house which services ahead of time. Specifically, you should make note of which services will be deployed to which containers on which servers. This will help you determine login names needed for each server, environment variables to set for each server, what software is needed for each server, the services endpoint URLs once software is installed, and certificates that need to be signed.

Using the deployment diagram in *Figure 4* as an example, a deployment planning sample has been laid out in the tables below. It can be used to provide some guidelines on the software that must be installed on each server.

Server	***Ant***	***Java***	***Tomcat***	***1.1 Grid***	***Globus***	***App SW***	***Data Base***	***Cert(s) RQRD***
Dorian	v	v	v	v	v		v	v
CDS	v	v	v	v	v	v	v	v
Index Service	v	v	v	v	v			
GME	v	v	v	v	v			
GTS	v	v	v	v	v		v	v
SyncGTS	Configure in GTS Tomcat Container in Grid Node Configure in C3D Tomcat Container in Windows Node Configure in Applications Tomcat Container in App Node							
C3D	v	v	v	v	v	v	v	
WebSSO	v	v	v		v	v	v	v
caXchange	v	v	v	v	v	v	v	v
ServiceMix	Configure with caXchange							

Table 1.4 Deployment planning sample

Server	Ant	Java	Tomcat	1.1 Grid	Globus	App SW	Data Base	Cert(s) RQRD
C3PR the AE Reporting System LabViewer PSC	v	v	v		v	v	v	

Table 1.4 Deployment planning sample

If multiple services are to run in separate Tomcat containers on the same server, map out the Tomcat container/port assignments. *Table 1.5* provides an example based on the deployment diagram in *Figure 4*:

Component	Port Assignment by Node	Database
Grid Node		
Dorian	Tomcat:8080 CATALINA_HOME /usr/local/tomcat-5.0.28.dorian	MySQL
CDS	Tomcat:18080 CATALINA_HOME: /usr/local/tomcat-5.0.28.cds	MySQL
GTS	Tomcat:28080 CATALINA_HOME: /usr/local/tomcat-5.0.28.gts	MySQL
SyncGTS	Tomcat:28080 CATALINA_HOME: /usr/local/tomcat-5.0.28.gts	
IndexService/ GME	Tomcat:38080 CATALINA_HOME: /usr/local/tomcat-5.0.28.gme	MySQL
C3D Node		
C3D	Tomcat:8080 CATALINA_HOME /usr/local/tomcat-5.0.28-8080	Oracle (not on DB Node)
SyncGTS	Tomcat:8080 CATALINA_HOME /usr/local/tomcat-5.0.28-8080	
App Node		
WebSSO	Tomcat:28080 CATALINA_HOME /usr/local/tomcat-5.0.28.websso	
caXchange	Tomcat:18080 CATALINA_HOME /usr/local/tomcat-5.0.28.caxchange	MySQL
C3PR	Tomcat:8080 CATALINA_HOME /usr/local/tomcat-5.0.28.app	Postgres

Table 1.5 Deployment example

Component	***Port Assignment by Node***	***Database***
Grid Node		
the AE Reporting System	Tomcat:8080 CATALINA_HOME /usr/local/tomcat-5.0.28.app	Postgres
LabViewer	Tomcat:8080 CATALINA_HOME /usr/local/tomcat-5.0.28.app	Oracle
PSC	Tomcat:8080 CATALINA_HOME /usr/local/tomcat-5.0.28.app	Postgres
SyncGTS	Tomcat:8080 CATALINA_HOME /usr/local/tomcat-5.0.28.app	
DB Node	- Other Databases Not Already Previously Defined	
ServiceMix		MySQL
CSM/CLM		MySQL

Table 1.5 Deployment example

Using this information, further plan out user name environment settings needed for each component to be installed. You should plan environment variables for CATALINA_HOME, GLOBUS_LOCATION, JAVA_HOME, and ANT_HOME. Proper planning prior to installation allows for consistency in how these are defined. Using the example deployment in *Table 1.5*, the following users could be created to support a unique setup of each Tomcat container:

Server	***Userid/Port***	***Environment Variable Settings***
Server	Userid/Port	Environment Variable Settings
Dorian Server 1	tomcat-dor Port: 8080	HOME: /local/home/tomcat-dor CATALINA_HOME: /usr/local/tomcat-dor GLOBUS_LOCATION: /local/home/tomcat-dor/ws-core-4.0.3 JAVA_HOME: /usr/jdk1.5.0_06 ANT_HOME: /usr/local/apache-ant-1.6.5
CDS Server 1	tomcat-cds Port: 18080	HOME: /local/home/tomcat-cds CATALINA_HOME: /usr/local/tomcat-cds GLOBUS_LOCATION: /local/home/tomcat-cds/ws-core-4.0.3 JAVA_HOME: /usr/jdk1.5.0_06 ANT_HOME: /usr/local/apache-ant-1.6.5

Table 1.6 Users created to support unique set up of each Tomcat container

Server	***Userid/Port***	***Environment Variable Settings***
GTS & SyncGTS Server 1	tomcat-gts Port: 28080	HOME: /local/home/tomcat-gts CATALINA_HOME: /usr/local/tomcat-gts GLOBUS_LOCATION: /local/home/tomcat-gts/ws-core-4.0.3 JAVA_HOME: /usr/jdk1.5.0_06 ANT_HOME: /usr/local/apache-ant-1.6.5
IndexService & GME Server 1	Tomcat-idx Port: 38080	HOME: /local/home/tomcat-gme CATALINA_HOME: /usr/local/tomcat-gme GLOBUS_LOCATION: /local/home/tomcat-gme/ws-core-4.0.3 JAVA_HOME: /usr/jdk1.5.0_06 ANT_HOME: /usr/local/apache-ant-1.6.5
C3D & SyncGTS Server 2	tomcat-c3d Port: 8080	HOME: /local/home/tomcat-c3d CATALINA_HOME: /usr/local/tomcat-c3d GLOBUS_LOCATION: /local/home/tomcat-c3d/ws-core-4.0.3 JAVA_HOME: /usr/jdk1.5.0_06 ANT_HOME: /usr/local/apache-ant-1.6.5
WebSSO Server 3	tomcat-sso Port: 28080	HOME: /local/home/tomcat-sso CATALINA_HOME: /usr/local/tomcat-sso GLOBUS_LOCATION: /local/home/tomcat-sso/ws-core-4.0.3 JAVA_HOME: /usr/jdk1.5.0_06 ANT_HOME: /usr/local/apache-ant-1.6.5
caXchange Server 3	tomcat-cax Port: 18080	HOME: /local/home/tomcat-cax CATALINA_HOME: /usr/local/tomcat-cax GLOBUS_LOCATION: /local/home/tomcat-cax/ws-core-4.0.3 JAVA_HOME: /usr/jdk1.5.0_06 ANT_HOME: /usr/local/apache-ant-1.6.5
Service Mix Server 3	smix312	JAVA_HOME: /usr/jdk1.5.0_06 ANT_HOME: /usr/local/apache-ant-1.6.5
C3PR the AE Reporting System LabViewer PSC SyncGTS Server 3	tomcat-app Port: 8080	HOME: /local/home/tomcat-app CATALINA_HOME: /usr/local/tomcat-app GLOBUS_LOCATION: /local/home/tomcat-app/ws-core-4.0.3 JAVA_HOME: /usr/jdk1.5.0_06 ANT_HOME: /usr/local/apache-ant-1.6.5

Table 1.6 Users created to support unique set up of each Tomcat container

There are components of ServiceMix that are needed by caXchange during the install process of caXchange. Ensure that the user defined for caXchange is given read, write, and modify privileges to the ServiceMix folder structure to enable copying of ServiceMix files.

Install Java and Ant on each server prior to beginning any other software install. It is recommended that the same path/location be used on each server to maintain consistency in the installation – for example:

```
JAVA_install path: /usr/jdk1.5.0_06
ANT_install path:  /usr/local/apache-ant-1.6.5
```

This allows for consistency in setting these two environment variables on each server.

3.2 Install Prerequisites

Install all software prerequisites found in *Section 2.2 Software Requirements*, including Java, Ant, Tomcat, Globus, database software, and ServiceMix. Refer to the installation guides of each individual piece of software for more specific instructions.

You do not need to deploy Tomcat for caGrid services deployed using the caGrid 1.1 installer, which include Dorian, GTS, IndexService, and the GME. The caGrid installer will be used for this.

3.3 Setting Environment Variables

The following environment variables should be set for each user on each server:

- `CATALINA_HOME`: base installation of Tomcat
- `GLOBUS_LOCATION`: base installation of Globus
- `JAVA_HOME`: base installation of Java
- `ANT_HOME`: base installation of Ant

The following subsections describe how to set environment variables in different operating systems.

3.3.1 Setting Environment Variables on Linux Servers

To set an environment variable for each time you login, follow these steps:

1. Login to the server.
2. Using vi or some other edit tool, modify the `.bash_profile` file in the home directory for each server to define the environment variables determined during the planning phase.

The following is an example of the `.bash_profile` for tomcat-dor based on the planning details defined above:

```
%> vi .bash_profile

-----------------------------------------------------------
# .bash_profile

# Get the aliases and functions
```

```
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs
JAVA_HOME=/usr/jdk1.5.0_06
ANT_HOME=/usr/local/apache-ant-1.6.5
CATALINA_HOME=/usr/local/tomcat-dor
GLOBUS_LOCATION=/local/home/tomcat-dor/ws-core-4.0.3

PATH=$PATH:$HOME/bin:$JAVA_HOME/bin:$ANT_HOME/
bin:$CATALINA_HOME/bin:$GLOBUS_LOCATION/bin

export PATH JAVA_HOME ANT_HOME CATALINA_HOME GLOBUS_LOCATION
unset USERNAME

-----------------------------------------------------------------
```

Once the `.bash_profile` file has been updated, perform the following to execute the file so that the newly defined environment variables are set:

```
%> source .bash_profile
```

3.3.2 Setting Environment Variables on a Windows Server

On a Windows Server (such as the C3D server):

1. Right click on **My computer > Select Properties**
2. Select the **Advanced** tab.

3. On this window, select the **Environment Variables** button, then select the **New** button. This opens the environment variables window:

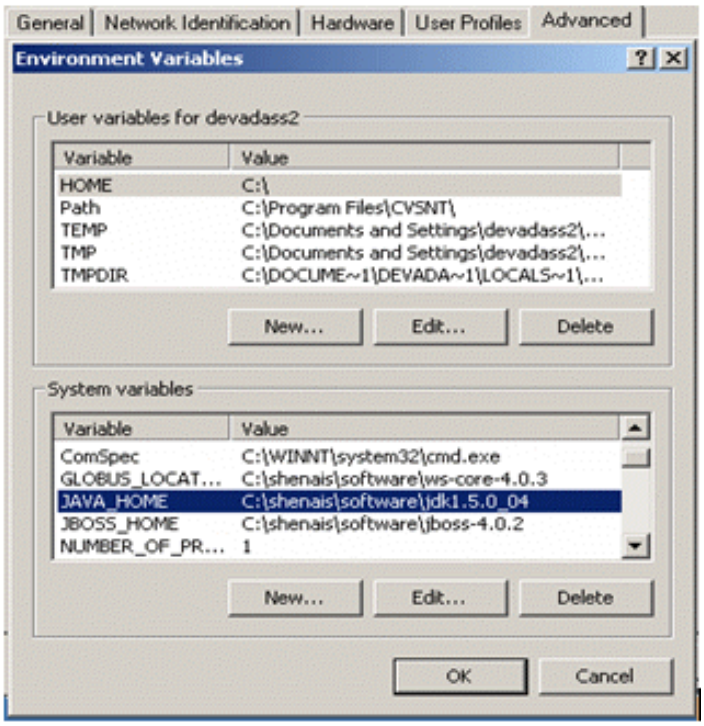

Figure 6 Environment Variables window

4. To define System variables, click **Edit** or **New** on the appropriate variable.

4. caGrid Installation

The caGrid 1.1. installer is used to deploy Dorian and GTS, as well as (optionally)

The IndexService and GME. The following sections describe this process.

4.1 GUI Emulator for Linux Servers

The caGrid installer is a graphical, wizard-like installer for deploying the grid components. For it to work properly on a Linux or Unix-like system, you may have to make special arrangements to graphically display the installer. For example, XMing is a free GUI emulator for Linux Servers (http://sourceforge.net/projects/xming). This software and others may require setting the DISPLAY to export to your local terminal. Consult the documentation for your operating system and GUI emulator software for more information.

4.2 Dorian

4.2.1 Launch the caGrid 1.1 Installer

To download the installer, follow these steps:

1. Log into the server to be installed.

2. At the $HOME directory (`%> cd $HOME`):

```
%> mkdir installer
%> wget http://gforge.nci.nih.gov/frs/download.php/2371/caGrid-1.1-installer-rc11-ncicb.zip
%> cd installer
%> unzip caGrid-1.1-rc1-installer.zip
```

3. From this same directory, launch the installer:

```
%> java -jar caGrid-1.1-installer.jar
```

This brings up a GUI installer.

4.2.2 Install Dorian

The following steps walk you through the procedure for installing the Dorian service into a Tomcat container provided by the installer.

1. Select installation components.
   a. Accept the caGrid license and click **Next**.
   b. Select the **Install caGrid** checkbox. Select the **Install caGrid Services** checkbox. De-select all other check boxes.
   c. Click **Next**.
   d. Select the **Dorian** check box for the Grid Services and click **Next**.
   e. From the **Container Type** drop down select **Tomcat** and click **Next**.
2. Next you will be asked to specify a directory in which to install Ant.
   a. In the **Directory** text field enter {***the path defined for your ANT_HOME variable***} and click **Next**.

      If you followed the instructions and installed all the pre-requisite software prior to beginning the 1.1 Grid install process, ANT will already be installed and the `ANT_HOME` environment variable already set. If this is the case, leave the check box blank and press **Next.**
3. Next you will be asked to specify a directory in which to install Tomcat.
   a. In the **Directory** text field enter {***the path defined for your CATALINA_HOME variable***} and click **Next**.

      If you followed the instructions and installed all the pre-requisite software prior to beginning the 1.1 Grid install process, TOMCAT will already be installed and the `CATALINA_HOME` environment variable already set. If this is the case, leave the check box blank and press *Next*.
4. Next you will be asked to specify a directory in which to install Globus. If you followed the pre-requisite instructions, the path for `GLOBUS_LOCATION` will already be defined.
   a. In the **Directory** text field enter {***the path defined for your GLOBUS_LOCATION variable***} and click **Next**.

5. Next you will be asked to specify a directory in which to install caGrid. If you followed the pre-requisite instructions, the environment variable for `USER_HOME` will already be set.
   a. In the **Directory** text field enter `USER_HOME/.cagrid` and click **Next**.
6. You will then be asked if you would like to reconfigure caGrid for another target grid
   a. Leave the check box blank and press **Next**.
   b. Select the **NCICB Production Grid** from the **Target Grid** drop down and click **Next**.
   c. Click the **Start** button, the installer will begin to download and install Globus and caGrid

   Note: This step may take a while to download and extract all the files, and build caGrid

   d. Once the tasks have finished, click **Next**.
   e. In the **Hostname** text field, enter the name of the host that will run the service.

      Example: *cagrid-dorian.nci.nih.gov*

   f. Click **Next**.
   g. In the **Shutdown** port text field enter the port assignment number for the service being installed (Example: ***8005***). In the **HTTPS Port** text field enter the port assignment number for the service being installed (Example: ***8443)***.

   Note: Port Assignments will vary based on the service being installed and the planning done. Using the planning example, the following *shutdown* and *HTTPS* port assignments were made for Dorian:

   `Dorian:8005/8443`

7. Edit the service metadata for your deployment. Be sure to provide:
   — Appropriate *Research Center Metadata* (including the Address, etc)
   — Appropriate *Research Center Points of Contact* (including a point of contact for support questions)
   a. Click **Next**.
8. In the next screen you will configure Dorian Standard Properties.
   a. Verify the perform.index.service.registration is *true* and verify the `index.service.url` is set correctly based on the planning done in section

      Example: https://cagrid-index.nci.nih.gov:38443/wsrf/services/DefaultIndexServices

   b. Click **Next**
9. In the next screen, edit the database connection information specifying the **Database Hostname**, **Database Port**, **Database Name**, ***Database Username,***

and ***Database Password.*** This information can be obtained from the DBA (Database Administrator)

Example:

```
Hostname:  cbiodb580
Database Port:  3641
Database Name:  ccts_dorian
Database username:  cctsbunduser
Database Password:  ct5jzpk#5
```

a. Click **Next**.

Note: This validates the JDBC connection settings (as they are needed later). If you get an error, be sure MySQL is installed and running with a username and password matching your specifications.

10. The next screen allows for the configuration of the Dorian IdP. Complete the following steps:

    a. In the **IdP Name** field enter ***`Dorian`***.

    b. From the **Registration Policy** drop down, select **Automatic Registration**.

    c. Click **Next**.

11. The next screen allows for the configuration of the federation properties of Dorian. Complete the following steps:

    a. In the **Credential Lifetime Years** text field enter `5`.

    b. In the **GTS URL** text field enter the url/port assignment for the server to be configured for GTS. The remaining pathname - `/wsrf/services/cagrid/GTS` should be retained as is.

    Example: https://cagrid-gts.nci.nih.gov:28443/wsrf/services/cagrid/GTS.

    c. Click **Next**.

12. The next screen asks you which Dorian Certificate Authority type to use, from the **CA Type** drop down select **DBCA**.

    a. Click **Next**.

13. The next screen allows for the configuration of the Dorian Certificate Authority. To configure the Dorian CA, complete the following steps:

    a. In the ***Credential Auto-renew Years*** text field enter *5*.

    b. In the **Certificate Subject** text field enter an appropriate subject, for example:

    `O=caBIG,OU=caGrid,OU=PS LOA1,CN=caGrid PS LOA1 Certificate Authority`.

    c. In the **Lifetime Years** text field enter ***`25`***.

    d. Click **Next**.

14. The next screen facilitates the creation of host credentials for the Tomcat container running Dorian. To obtain the credentials complete the following steps:

a. In the **Hostname** text field enter the hostname of the host that will run Dorian

   Example: ***cagrid-dorian.nci.nih.gov***

b. In the **Directory** text field specify a location to write those credentials to

   Example: **USER_HOME/.globus/certificates)**

c. Click **Next**.

d. Click the **Start** button to install Dorian as configured.

e. Once Dorian has finished installing click **Next.**.

15. The installer will instruct you set the following environment variables:

    a. ***ANT_HOME***, ***GLOBUS_LOCATION***, and ***CATALINA_HOME***.

    b. These should already be set based on performing all the pre-requisite procedures; however, if they are not, set them now.

    c. Click the **Finish** button and the click the **Close** button to close the installer.

During installation a copy of the Dorian CA certificate was place in the directory, `USER_HOME/.globus/certificates` (based on the path you defined during the install process).

The CA certificate has a file prefix containing a hash code of the CA and a extension of *.0*, for example `68907d53.0`.

The CA Signing policy has a file prefix containing a hash code of the CA and a extension of `.signing_policy`, for example `68907d53.signing_policy`.

Important: You should make a copy of the Dorian CA certificate, the file with the *.0* extension and make a note of the location where you copied it to.

4.2.3 Start Dorian

To complete the setup procedures for Dorian, follow these steps:

1. Start the service on the Dorian Server.
2. Login to the Dorian server using the user defined for Dorian (Example `tomcat-dor`)

   ```
   %> cd $CATALINA_HOME/bin
   %> ./startup.sh
   ```

3. Once it has started up, be sure to look in `$CATALINA_HOME/logs/catalina.out` for any errors. From the folder where the service was started from, perform the following:

   ```
   %> cd ../logs
   %> more catalina.out
   Administer Dorian
   ```

4.2.3.1 Run GAARDS UI

When first run Dorian comes configured with a default user account, the *dorian* user. For security reasons the first thing you want to do is register a second user. The second

user should be considered a "real user" or bound to a "real person". Once you have created this user you will assign the user administrative rights and then remove the *dorian* or default user. The GAARDS Admin UI (distributed with caGrid) provides a mechanism for administrating Dorian.

To launch the GAARDS UI, complete the following steps:

1. Login to the Dorian server with the Dorian userid (Example `tomcat-dor`). Ensure the GUI emulator has been set, if necessary (see *Section 4.1 GUI Emulator for Linux Servers*).
2. From the directory where caGrid was installed on Dorian. (If you are following the deployment plan and sample folder structure, this is probably `$HOME/.cagrid/caGrid`.) See *Figure 7*.

```
%> cd USER_HOME/.cagrid/caGrid
%> ant security
```

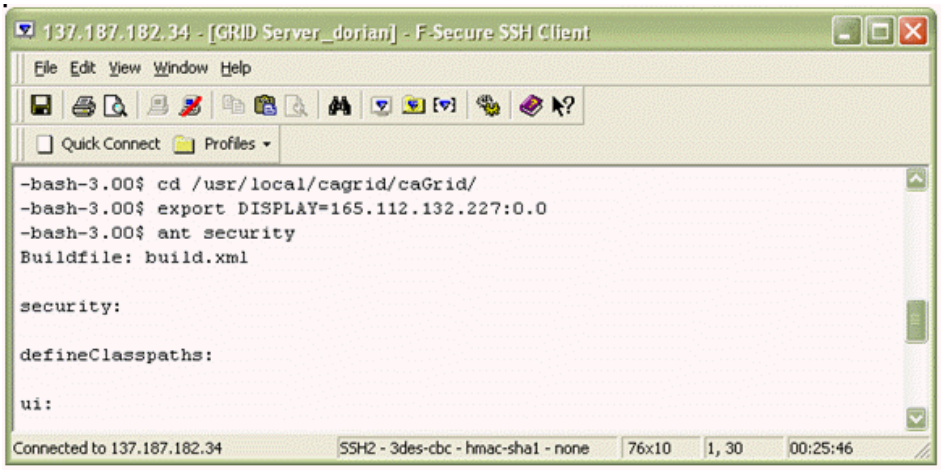

Figure 7 Details displaying on Linux; brings up GUI which allows for the definition of a new user

4.2.3.2 Setup GAARDS Preferences

Since this is the first instance of using the GAARDS UI since Dorian has been installed, follow these steps to define the server/port details for your site using the preferences:

1. Select **Window > Preferences**.
2. In the Preferences window that opens, under **Grid User Management**, select **Dorian Service(s)** (*Figure 8*).
3. In the **Add/Remove Value(s)** text box, enter the server/port assignment for Dorian. The remainder of the path /wsrf/services/cagrid/Dorian should be included in the new server/port you define (Example https://cagrid-dorian.nci.nih.gov:8443/wsrf/services/cagrid/Dorian) .
4. Click the **+ Add** button.

Note: Other default values can be removed if the server/port does not apply to your configuration by highlighting that server/port line and then clicking the **- Remove** button.

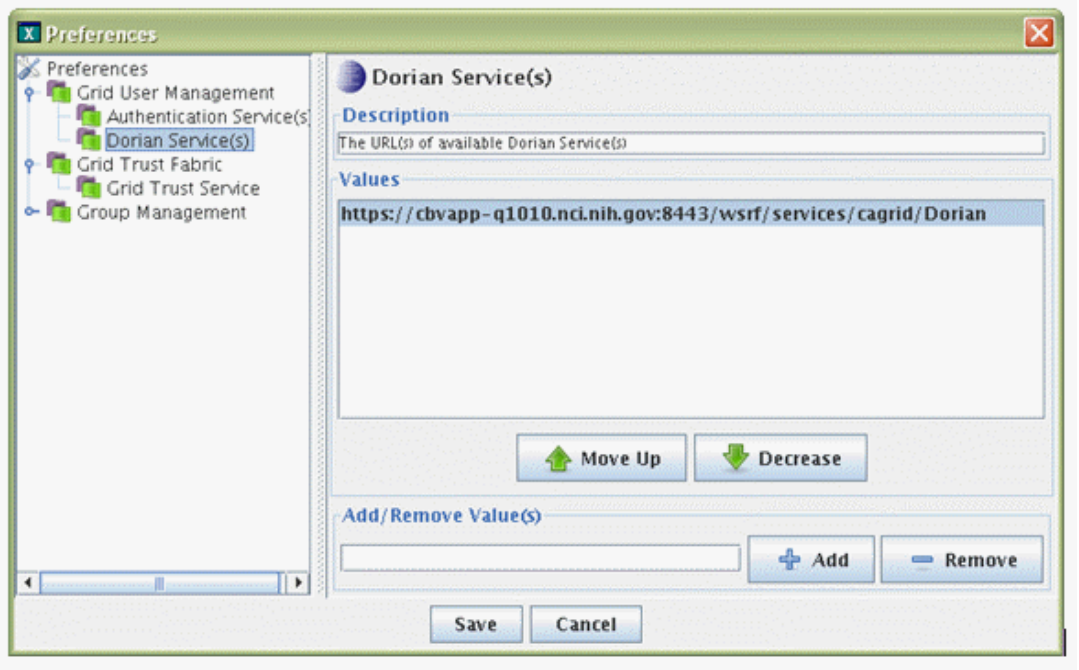

Figure 8 Setting Dorian Services preferences

Since the Authentication Service is not being used and Dorian will always need to be defined, you can perform the same changes under the Authentication Service(s) option to replace this with your local server/port and the Dorian service (*Figure 9*).

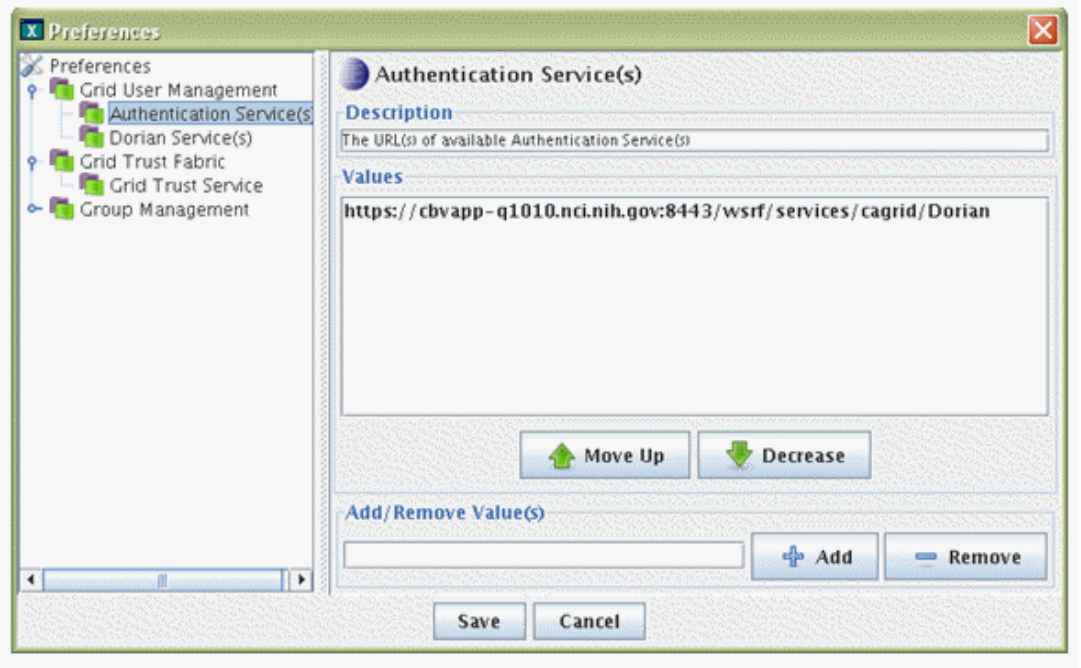

Figure 9 Authentication Service preferences

Since the Authentication Service is not being used and Dorian will always need to be defined, you can perform the same changes under the Authentication Service(s) option to replace this with your local server/port and the Dorian service:

The same update can be made to define the preferences for GTS (*Figure 10*). The server/port assignments will change based on where you are configuring the GTS service to run. Since it will be in its own Tomcat container, the port assignment should be different from Dorian if they are running on the same physical server in different Tomcat containers.

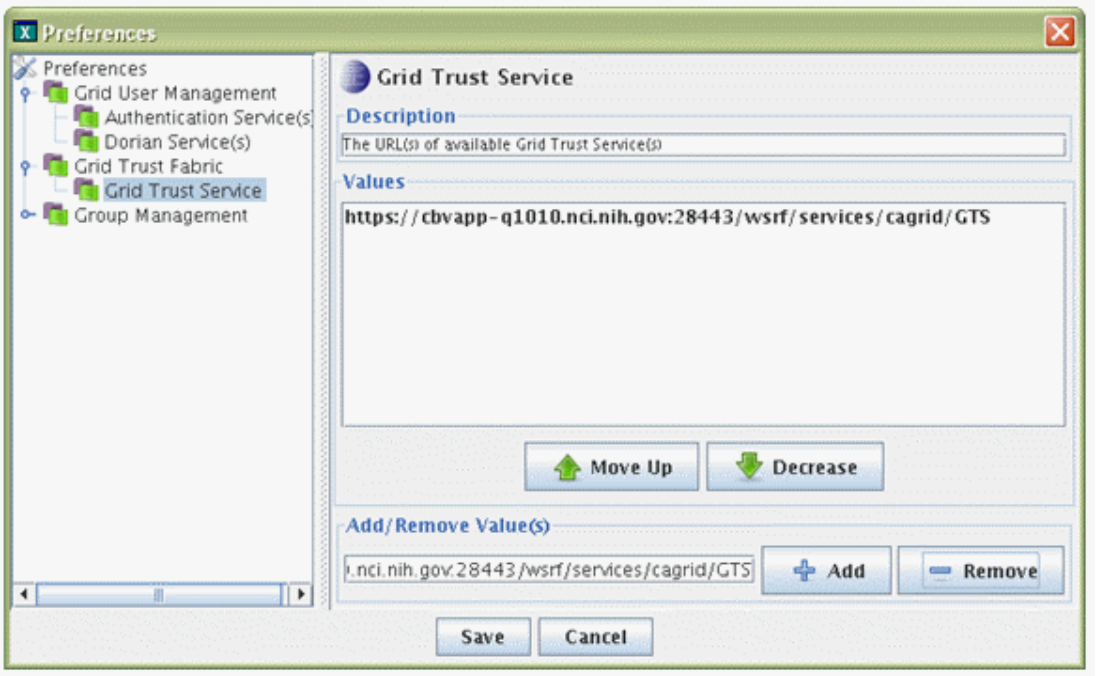

Figure 10 Grid Trust Service preferences

Now that the administrative setup to define your specific port/server assignments for Dorian and GTS has been completed, you are ready to create a new user so that you can remove the default user provided with the Dorian install.

4.2.3.3 Create a New User

To create a new user, follow these steps:

1. Login using the default administrator account that was created during the Dorian Installation process (userid: ***dorian***; password: ***DorianAdmin$1***).

2. Be sure the correct server is defined in the Dorian Service & Authentication Service pull-down options (*Figure 11*). (This should be the only server in the list if the previous steps were performed):

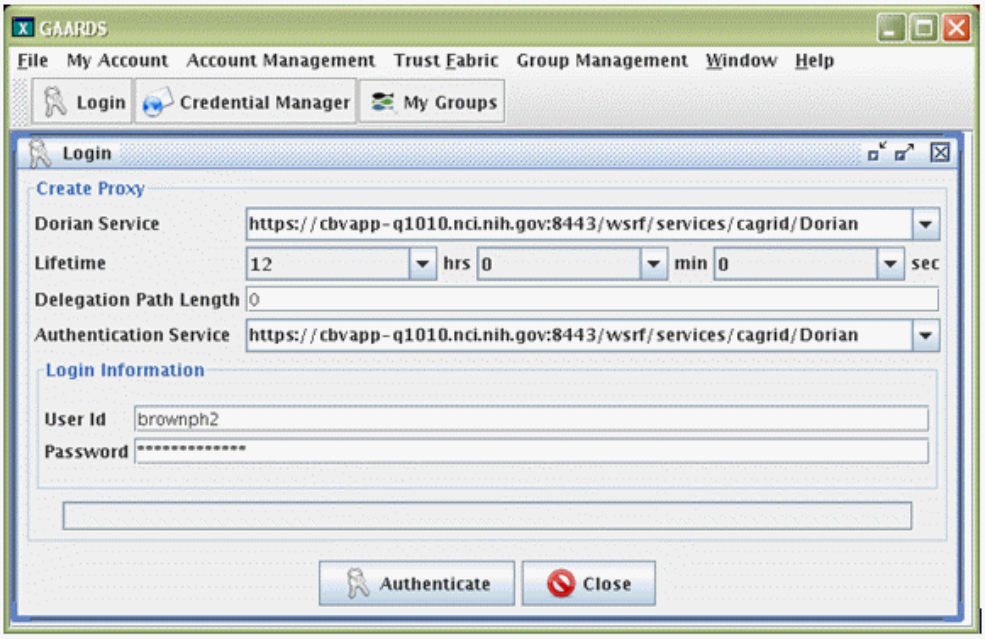

Figure 11 Dorian Service & Authentication Service

3. Select **Authenticate**.
4. On the Main window that opens, select **Account Management > Local Accounts > Registration**.
5. In the form that opens, enter the appropriate information to complete the user setup (*Figure 12*). Ensure the correct Server & Service is selected in the

Service pull-down. This should be the “secure” server that Dorian is configured (installed) on.

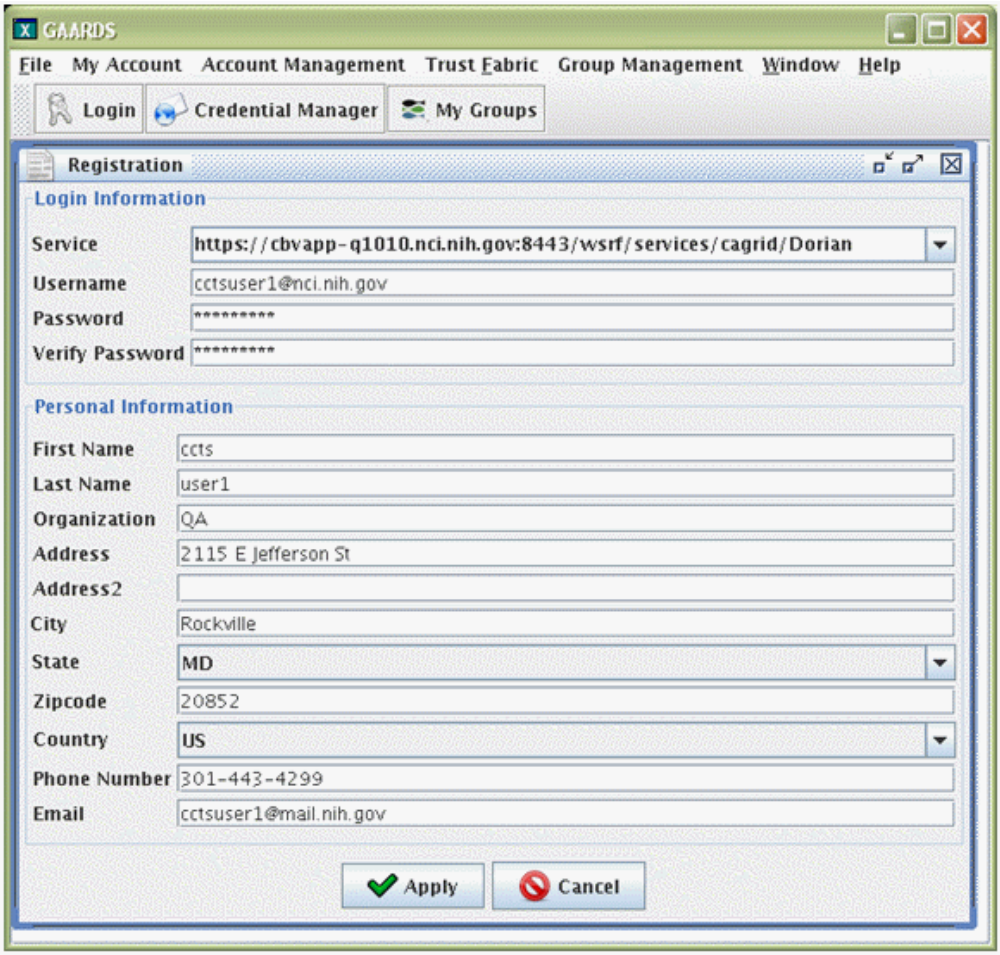

Figure 12 Registration form

6. Hit the **Apply** button and if all information is correct, a message opens confirming that the account is approved and the account status is Active.
7. To verify that the user has been added select **Account Management > Local Accounts > Local Account Management**.

8. Select the local Dorian server in the **Service URL** and the credentials for the administrator ID currently logged in as (dorian), and then select **Find Users** (*Figure 13*).

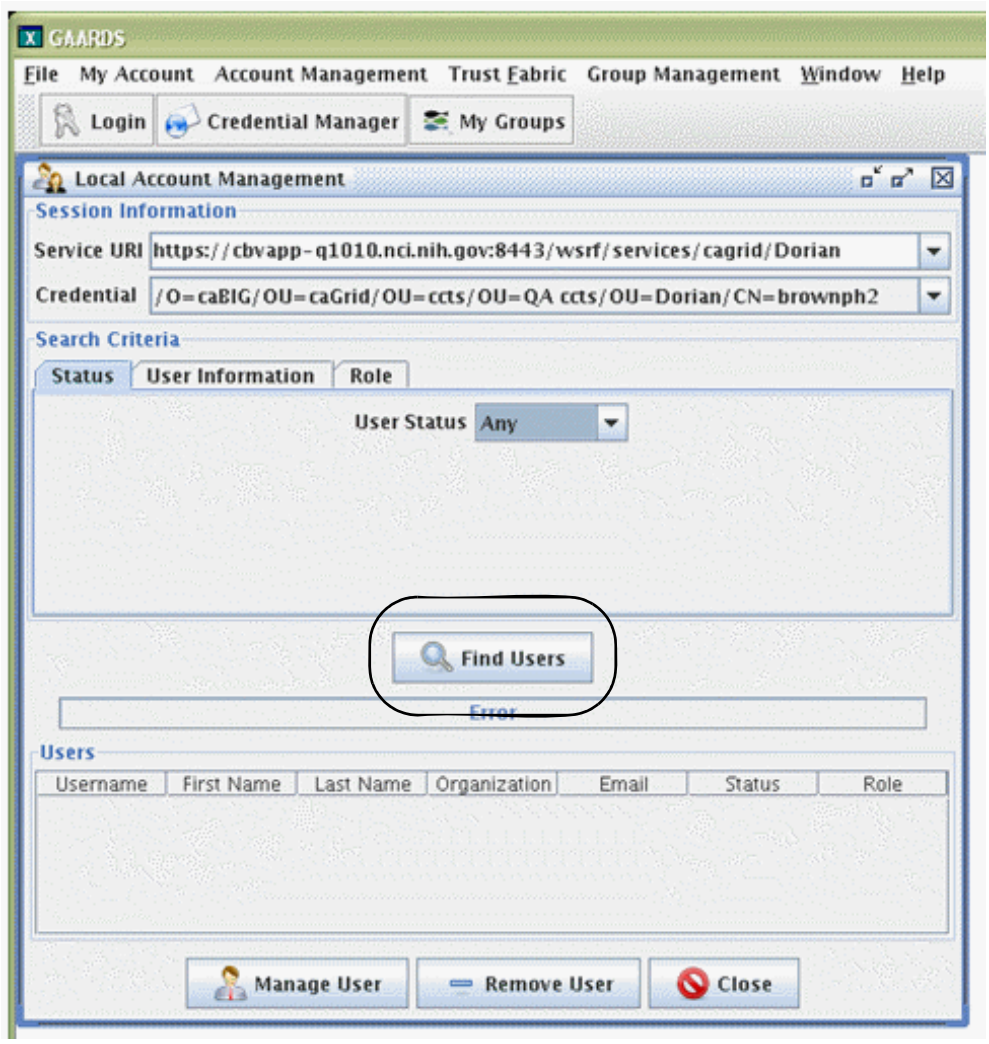

Figure 13 Verifying user information

The user id just added should be displayed in the list as an **Active Non Administrator:Supplementary Resources** (*Figure 14*). **(**

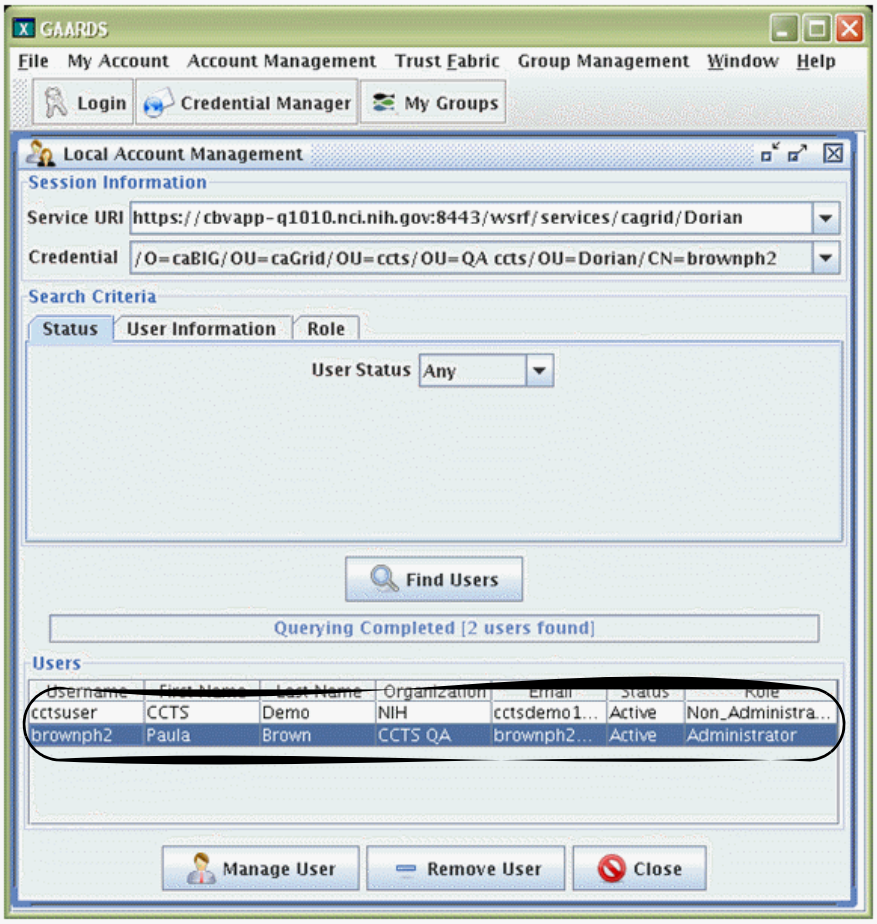

Figure 14 User ID displays at the bottom of the window

9. Give the new user administrator privileges by following these steps:
   a. Highlight the added user and select the **Manage User** button.
   b. Click the **Account Information** tab. If not already selected, from the **User Status** drop down select *Active*.
   c. From the **User Role** drop down select **Administrator**.
   d. Click the ***Update User*** button. This updates the User just added to an Administrator.
   e. Close the window.
10. Verify the new user is able to successfully perform a login.
   a. Click the **Login** button.
   b. Select the local server where Dorian is installed for the Dorian and Authentication Service.

c. Enter the userid and password for the user just created:

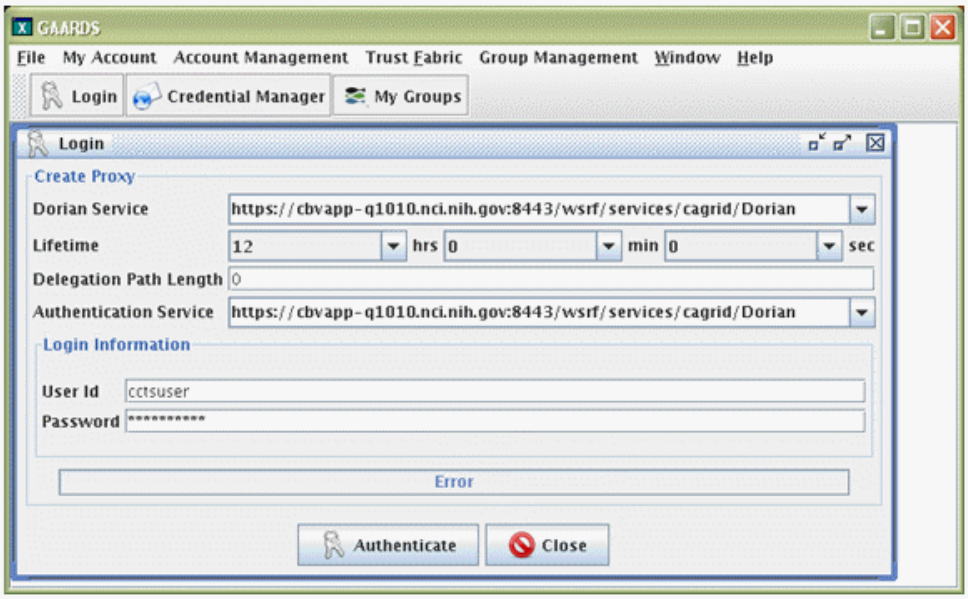

Figure 15 Verifying that a new user can perform a login.

d. Press **Authenticate**.

If the login is successful, the screen shown in *Figure 16* displays:

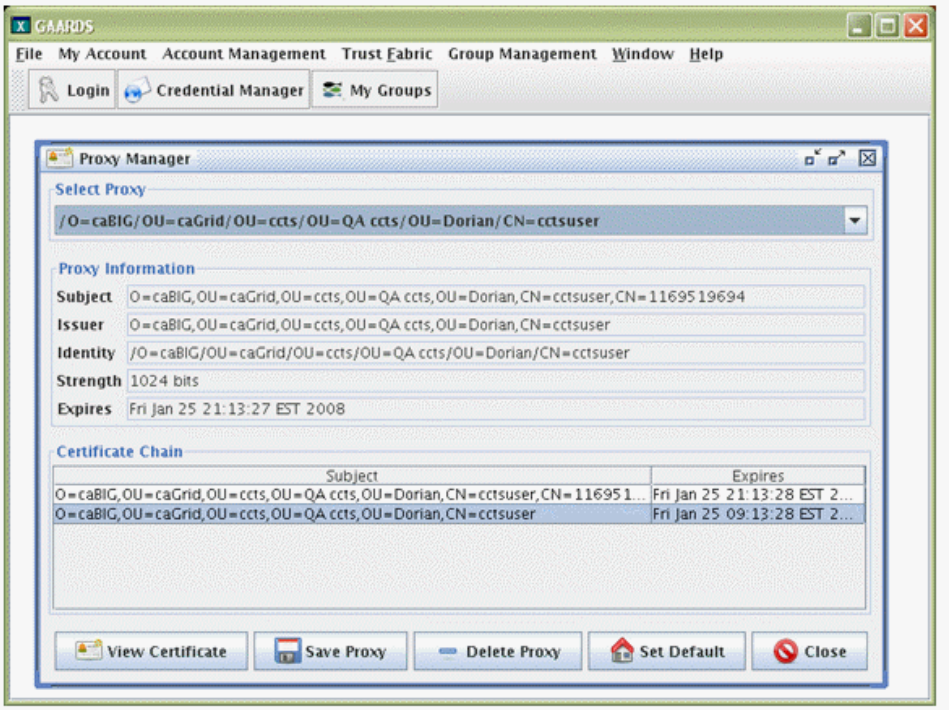

Figure 16 A screen showing a successful login

11. Close all open windows, but <u>do not exit</u> the GAARDS UI tool.

4.2.4 Bind Host Credentials to the New User

You should already be logged into the GAARDS UI tool with the user created in *Section 4.2.3.3 Create a New User*. You should have verified the ability to login to this new account. (If not, login with this user/password.

To bind host credentials to the new user, perform these steps:

1. Click the **Authenticate** button. This authenticates you to Dorian using the account just created and launches the Proxy Manager window.
2. Click the **Set Default** button.

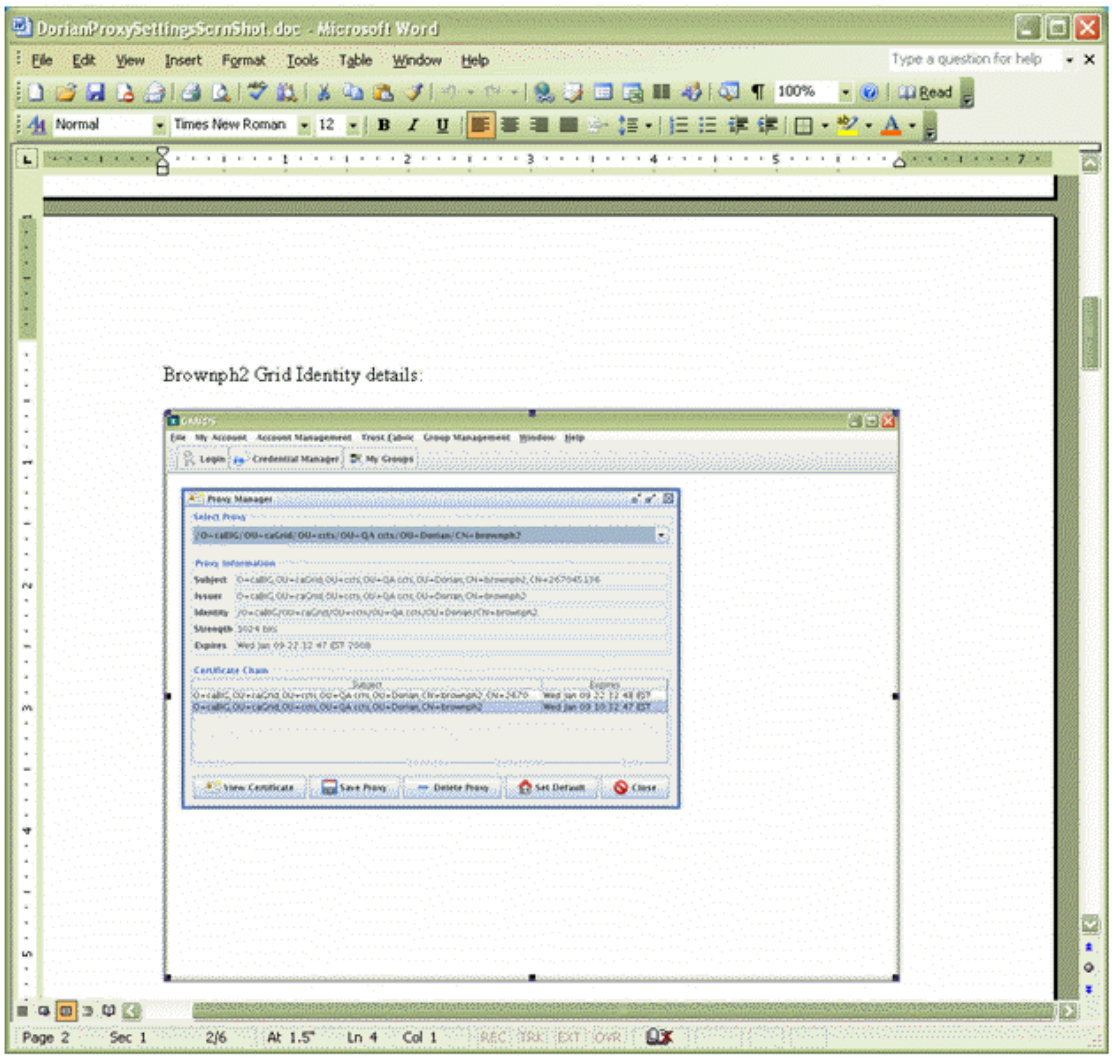

Figure 17 Proxy Manager

3. If you have not yet done so, in the Proxy Manager, take note of your **Grid Identity.** This is the grid wide unique identifier for this user, which authorization policies can be set against.
4. Close the window.
5. From the **Account Management** menu, select **Grid Account Management > Host Certificate Management**.

This opens the Host Certificate Management window.

6. Click the **Find Host Certificates** button.

   This lists all the host certificates issued by this Dorian service. At this point there should most likely be one (the one for Dorian itself).

7. Select the host certificate for the host running Dorian and click the **View/Update Host Certificate** button. This launches the Host Certificate window for the selected host certificate.

8. **Record the Host Grid Identity for future use**. This is the Identity of the host credential the Dorian service is running with.

9. Click the **Find** button next to the **Owner** text field. This launches the Find Users window.

10. Click the **Find Users** button. This lists all the users with Grid Accounts on Dorian.

11. Select the user you created in *Section 4.2.3.3 Create a New User* and click the **Select Users** button.

    This returns you to the Host Certificate window populating the **Owner** text field with the grid identity of the user just selected (*Figure 18*).

12. Click the **Update Certificate** button.

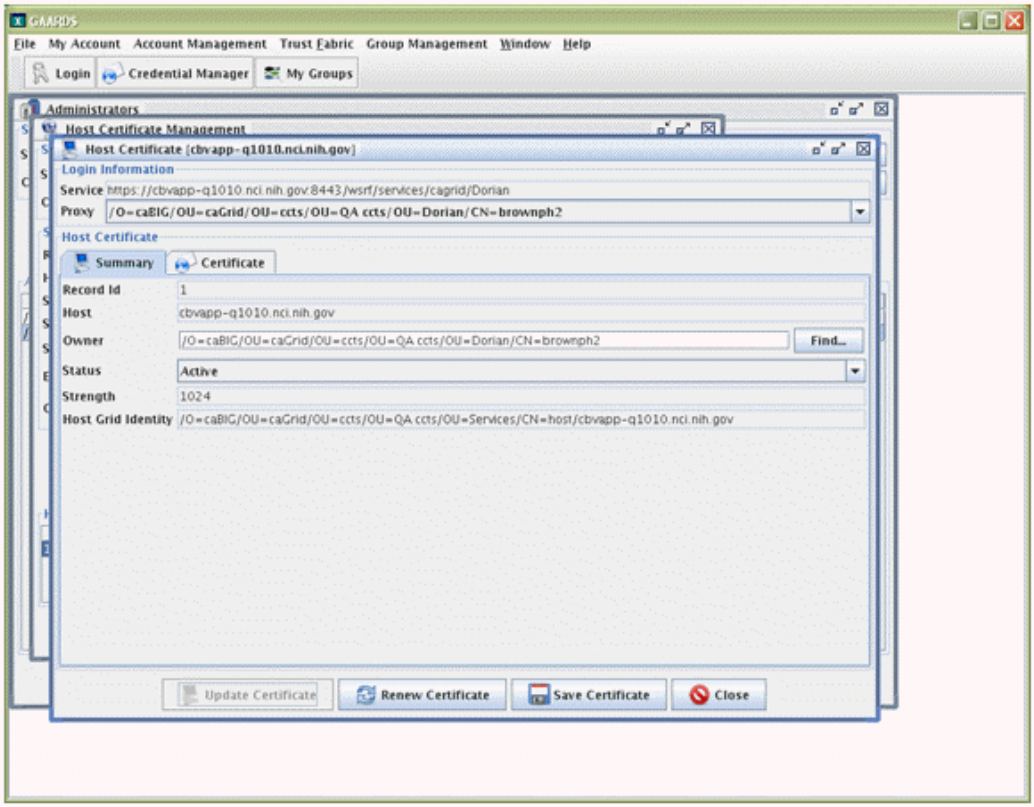

Figure 18 Host Certificate window

13. To verify that this change was successfully made, from the Host Certificate Management window, click the **Find Host Certificates** button.

    This lists all the host certificates issued by this Dorian, at this point there should most likely be one (the one for Dorian itself).

14. Select the host certificate for the host running Dorian and click the **View/Update Host Certificate** button.
15. This launches the Host Certificate window for the selected host certificate. Double check that the **Owner** text field contains the grid identity of the user created above.
16. Close all windows with exception the security UI itself (the main application).

    At this point you are ready to remove the default Dorian account. Before doing so make sure that you have completed the above steps, specifically, that you have 1) registered a new user; 2) approved a new user and made him or her an administrator of the Dorian IdP; 3) added the new user as a grid account administrator; and 4) bound the Dorian host credentials to the new user. Once you are confident that you have successfully completed these steps, remove the default user account aby following these steps:

    a. From the **Account Management** menu, select **Local Accounts > Local Account Management**. This opens a Local Account Management window.
    b. Click the **Find Users** button.
    c. Select the **dorian** user and click the **Remove User** button.

Note: After this is done, you will no longer be able to administer Dorian using the default user. You must be sure you remember the username and password for the new administrative account

DO NOT EXIT the GAARDS UI utility as it will be used in the next section:

4.3 Create Additional Host Certificates

Additional certificates will be needed as the installation of other components for the Suite are installed. To simplify this process, all certificates will be created on the Dorian server using the GAARDS UI utility. These certificates will be saved in a common folder with a standard naming convention to allow for uploading them to their respective servers as each install is completed.

Note: Every secure Tomcat instance needs to run using a host certificate. This section describes typical host certificates needed; however, this may change depending on your environment.

To create addition host certificates, follow these steps:

1. In the GAARDS UI, select **My Accounts > Request Host Certificate**. This opens the Request Host Certificate window (*Figure 19*)

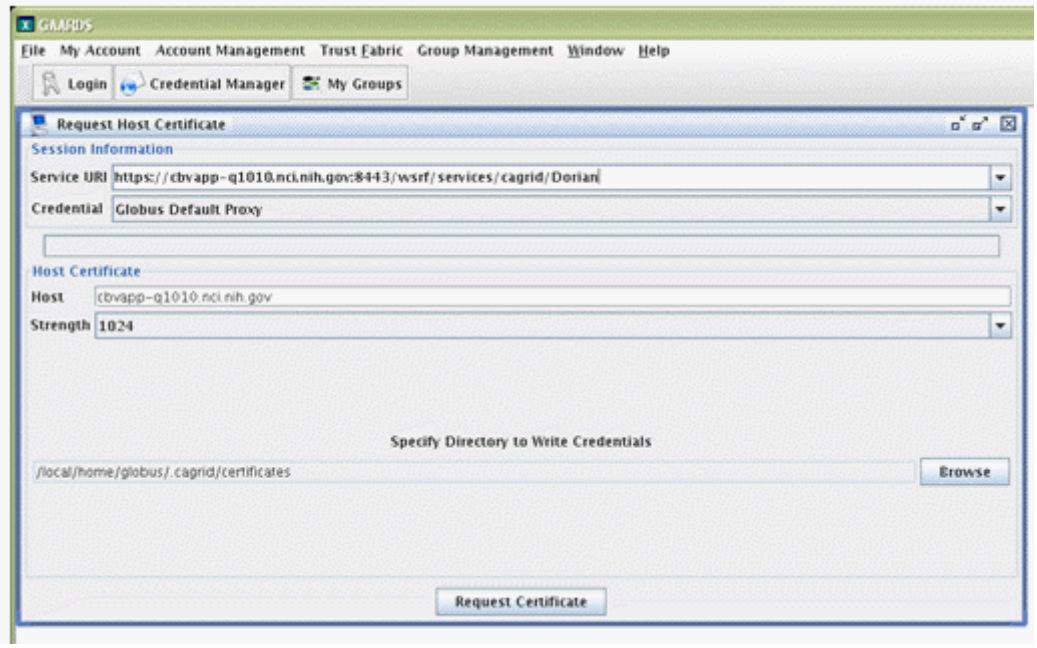

Figure 19 Request Host Certificate window

2. Ensure that the **Service URL** is the correct server/port for where the Dorian Service is installed.
3. Define the **Host** in the **Host Certificate** section, using a standard naming convention for the certificates to be created:
   - cds-cert (to be used during the CDS install process)
   - gts-cert (to be used during the GTS install process)
   - c3d-cert (to be used during the C3D install process)
   - sso-cert (to be used during the WebSSO install process)
   - app-cert (to be used during the application install process)
   - cax-cert (to be used during the caXchange install process)
4. Specify the **Directory** to write the credential. (Use the same common directory for all host certificates created so they are all located in one standard location. This makes it easy to remember where to retrieve them when they are needed later).

   Example: `/local/home/tomcat-dor/HostCertificates`

Once the certificate has been created, the screen shown in *Figure 20* is displayed

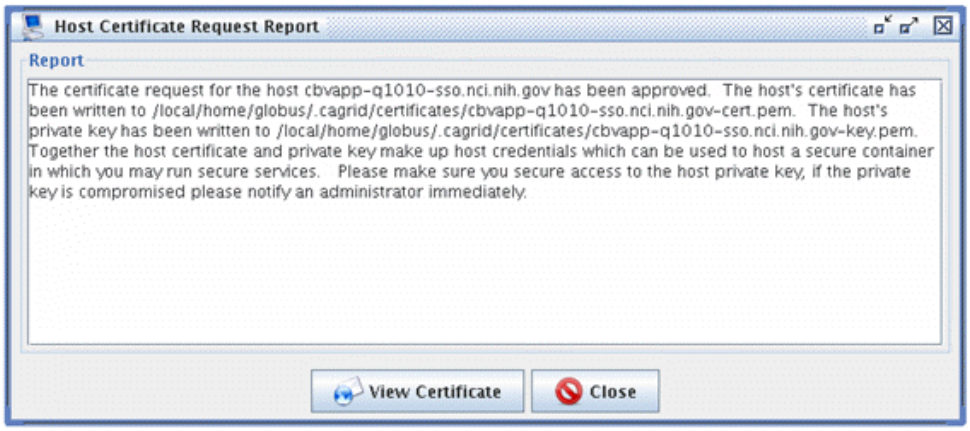

Figure 20 Host Certificate Request Report

5. Select the **Close** button and repeat the process to create host certificates for all those listed in the naming convention example above.

This process creates two files for each certificate requested: a `cert.pem` and a `key.pem`. Using the standard naming convention recommended above, the files would have a naming standard such as `gts-cert.pem` and `gts-key.pem`, making it easy to determine which certificate files belong to which service. It also makes it easy to distinguish which file is the certification file and which file is the key file for each service.

4.4 GTS

GTS maintains trust across the grid service in the Suite. The following sections describe how to install GTS and begin the process of syncing the trust fabric.

4.4.1 Copy GTS Certificates

Prior to running the grid installer, you must download the `gts-cert.pem` and `gts-key.pem` files from the Dorian server to a local folder on the GTS server (if these services are running on two physically separate servers).

1. On the GTS server at the `$HOME directory:`

   ```
   %> mkdir certificates
   ```

2. Upload the gts cert and key files from the Dorian server to this local folder on GTS.

These files will be used by the installer.

4.4.2 Launch the caGrid 1.1 Installer

1. To download the installer, login to the server to be installed.
2. At the $HOME directory (`%> cd $HOME`):

   ```
   %> mkdir installer
   %> wget http://gforge.nci.nih.gov/frs/download.php/2371/caGrid-1.1-installer-rc11-ncicb.zip
   ```

```
%> cd installer
%> unzip caGrid-1.1-rc1-installer.zip
```

3. From this same directory, launch the installer:

```
%> java -jar caGrid-1.1-installer.jar
```

This brings up a GUI installer.

4.4.3 Install GTS

The following steps walk you through the procedure for installing the GTS service into a Tomcat container provided by the installer.

1. Select installation components.
   a. Accept the caGrid license and click **Next**.
   b. Select the **Install caGrid** checkbox. Select the **/nstall caGrid Services** checkbox. De-select all other check boxes.
   c. click **Next**.
   d. Select the **GTS** check box.
   e. If SyncGTS is to be installed in the same tomcat container as GTS, select the **syncGTS** check box, and click **Next**.
   f. From the **Container Type** drop down, select **Tomcat** and click **Next**.
2. You are asked to specify a directory in which to install Ant.
   a. In the **Directory** text field, enter ***{the path defined for your ANT_HOME variable}*** and click **Next**.

      If you followed the instructions and installed all the pre-requisite software prior to beginning the 1.1 Grid install process, ANT will already be installed and the `ANT_HOME` environment variable already set. If this is the case, leave the check box blank and press **Next**
3. Next you are asked to specify a directory in which to install Tomcat.
   a. In the **Directory** text field enter ***{the path defined for your CATALINA_HOME variable}*** and click **Next**.

      If you followed the instructions and installed all the pre-requisite software prior to beginning the 1.1 Grid install process, TOMCAT will already be installed and the `CATALINA_HOME` environment variable already set. If this is the case, leave the check box blank and press **Next**.
4. Next you are asked to specify a directory in which to install Globus. If you followed the pre-requisite instructions, the path for `GLOBUS_LOCATION` is already defined.
   a. In the **Directory** text field enter ***{the path defined for your GLOBUS_LOCATION variable}*** and click **Next**.
5. Next you are asked to specify a directory in which to install caGrid. If you followed the pre-requisite instructions, the environment variable for `USER_HOME` is already set.
   a. In the **Directory** text field enter ***USER_HOME/.cagrid*** and click **Next**.

6. You are then asked if you would like to reconfigure caGrid for another target grid.
   a. Leave the check box blank and press **Next**.
   b. Select the **NCICB Production Grid** from the Target Grid drop down and click **Next**.
   c. Click the **Start** button. The installer begins to download and install Globus and caGrid

   Note: This step may take a while to download and extract all the files, and build caGrid

   d. Once the tasks have finished, click **Next**.
   e. In the **Hostname** text field, enter the name of the host (***`cagrid-gts.nci.nih.gov`***) that will run the service and click **Next**.
   f. In the **Shutdown** port text field, enter the port assignment number for the service being installed (Example: ***`8005`***). In the ***HTTPS Port*** text field, enter the port assignment number for the service being installed (Example ***`8443`***).

   Note: Port assignments vary based on the service being installed and the planning done. Using the planning example, the following *shutdown* and *HTTPS* port assignments were made for GTS:

   `GTS:28005/28443`

7. Next the installer asks if server credentials are present. Select the **Yes** check box and click **Next**.
   a. In the **Certificate Path** text field, browse to the location of the local certificate for GTS that was downloaded from Dorian to GTS prior to beginning the grid install process. If the example was used, this would be `$HOME/certificates/gts-cert.pem`.

   Note: If GTS is being installed on a physically different server than the Dorian server, the certificates for GTS need to be downloaded to a local folder on the GTS server.

   b. In the **Certificate Key** text field, browse to the location of the private key for GTS that was downloaded from Dorian to GTS prior to beginning the grid install process. If the example was used, this would be `$HOME/certificates/gts-key.pem`.
   c. Click **Next**.
8. Edit the service metadata for your deployment. Be sure to provide:
   — Appropriate **Research Center Metadata** (including the Address, etc)
   — Appropriate **Research Center Points of Contact** (including a point of contact for support questions)
   a. Click **Next**.
9. In the next screen, GTS Standard Properties, you DO NOT need to edit anything; just click **Next**.

10. Verify that `perform.index.service.registration` is true and `index.service.url` is set appropriately based on where the index service has been configured in your planned deployment. Click **Next**.
11. In the next screen, edit the database connection information specifying the **Database Hostname**, **Database Port**, **Database Name**, ***Database Username,*** and **Database Password**. Click **Next**.

    Note: This validates the JDBC connection settings (as they are needed later). If you get an error, be sure MySQL is installed and running with a username and password matching your specifications.
12. If a database with the same name exists in the database service, the next panel will indicate that the existing database will be destroyed. press **Next**.
13. The next screen prompts you for an initial administrator for the GTS. In the **Identity** text field, enter the Grid Identity for the GTS user.

    Example: ***`O=caBig/OU=caGrid/OU=ccts/OU=PD ccts/CN=caGrid GTS CA`***

    a. Click **Next**.

    b. Click the **Start** button to install the GTS as configured.

    c. Once the GTS has finished installing click **Next**.

    The installer instructs you to set the following environment variables: `ANT_HOME`, `GLOBUS_LOCATION`, and `CATALINA_HOME`. If these were not set during the pre-requisite steps, **set these environment variables now.**

    d. Click the **Finish** button, and the click the **Close** button to close the installer.

4.4.4 Copy the GTS Root CA File

To copy the GTS root CA file, follow these steps:

1. On the GTS server, if the directory does not already exist, create the following:

```
%> mkdir $HOME/.globus/certificates
```

2. Copy the **GTS** certificate file from the `$HOME/certificates` directory (Example: `gts-cert.pem`) to the `$HOME/.globus/certificates` folder.

   Note: The filename must end with a digit extension when copied to this directory (Example: `gtsca.0`):

```
%> cd $HOME/.globus.certificates
```

   If the example directories from this install process have been used, the following command would apply:

```
%> cp $HOME/certificates/gts-cert.pem  gtsca.0
```

4.4.5 Copy the Dorian CA File

The Dorian certificate created when the Dorian server was installed also needs to be copied to the GTS server at this same location. This file will also be located under the `$HOME/.globus/certificates` folder on the DORIAN server and must be copied to the GTS server under the same folder structure. If Dorian and GTS are on separate physical servers, this file must be uploaded from Dorian in order to copy it to the correct folder on GTS.

If Dorian and GTS are on the same physical servers, follow these steps:

1. On the Dorian `$HOME/.globus/certificates` there should be a file with a .0 extension that was created during the dorian grid install. Locate this file.
2. On the GTS server, perform the following:

   ```
   %> cd $HOME/.globus/certificates
   %> cp /local/home/tomcat-dor/.globus/certificates/*.0 .
   ```

   Note: The path defined here is the location of the certificate on the Dorian server. The . copies the file over to the GTS `$HOME/.globus/certificates` folder, retaining the filename as it exists on the Dorian server.

If Dorian and GTS are on physically separate servers, follow these steps:

1. Upload the .0 file from the Dorian server and then download it to the GTS server in the `$HOME/.globus/certificates folder`.

   Once this is complete, on the GTS server, there should be 2 .0 files in the `$HOME/.globus/certificates` folder – one that belongs to GTS and one that belongs to Dorian.

4.4.6 Start GTS/SyncGTS

To complete the setup procedures for GTS and SyncGTS, start the service on the GTS server. To start the service, follow these steps:

1. Login to the GTS server using the userid defined for GTS (Example: `tomcat-gts`)

   ```
   %> cd $CATALINA_HOME/bin
   %> ./startup.sh
   ```

2. Once it has started up, be sure to look in `$CATALINA_HOME/logs/catalina.out` for any errors. From the folder from which the service was started, perform the following:

   ```
   %> cd ../logs
   %> more catalina.out
   ```

This completes the installation of the GTS server. You will return to GTS to create Trust Levels and Trust Fabrics later in the install process.

4.5 Complete the Trust Level Setup

4.5.1 Create Trust Levels

A level of assurance or trust level specifies the level of confidence with which a given certificate authority is trusted in the Grid. The GAARDS Admin UI provides a mechanism for administrating the GTS, and this includes creating trust levels.

To launch the GAARDS UI complete the following:

1. Change to the directory where caGrid was installed on the GTS server and run ant security to bring up the GAARDS UI tool.

   If the examples of folders and environment variables provided in this install plan have been followed it should be

```
%> cd $HOME/.cagrid/caGrid
%> ant security
```

To add a trust level to the GTS using the GAARDS UI complete the following steps:

1. Click the **Login** button. This launches the Login window.
2. From the **Dorian Service** drop down, select the server where dorian was installed.

   https://cagrid-dorian-qa.nci.nih.gov:8443/wsrf/services/cagrid/Dorian.

3. From the **Authentication Service** drop down select the server where dorian was installed.

   https://cagrid-dorian-qa.nci.nih.gov:8443/wsrf/services/cagrid/Dorian.

4. In the **User Id** text field enter the username for the administrator account previously created.
5. In the **Password** text field enter the password for the administrator account previuosly created.
   a. Click the **Authenticate** button. This authenticates you to Dorian using the administrator account previously created and launch the **Proxy Manager** window
   b. Click the **Set Default** button.

   Note: If you have not yet done so, take note of your Grid Identity; this is the grid wide unique identifier for this user, which authorization policies can be set against.

   c. Close the window.
6. From the **Trust Fabric** menu, select **Levels of Assurance**, to launch the Levels of Assurance window.
   a. Click the **Add Trust Level** button, to launch the Add Trust Level window.
7. In the **Service** drop down, select the server where GTS is installed. If the server is not available in the dropdown, replace the `server:port` assignment in the path with the server name/port for GTS (retaining the `/wsrf/services/cagrid/GTS` in the pathname)

   https://cagrid-gts.nci.nih.gov:28443/wsrf/services/cagrid/GTS.

8. In the **Name** text box enter ***LOA1***.
9. In the **Description** text box enter ***This trust level maintains a grouping of caGrid LOA1 Authorities.***
   a. Click the **Add Trust Level** button. This adds the trust level to the GTS.
10. To verify that the trust level was successfully added, complete the following steps:
    a. From the **Trust Fabric** menu, select **Levels of Assurance**. This launches the Levels of Assurance window.

b. In the **Service** drop down select the server where GTS is installed or replace the `server:port` in the path displayed (retaining the `/wsrf/services/cagrid/GTS`)

https://cagrid-gts.nci.nih.gov:28443/wsrf/services/cagrid/GTS.

c. Click the **List Trust Levels** button. This should list all the trust levels for the selected GTS in the table.

If the `LOA1` trust level appears in the table, then it was successfully added.

4.5.2 Add Dorian to the Trust Fabric

You are now ready to add Dorian to the Trust Fabric. Continuing to work in the GAARDS UI on the GTS server, perform the following steps to add the Dorian CA to GTS.

1. From the **Trust Fabric** menu select **Certificate Authorities** to launch the Certificate Authorities window.

   a. Click the **Add Trusted Authority** button to launch the Add Certificate Authority window.

2. In the **Service** drop down, select the server where GTS is installed. If the server is not available in the dropdown, replace the `server:port` assignment in the path with the server name/port for GTS (retaining the `/wsrf/services/cagrid/GTS` in the pathname)

   Example: https://cagrid-gts.nci.nih.gov:28443/wsrf/services/cagrid/GTS.

   a. Click the **Import Certificate** button. This brings up a file browser. Browse to the file containing the Dorian CA Certificate and click the **Open** button. This imports the certificate into the UI.

   Note: This is the Dorian CA .0 file previously copied from Dorian to GTS, which was stored in the `$HOME/.globus/certificates` folder.

   b. Click on the **Trust Levels** tab.

   c. Select the check box for the **LOA1** trust level.

   d. Click the **Add Trusted Authority** button. This adds the Dorian CA to the GTS.

3. To verify that the Dorian Certificate Authority was successfully added, complete the following steps:

   a. From the **Trust Fabric** menu select **Certificate Authorities**. This launches the Certificate Authorities window.

   b. In the **Service** drop down, select the server where GTS is installed. If the server is not available in the dropdown, replace the `server:port` assignment in the path with the server name/port for GTS (retaining the `/wsrf/services/cagrid/GTS` in the pathname)

   https://cagrid-gts.nci.nih.gov:28443/wsrf/services/cagrid/GTS.

   c. Click the **Find Trusted Authorities** button. This should list all the certificate authorities registered with GTS in the table.

If the Dorian CA is listed in the table then it was successfully added.

4.5.3 Grant Dorian CRL Publication Rights

You are now ready to grant Dorian rights to publish its CRL. Continuing to work in the GAARDS UI on the GTS server, perform the following steps to grant Dorian rights to Publish its CRL

1. From the **Trust Fabric** menu select **Permissions** to launch the Permission window.
   a. Click the **Add Permission** button to launch the Add Permission Window.
2. From the **Service** drop down select the server where GTS is installed. If the server is not available in the dropdown, replace the `server:port` assignment in the path with the server name/port for GTS (retaining the `/wsrf/services/cagrid/GTS` in the pathname)

   https://cagrid-gts.nci.nih.gov:28443/wsrf/services/cagrid/GTS.
3. In the **Grid Identity** text box enter:

   /O=caBIG/OU=caGrid/OU=QA LOA1/OU=Services/CN=host/cagrid-dorian-qa.nci.nih.gov
4. From the **Trusted Authority** drop down select: **O=caBIG,OU=caGrid,OU=QA LOA1,CN=caGrid QA LOA1 Certificate Authority**
5. From the **Role** drop down select: **TrustAuthorityManager**.
   a. Click the **Add Permission** button. This grants Dorian the ability to publish its CRL to the GTS.
6. To verify that Dorian was successfully granted rights to publish its CRL complete the following steps:
   a. From the **Trust Fabric** menu select **Permissions** to launch the Permission window.
   b. From the **Service** drop down, select the server where GTS is installed. If the server is not available in the dropdown, replace the `server:port` assignment in the path with the server name/port for GTS (retaining the `/wsrf/services/cagrid/GTS in the pathname`)

      https://cagrid-gts.nci.nih.gov:28443/wsrf/services/cagrid/GTS.
   c. Click the **List Permissions** button.

   This lists all the rights granted on the GTS in the table. If Dorian was successfully granted rights to publish its CRL, you will see a listing as follows:

   Grid Identity - *`/O=caBIG/OU=caGrid/OU=QA LOA1/OU=Services/CN=host/cagrid-dorian-qa.nci.nih.gov`*

   Trusted Authority - *`O=caBIG,OU=caGrid,OU=QA LOA1,CN=caGrid QA LOA1 Certificate Authority`*

   Role - *`TrustAuthorityManager`*
7. If this is correct, exit the GAARDS UI.

4.6 CDS

CDS also requires a secure tomcat container; however it is not a Grid service that can be installed using the caGrid 1.1 installer. CDS also has its own software package that

must be installed in addition to building out the secure Tomcat. This section describes the installation process.

4.6.1 Download CDS

Login to the CDS server and perform the following steps;

```
%> cd $HOME
%> mkdir cds-sw
%> cd cds-sw
%> wget http://bmi.osu.edu/~oster/caGrid-1.2M1/RC2/caGrid-1.2M1-CDS-Client-rc2.zip
%> unzip caGrid-1.2M1-CDS-Client-rc2.zip
```

This unzips the file into a folder named `cds`.

4.6.2 Build CDS

Perform the following to build the CDS software component:

```
%> cd cds
%> ant clean all
```

4.6.3 Obtain CDS Certificates

You need the host certificates previously created for CDS from the Dorian server. Upload the CDS relevant certificates (Example: `cds-cert.pem` and `cds-key.pem`) from the Dorian server and then login to the CDS server and perform the following:

```
%> cd $HOME
%> mkdir certificates
```

Download the `cds cert.pem` and `cds-key.pem` files from Dorian to the certificates folder on the CDS server

4.6.4 Setup CDS Trust

The certificate (`cds-cert.pem`) must now be stored in the correct directory with a file name extension of .0 in order for Globus to trust the CA that issued the host certificate. On the CDS server, perform the following:

```
%> cd $HOME
%> mkdir .globus/certificates
%> cp $HOME/certificates/cds-cert.pem  $HOME/.globus/certificates/caCDScert.0
```

4.6.5 Deploy Globus

Now that the host credentials exist, you may configure the CDS Secure Tomcat container. To build out the Tomcat container on CDS, perform the following:

```
%> cd $GLOBUS_LOCATION
%> ant -f share/globus_wsrf_common/tomcat/tomcat.xml deploySecureTomat -Dtomcatdir
```

Where `<tomcat.dir>` is the absolute path to the Tomcat installation directory (the path for `CATALINA_HOME`). This was determined during the deployment planning phase. Using the example, the full path for this command would look like this:

```
%> ant -f share/globus_wsrf_common/tomcat/tomcat.xml
deploySecureTomat -Dtomcatdir=/usr/local/tomcat-5.0.28.cds
```

This deploys a subset of the files from the Java WS Core installation into the Tomcat container for CDS.

4.6.6 Secure Tomcat

Once complete, environment specific settings must be updated in the Tomcat container `server.xml` file in order to define the HTTPS connector. On the CDS server, perform the following:

```
%> cd $CATALINA_HOME/conf
```

1. Edit the `server.xml` file to add a HTTPS Connector in the <Service name='Catalina"> section of this file, for example:

```
<Service name="Catalina">
<Connector
  className="org.globus.tomcat.coyote.net.HTTPSConnector"
  port="8443" maxThreads="150" minSpareThreads="25"
maxSpareThreads="75"
  autoFlush="true"
  disableUploadTimeout="true" scheme="https"
  enableLookups="true" acceptCount="10" debug="0"
  cert="/local/home/tomcat-cds/certificates/cds-cert.pem"
  key="/local/home/tomcat-cds/certificates/cds-cert.pem"
  cacertdir="/local/home/tomcat-cds/.globus/certificates"/>
```

This example defines the paths based on the environment variables set during the deployment planning phase. Defining the cert and key attributes are optional. The `cacertdir` must be defined. This is the directory where the `CDS .0` certificate file was copied to in the previous steps.

Also in the `server.xml` file and HTTPSValve must be added.

2. Find the following section:

```
<Engine name='Catalina"…>
```

The section should look something like this:

```
<!-- Define the top level container in our container hierarchy -->
<Engine debug="0" defaultHost="localhost" name="Catalina">
```

3. Add the following line under the above `<Engine Debug "Catalina">` line in the `server.xml` file:

```
<Valve className="org.globus.tomcat.coyote.valves.HTTPSValve"/>
```

It may also be necessary to modify the `web.xml` file if you are using a non-standard port assignment for the Tomcat container for CDS. In your deployment planning, you determined the secure port for CDS would be **`18443`**. The standard port assignment is **`8443`**; therefore, in the deployment example, you would have to modify the `web.xml` file. If your CDS Tomcat is running on a port other than 8443, perform the following on the CDS server:

```
%> cd $CATALINA_HOME/webapps/wsrf/WEB-INF
```

4. Modify the `web.xml` file to update the default port assignment in the following section:

```
<web-app>
...
    <servlet>
        <servlet-name>WSRFServlet</servlet-name>
        <display-name>WSRF Container Servlet</display-name>
        <servlet-class>
            org.globus.wsrf.container.AxisServlet
        </servlet-class>
        <init-param>
            <param-name>defaultProtocol</param-name>
            <param-value>https</param-value>
        </init-param>
        <init-param>
            <param-name>defaultPort</param-name>
            <param-value>18443</param-value>
        </init-param>
        <load-on-startup>true</load-on-startup>
    </servlet>
...
</web-app>
```

4.6.7 Create a Globus Security Descriptor

To complete the configuration of the secure tomcat container for CDS, you must create a global security descriptor file. The file consists of the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<securityConfig xmlns="http://www.globus.org">
 <credential>
      <key-file value="PATH_TO_HOST_PRIVATE_KEY"/>
      <cert-file value="PATH_TO_HOST_CERTIFICATE"/>
   </credential>
</securityConfig>
```

1. Copy the above and then on the CDS server, perform the following:

   ```
   %> cd $HOME/.globus
   ```

2. Open an editor such as vi, defining a filename as you open the editor (Example: vi `globsecdescrip`) and paste the above details in this file. Edit the key and cert file paths to define the path where `cds-cert.pem` and `cds-key.pem` files are located and save the file. When complete, if following the folder/path examples provided in this install, the file would look something like this:

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <securityConfig xmlns="http://www.globus.org">
    <credential>
         <key-file value="/usr/local/tomcat-cds/certificates/
   cds-key.pem"/>
   ```

```
      <cert-file value="/usr/local/tomcat-cds/
certificates/cds-cert.pem"/>
   </credential>
</securityConfig>
```

4.6.8 Finalize Tomcat Security

The final step in configuring the secure Tomcat container for CDS is to add the `containerSecDesc` parameter to the `<globalConfiguration>` in the file called `server-config.wsdd` to tell Globus the location of the global security descriptor file created previously.

Perform the following on the CDS server:

```
%> cd $CATALINA_HOME/webapps/wsrf/WEB-INF/etc/
globus_wsrf_core
```

1. Edit the server-config.wsdd file to add the following line within the <globusConfiguration> parameters:

```
<globalConfiguration>
   <parameter name="containerSecDesc"
value="PATH_TO_YOUR_GLOBAL_SECURITY_DESCRIPTOR"/>
```

2. If you are following the folder/path conventions documented in this process, the line to add would look something like this:

```
<parameter name="containerSecDesc" value="/usr/local/
tomcat-cds/.globus/globsecdescrip”/>
```

4.6.9 Configure CDS

You are now ready to complete the configuration setup of CDS by updating the `cds.properties` file. This file contains parameters to define the host name, port connection and password details for connecting to the CDS database. The following parameters are in the cds.properties file:

- **gaards.cds.name** - Unique name of the CDS, distinguishing one CDS from another CDS on the same host. This will be used as the database name.
- **gaards.cds.max.delegation.path.length** - The maximum delegation path length of credentials issued by the CDS. A value of 0 specifies that credentials issued by the CDS cannot be delegated.
- **gaards.cds.db.host** - The host name of the CDS database.
- **gaards.cds.db.port** - The port to connect to the CDS database on.
- **gaards.cds.db.user** - The user id to use to connect to the CDS database.
- **gaards.cds.db.password** - The password to use to connect to the CDS database.
- **gaards.cds.dbkeymanager.key.encyption.password** - This property is needed if you are using the DB Key Manager (`org.cagrid.gaards.cds.service.DBKeyManager`), it specifies a password which is used to encrypt the keys of the delegated credentials in the database.

1. To edit the file, change to the path location where the CDS software was downloaded and unzipped in the first step. Change directories to the CDS folder and then change directories (cd) to the etc folder.
2. Open the `cds.properties` file with an editor (such as vi) and define the correct database name, port assignments, userid and password as obtained from the DBA.

   An Example of this file once updated:

```
#General Configuration
gaards.cds.name=ccts_cds
gaards.cds.max.delegation.path.length=0
# Database Configuration
gaards.cds.db.host=cbiodb580
gaards.cds.db.port=3641
gaards.cds.db.user=cctsbunduser
gaards.cds.db.password=ct7jz#35
#Key Manager Configuration
gaards.cds.dbkeymanager.key.encyption.password=password
```

4.6.10 Start CDS

To start CDS, follow these steps:

1. Login to the CDS server using the userid defined for CDS (Example tomcat-cds)

```
%> cd $CATALINA_HOME/bin
%> ./startup.sh
```

2. Once it has started up, be sure to look in `$CATALINA_HOME/logs/catalina.out` for any errors. From the folder where the service was started from, perform the following:

```
%> cd ../logs
%> more catalina.out
```

This completes the setup for CDS.

4.7 IndexService and GME

4.7.1 Launch the caGrid 1.1 Installer

To download the installer, follow these steps:

1. Login to the server to be installed.
2. At the $HOME directory (`%> cd $HOME`):

```
%> mkdir installer
%> wget http://gforge.nci.nih.gov/frs/download.php/2371/caGrid-1.1-installer-rc11-ncicb.zip
%> cd installer
%> unzip caGrid-1.1-rc1-installer.zip
```

3. From this same directory, launch the installer:

```
%> java -jar caGrid-1.1-installer.jar
```

This brings up a GUI installer.

4.7.2 Install the IndexService and GME

1. Select installation components.
   a. Accept the caGrid license and click **Next**.
   b. Select the **Install caGrid** checkbox. Select the **Install caGrid Services** checkbox. De-select all other check boxes.
   c. Click **Next**.
   d. Select the **Index Service** and **GME** check box and click **Next**.
   e. From the **Container Type** drop down select **Tomcat**"and click **Next**.
2. Next you are asked to specify a directory in which to install Ant.
   a. In the **Directory** text field enter ***{the path defined for your ANT_HOME variable}*** and click **Next**.

      If you followed the instructions and installed all the pre-requisite software prior to beginning the 1.1 Grid install process, ANT will already be installed and the `ANT_HOME` environment variable already set. If this is the case, leave the check box blank and press *Next*
3. Next you are asked to specify a directory in which to install Tomcat.
   a. In the *Directory* text field enter *the* ***{path defined for your CATALINA_HOME variable}*** and click **Next**.

      If you followed the instructions and installed all the pre-requisite software prior to beginning the 1.1 Grid install process, TOMCAT will already be installed and the `CATALINA_HOME` environment variable already set. If this is the case, leave the check box blank and press **Next**.
4. Next you are asked to specify a directory in which to install Globus. If you followed the pre-requisite instructions, the path for `GLOBUS_LOCATION` will already be defined
   a. In the **Directory** text field enter ***{the path defined for your GLOBUS_LOCATION variable}*** and click **Next**.
5. Next you are asked to specify a directory in which to install caGrid. If you followed the pre-requisite instructions, the environment variable for `USER_HOME` will already be set.
   a. In the **Directory** text field enter ***USER_HOME/.cagrid*** and click **Next**..
6. You are then asked if you would like to reconfigure caGrid for another target grid
   a. Leave the check box blank and press **Next**.
   b. Select the **NCICB Production Grid** from the Target Grid drop down and click **Next**.
   c. Click the **Start** button, the installer will begin to download and install Globus and caGrid

Note: This step may take a while to download and extract all the files, and build caGrid

d. Once the tasks have finished, click **Next**.

e. In the **Hostname** text field, enter the name of the host that will run the service

Example: ***cagrid-gme.nci.nih.govC***

f. Click **Next**.

g. In the **Shutdown** port text field enter the port assignment number for the service being installed (Example: ***8005***). In the ***"HTTPS" Port*** text field enter the port assignment number for the service being installed (Example: ***8443)***.

Note: Port Assignments will vary based on the service being installed and the planning done in. Using the planning example, the following *shutdown* and *HTTPS* port assignments were made for Dorian:

`IndexService/GME:38005/38443`

7. Edit the service metadata for your deployment. Be sure to provide:

   — Appropriate **Research Center Metadata** (including the Address, etc)

   — Appropriate **Research Center Points of Contact** (including a point of contact for support questions)

   a. Click **Next**.

8. In the next screen, edit the database connection information specifying the ***Database Hostname***, ***Database Port***, ***Database Name***, ***Database Username***, and ***Database Password*** for GME.

   a. Click **Next**.

   Note: This validates the JDBC connection settings (as they are needed later). If you get an error, be sure MySQL is installed and running with a username and password matching your specifications.

   Note: Be sure to pick a unique name for the database if you are using the same database server as other services.

   If there is a database with the same name in the database server, the next panel will indicate that the database will be destroyed.

   b. Press **Next**.

9. In the next screen, GME Standard Properties edit the following properties:

   Note: These are important! If they don't match your deployment, then GME won't work properly):

   — The `service.deployment.host` and set it to the host running GME (Example cagrid-gme.nci.nih.gov)

   — The `service.deployment.port` and set it to the port running GME (Example: ***38080***)

   — The *service.deployment.protocol* and set it to the protocol for the GME (http)

a. Then click **Next**.

b. Click the **Start** button to install the services as configured.

c. Once the services have finished installing click **Next**.

The installer will instruct you set the following environment variables: ***ANT_HOME***, ***GLOBUS_LOCATION***, and ***CATALINA_HOME***. If these were not set during the pre-requisite steps, **set these environment variables now.**

d. Click the **Finish** button and the click the **Close** button to close the installer.

4.8 Secure Application Containers

4.8.1 Secure Application Tomcats

Follow the instructions outlined in the CDS section on Securing Tomcat in order to secure the rest of the Tomcat instances related to applications and WebSSO. (See sections *Section 4.6.5 Deploy Globus*, *Section 4.6.6 Secure Tomcat*, *Section 4.6.7 Create a Globus Security Descriptor*, and *Section 4.6.8 Finalize Tomcat Security*).

4.8.2 Deploy SyncGTS

Run the caGrid installer to deploy SyncGTS on a per user, per node basis. If each Tomcat on each node is running under the same system user, then syncGTS should be deployed to only one Tomcat. If Tomcat is being run by different users, then syncGTS should be deployed to each Tomcat.

4.9 WebSSO

4.9.1 Build WebSSO

Follow the WebSSO installation documentation at:

http://www.cagrid.org/mwiki/index.php?title=WebSSO:1.2:Administrators_Guide

Follow the instructions through building, but do **not** perform the deploy step.

4.10 Inject the Suite UI

1. Get the CCTS WebSSO UI project from the Suite package at `ccts-ui-1-0/ccts-websso-ui.zip`.
2. Unzip it into a directory, for example, `ccts-ui`. Perform the following steps:
   a. Place `cas.war` from the WebSSO build directory into the input directory in the unzipped archive
   b. Run "ant all" from the unzipped directory
   c. The output directory will have the modified `cas.war`.

4.10.1 Deploy WebSSO

Place the output/cas.war in your `$CATALINA_HOME/webapps` directory and restart Tomcat.

5. Suite Installation

After the Suite pre-requisites have been installed or met, the next step is to install each of the individual applications. The following sections provide some background on each application, pointers to each installation guide, and any guidance needed to install the applications for a Suite deployment. Once installed, the final step is to configure the suite, which is described in the section after this.

Note: It is important that you deploy the AE Reporting System first. Otherwise, some configuration files may be overwritten.

5.1 Cancer Adverse Event Reporting System (caAERS)

5.1.1 Component Overview

The Cancer Adverse Event Reporting System (hereafter referred to as the AE Reporting System) is an open source, web-based application for documenting, managing, reporting, and analyzing adverse events (AEs). The system operates both as a repository for capturing and tracking routine and serious AEs (SAEs) and as a tool for preparing and submitting expedited AE reports to regulatory agencies. Currently, the AE Reporting System works with cancer prevention and therapeutic trials and can accommodate a range of intervention types, including investigational and commercial agents, radiation, surgery, and medical devices. Adverse events can be coded in the AE Reporting System using either CTCAE or MedDRA.

To help organizations stay in compliance with AE reporting regulations, the the AE Reporting System application comes loaded with a full complement of industry-standard AE reports, including the FDA MedWatch 3500A form, the CTEP AdEERS reports, and the NCI-DCP SAE form. In addition, the the AE Reporting System system features a powerful, state-of-the-art rules engine, which can capture a range of sponsor, institution, and protocol-level reporting requirements. Using these rules, the AE Reporting System can automatically determine if an adverse event requires expedited reporting and when and to whom the report must be submitted – for any of an organization's trials. The business rules used by the AE Reporting System can be authored within the application itself or imported from a library of approved rule sets.

the AE Reporting System also features an advanced email-based alert system that can be customized along a number of dimensions (message content, recipients, delivery times) to ensure that notifications and reminders are sent out as needed. Also included as part of the AE Reporting System is an easy-to-use report template generator, which allows users to build and customize reports.

the AE Reporting System can be deployed as a stand-alone application or as an integrated module within the Suite.

5.1.2 Component Installation

The the AE Reporting System installation guide can be found in the Suite package under the AE Reporting System-1-0. Follow the instructions in this guide to deploy the component.

5.2 Cancer Centralized Clinical Database (C3D) Grid Component

5.2.1 Component Overview

The C3D Connector is an example of a component that allows a legacy Clinical Data Management System (CDMS), or any other kind of legacy system, to interface with the suite of applications in version1.0 of the Suite. The C3D Connector is a component of the Suite, however the Clinical Database is not. For a basic understanding, both C3D Connector (a Suite application) and the Clinical Database (a vendor-provided solution resident at NCI-CBIIT) are reviewed here.

5.2.1.1 C3D Connector (a Suite component)

The C3D Connector provides the Suite community the ability to enroll patients and load labs into the CDMS, C3D in this case. The EnrollPatient service provides a mechanism for the greater community to enroll patients into studies maintained by C3D, without having to interact with the normal C3D User Interface. The service leverages the Data Capture API of Oracle Clinical to ensure the proper handling of data. The Lab Load service provides a mechanism that allows for the automatic processing and loading of Laboratory Test Result data into specific patients of studies maintained by C3D. This particular service leverages the current C3D Lab Loader process so that all of the required data qualification and validation procedures are followed.

5.2.1.2 Cancer Centralized Clinical Database (a vendor-supplied solution)

Cancer Central Clinical Database (hereafter called the Clinical Database) is a clinical trials data management system. The Clinical Database collects clinical trial data using standard case report forms (CRFs) based on common data elements (CDEs). The Clinical Database utilizes security procedures to protect patient confidentiality and maintain an audit trail as required by FDA regulations. The Clinical Database currently supports electronic submission of clinical trials data to the National Cancer Institutes (NCI) Clinical Data System (CDS) and the Clinical Trials Monitoring Service (CTMS/ Theradex). C3D consists of three web-based components: Oracle Clinical, for protocol building; Remote Data Capture, for data entry and management; and Integrated Review / Java Review, for real-time access to clinical data within and across clinical studies to authorized users.

5.2.2 Component Installation

The C3D Connector installation guide can be found in the Suite package under c3d-connector-1-0. Follow the instructions in this guide to deploy the component.

5.3 Cancer Central Clinical Participant Registry (C3PR)

5.3.1 Component Overview

The Cancer Central Participant Registry (hereafter referred to as the Participant Registry) Release 2 is a web-based application used for end-to-end registration of patients to clinical trials. This includes capturing the consent signed date, eligibility criteria, stratification, randomization, and screening. Clinical workflows are enabled by both subject- and study-centric views into the registration process. The Participant Registry can be run in a stand-alone mode where study definitions, investigators, study personnel, and sites are entered into the system, or the Participant Registry can be run in an integrated mode with the Suite. The Participant Registry also enables multi-site

clinical trials where registration information is entered locally at affiliate sites and the registration is completed by call-out to the coordinating site.

5.3.2 Component Installation

The Participant Registry installation guide can be found in the Suite package under c3pr-2-0. Follow the instructions in this guide to deploy the component.

5.4 caXchange

5.4.1 Component Overview

The caXchange is a generic platform for exchanging all types of clinical trial data and messages using service invocations and data exchange. caXchange supports routing raw lab data in a flat file format to a transformation service that converts the data into HL7 v3 messages and then routes the messages to a persistence service that stores the data in a database from which the Lab Viewer could query lab results. caXchange additionally allows for the necessary extensibility to meet the long-term objective as an enterprise interface engine. It includes an expanded message auditing service, clustering to show enterprise scalability for use in multi-center deployment, a messaging framework that allows for routing, transformation and handling of complex workflow scenarios, and a GUI for the client to allow the user to configure it more easily. Furthermore, caXchange support a service oriented architecture (SOA) to provide additional functionality currently not provided by caGrid including routing, logging, message handshaking, etc. as specifically defined by the Suite architecture team and in compliance with the Suite Technical Governance framework defined by the Suite team.

The two main components of caXchange are the Enterprise Service Hub and the Cancer Center Hub Client. The Enterprise Service Hub provides the routing and exchange of clinical trial messages and data for all the Suite applications and the hub client over caGrid. The Cancer Center Hub Client is used to transform lab data into HL7v3 and submit it to the hub for persistence.

5.4.2 Component Installation

The caXchange installation guide can be found in the Suite package under caxchange-1-0. Follow the instructions in this guide to deploy the component.

5.5 LabViewer

5.5.1 Component Overview

The Lab Viewer application was originally created for the CTMSi project, the predecessor to version 1.0 of the Suite. It allows users to query and view lab data that is persisted in a Clinical Trials Object Data System (CTODS) database. It also integrates with other Suite applications by sending and receiving messages from each of them. While the user interface of Lab Viewer was modified somewhat, the more significant enhancements for version 1.0 of the Suite include the following:

- Updated CTOM Lab Domain Model to handle multiple identifiers
- Updated Lab Viewer to handle the new model
- Updated Lab Loader service to handle the new model

- Added a service to receive the Participant Registry enroll patient message and persist to CTOM
- Provided ability to submit abnormal labs to the AE Reporting System as candidate adverse events
- Provided ability to submit lab information to C3D
- Provided hotlinking to other Suite applications
- Improved performance

5.5.2 Component Installation

The LabViewer installation guide can be found in the Suite package under labviewer-1-0. Follow the instructions in this guide to deploy the component.

5.6 Patient Study Calendar (PSC)

5.6.1 Component Overview

The Patient Study Calendar (hereafter referred to as the Study Calendar) is an open source, standards-compliant, web-based application that assists with the management of the activities of subjects on clinical trials. The Study Calendar provides the ability to create and edit a standard template to represent the activities defined by a study protocol, use this template to generate and view prospective calendars of subject activities, track the state of activities as a subject progresses through the study, and manage subject calendars as they change during a study. The Study Calendar also provides interfaces for managing access to data across a multi-site environment and balancing the workload of Subject Coordinators.

5.6.2 Component Installation

The Study Calendar installation guide can be found in the Suite package under PSC-1-4. Follow the instructions in this guide to deploy the component.

6. Configuration

6.1 Advertisement (Optional)

Each grid service can be optionally configured to advertise to the IndexService.

6.1.1 Configuring Deployed Services

1. In a deployed service, you can edit the file:

   ```
   $CATALINA_HOME/webapps/wsrf/WEB-INF/etc/cagrid_SERVICE/
   SERVICE_registration.xml
   ```

   Where SERVICE is the name of the service. For example:

   ```
   $CATALINA_HOME/webapps/wsrf/WEB-INF/etc/
   cagrid_registrationConsumer/
   RegistrationConsumer_registration.xml
   ```

2. Set the `IndexService URL`:

```
<ServiceGroupEPR>
    <wsa:Address>INDEX_SERVICE_URL</wsa:Address>
</ServiceGroupEPR>
```

3. Set the refresh rate of the service (for example, ***600***):

```
<RefreshIntervalSecs>REGISTRATION_REFRESH</
RefreshIntervalSecs>
```

4. Set the polling interval (for example, ***300***):

```
<agg:PollIntervalMillis>INDEX_REFRESH</
agg:PollIntervalMillis>
```

5. Restart Tomcat.

6.1.2 Configuring Undeployed Services

1. Edit the deploy.properties.
   a. Set `perform.index.service.registration` is ***true***.
   b. Set the `index.service.url.`
   c. Deploy the service.

6.2 Schema Registration (Optional)

XML Schemas can optionally be registered to the Global Model Exchange Service (GME). This is accomplished using the Introduce toolkit. Install Introduce using the caGrid installer. Launch Introduce by running "`ant introduce`" from the installation directory. Access the GME by clicking on the **Browse Data Types** button at the top (see *Figure 21*). XML Schemas can be uploaded by clicking the Upload tab, selecting

an XML Schema from the file system, and clicking upload (*Figure 22*). XML Schemas can be found in the Suite package under documentation/schemas.

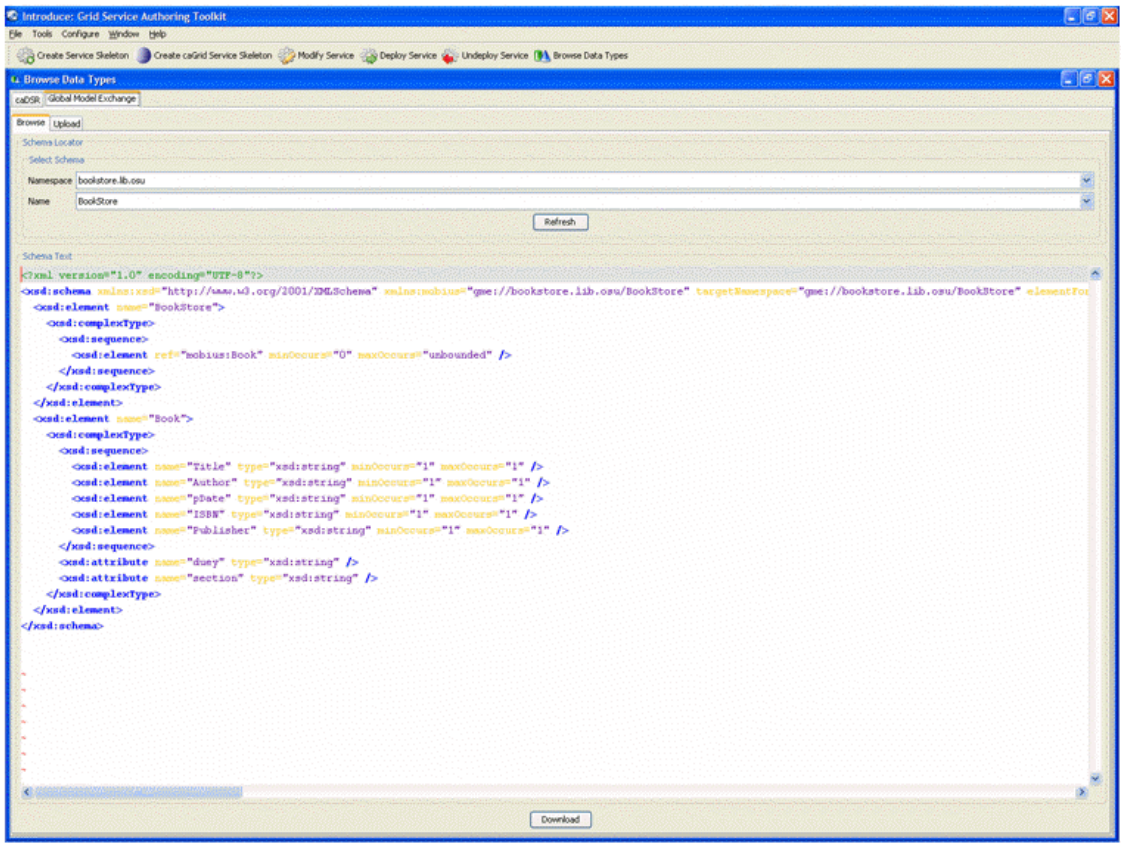

Figure 21 GME interface in "Browse Data Types" in Introduce

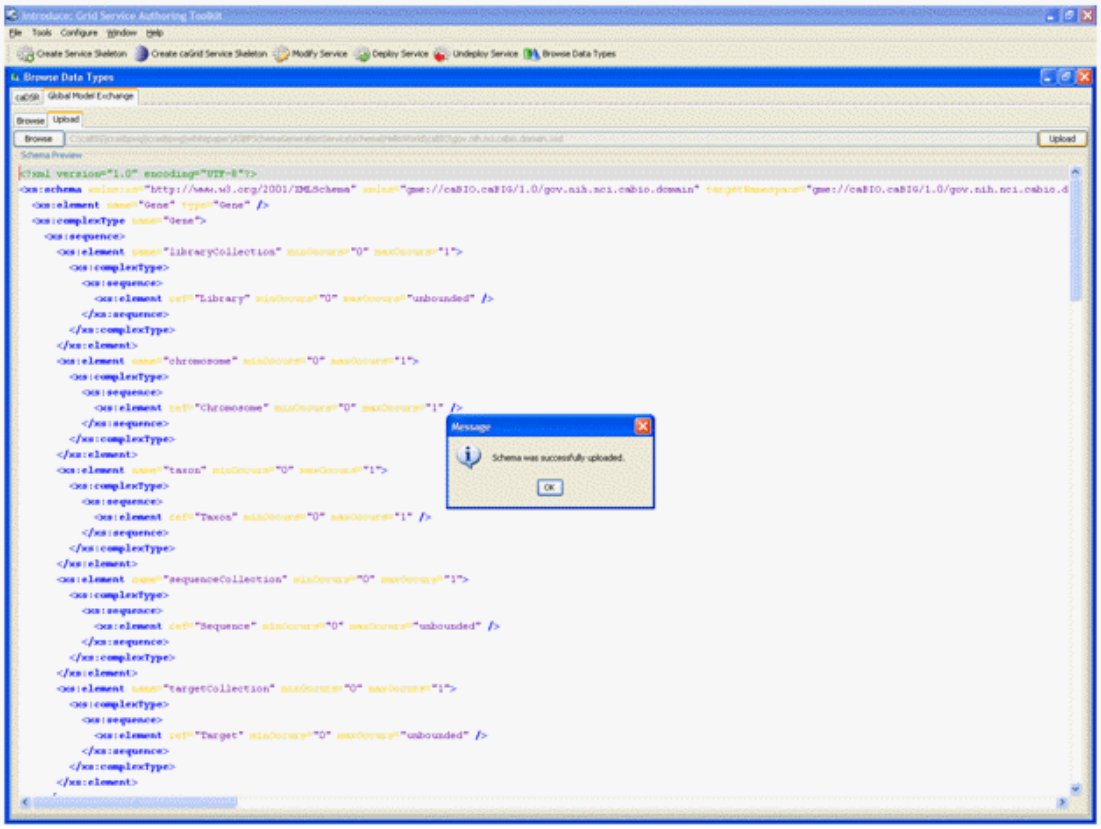

Figure 22 Upload example of an ßXSD to the GME

6.3 caXchange

Once caXchange and the rest of the applications are deployed, you will need to configure caXchange to point to the correct endpoints. Follow the instructions in the caXchange administration guide to accomplish this.

6.4 Inter-Application Linking (Hotlinks)

Hot-links need to be configured in the Participant Registry, the Study Calendar, the AE Reporting System, and LabViewer. Follow the instructions in the administration guides for each of these applications.

7. Validating the Installation

The installation can be validated by following the workflows identified in the caBIG™ Clinical Trials Suite End User Guide.

8. Troubleshooting

8.1 GSS-API Bad Certificate Error

8.1.1 Error

The following error is found in the Tomcat logs when Tomcat starts up, a user attempts to log in, or messages are attempting to be exchanged:

```
AxisFault
 faultCode:
{http://schemas.xmlsoap.org/soap/envelope/}Server.userException
 faultSubcode:
 faultString: org.globus.common.ChainedIOException: Authentication
failed [Caused by: Failure unspecified at GSS-API level [Caused by:
Bad
certificate (The signature of
'O=NIH,OU=NCI,OU=NCICB,OU=caGrid,CN=host/cbiovdev5035.nci.nih.gov'
certificate does not match its issuer)]]
 faultActor:
 faultNode:
 faultDetail:
        {http://xml.apache.org/axis/}stackTrace:Authentication failed.
Caused by Failure unspecified at GSS-API level. Caused by
COM.claymoresystems.ptls.SSLThrewAlertException: Bad certificate (The
signature of
'O=NIH,OU=NCI,OU=NCICB,OU=caGrid,CN=host/cbiovdev5035.nci.nih.gov'
certificate does not match its issuer)
        at COM.claymoresystems.ptls.SSLConn.alert(SSLConn.java:243)
        at
COM.claymoresystems.ptls.SSLHandshake.recvCertificate(SSLHandshake.java:
299)
        at
COM.claymoresystems.ptls.SSLHandshakeClient.processTokens(SSLHandshak
eCl
ient.java:128)
...
```

8.1.2 Solution

Remove the following jars from each web application and place a single copy of each in the Tomcat shared lib folder:

- `cog-jglobus.jar`
- `cryptix*`
- `jce-jdk13-125.jar`
- `jgss.jar`
- `puretls.jar`

9. Glossary

Term	*Definition*
CSM	Common Security Module
IdP	Identity Provider
API	Application Programming Interface
CDE	Common Data Element
GUI	Graphical User Interface
UML	Unified Modeling Language
JMS	Java Message Service
API	Application Programming Interface
caBIG	cancer Biomedical Informatics Grid
DAO	Data Access Objects
HTTP	Hypertext Transfer Protocol
JDBC	Java Database Connectivity
JSP	JavaServer Pages
ORM	Object Relational Mapping
RDBMS	Relational Database Management System
SDK	Software Development Kit
HASTE	High-level Automated System Test Environment
WSRF	Web service resource framework
ESB	Enterprise Service Bus
AJAX	Asynchronous JavaScript a
BC	Binding Component
SE	Service Engine
NMR	Normalized Message Router
JAAS	Java Authentication and Authorization Service
GAARDS	Grid authentication and authorization with Reliably Distributed Services
DWR	Direct Web Remoting

10. Supplementary Resources

The following documents may be useful to end users or technical staff implementing the Suite.

Document Title	***URL***
CCTS 1.0 Scope Document (product specification)	https://gforge.nci.nih.gov/plugins/scmcvs/cvsweb.php/ccts/documentation/requirements/CCTS%20Software%20Product%20Spec%2020070727.doc?cvsroot=ccts
CCTS 1.0 Interoperability Scenarios and Activity Diagrams	https://gforge.nci.nih.gov/plugins/scmcvs/cvsweb.php/ccts/documentation/requirements/CCTS%201.0%20Interoperability%20Scenarios%20and%20Activity%20Diagrams.doc?cvsroot=ccts
Information Modeling and Messaging Specifications as a Foundation to the caBIG CCTS	https://gforge.nci.nih.gov/plugins/scmcvs/cvsweb.php/ccts/documentation/design/ccts_messaging_and_information_models.doc?cvsroot=ccts
CCTS 1.0 Architecture Document	https://gforge.nci.nih.gov/plugins/scmcvs/cvsweb.php/ccts/documentation/design/ccts_architecture.doc?cvsroot=ccts
CCTS Integration Guide	https://gforge.nci.nih.gov/plugins/scmcvs/cvsweb.php/ccts/documentation/design/ccts_integration_guide.doc?cvsroot=ccts
CCTS Interface Specification	https://gforge.nci.nih.gov/plugins/scmcvs/cvsweb.php/ccts/documentation/design/ccts_interface_specification.doc?cvsroot=ccts
CCTS 1.0 Installation Guide	https://gforge.nci.nih.gov/plugins/scmcvs/cvsweb.php/ccts/documentation/installation/ccts_installation_guide.doc?cvsroot=ccts
CCTS 1.0 Release Notes	https://gforge.nci.nih.gov/plugins/scmcvs/cvsweb.php/ccts/documentation/installation/ccts_release_notes.txt?cvsroot=ccts

Table 1.7 Additional Suite resources