**caTissue Suite v1.2**

***Integrated Biospecimen Banking***

***Information System***

**deployment Guide**

Copyright and License page

**Copyright Notice.** Copyright 2010 Washington University in St. Louis (“caBIG® Participant”). caTissue 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, to (i) use, install, disclose, access, operate, execute, reproduce, copy, modify, translate, market, publicly display, publicly perform, and prepare derivative works of the caBIG® Software in any manner and for any purpose, and to have or permit others to do so; (ii) make, have made, use, practice, sell, and offer for sale, import, 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 Washington University in St. Louis.” 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 ”Washington University in St. Louis”, “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 WASHINGTON UNIVERSITY IN ST. LOUIS 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.

Document Change History

|  |  |  |  |
| --- | --- | --- | --- |
| Version Number | Date | Description | Contributor |
| 1.0 | 1/15/2011 | Updates for caTissue Suite 20 deployment | caTissue Release 1.2 Development Team , Dave Mulvihill |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **Contacts and Support** | | |
| KnowledgeCenter Main Page | <https://cabig-kc.nci.nih.gov/Biospecimen/KC/index.php/Main_Page> |
| KnowledgeCenter Discussion Forum | <https://cabig-kc.nci.nih.gov/Biospecimen/forums/> |
| KnowledgeCenter Contact | [tbpt\_kc\_support@mga.wustl.edu](mailto:tbpt_kc_support@mga.wustl.edu) |

Contents

[Chapter 1. Introduction 6](#_Toc288205332)

[1.1 Overview of caTissue Suite 6](#_Toc288205333)

[1.1.1 Document Text Conventions 7](#_Toc288205334)

[1.2 Deployment Considerations 7](#_Toc288205335)

[1.3 Software Prerequisites 8](#_Toc288205336)

[Chapter 2. Deploying the Web Application 9](#_Toc288205337)

[2.1 Installation Prerequisites 9](#_Toc288205338)

[2.2 Downloading caTissue Suite 9](#_Toc288205339)

[2.3 Performing Pre-Installation Configuration 10](#_Toc288205340)

[2.4 Creating Database Scripts 18](#_Toc288205341)

[2.5 Deploying caTissue Suite 20](#_Toc288205342)

[2.5.1 New Installation 20](#_Toc288205343)

[2.5.2 Upgrade 21](#_Toc288205344)

[2.5.3 Re-deploying the caTissue Application 22](#_Toc288205345)

[2.6 Performing Post Installation Configuration 22](#_Toc288205346)

[2.6.1 Configuring JBoss (Libraries, Steps to Minimize Security Risks, etc.) 22](#_Toc288205347)

[2.6.2 Configuring caTissue Suite for Keyword Search 30](#_Toc288205348)

[2.6.3 Configuring JBoss Server to deploy caTissue using HTTPS 30](#_Toc288205349)

[2.7 Configuring CSM 31](#_Toc288205350)

[2.8 Starting and shutting down the Application Server 33](#_Toc288205351)

[2.9 Accessing the Web Application 34](#_Toc288205352)

[2.10 Deployment Errors 34](#_Toc288205353)

[Chapter 3. caTIES (v2.3) Integration in Suite 37](#_Toc288205354)

[3.1 Downloading MMTx Library and data.zip 39](#_Toc288205355)

[3.2 Deploying Gate Library 39](#_Toc288205356)

[3.3 Configuring caTIES 39](#_Toc288205357)

[3.4 Deploying caTIES 43](#_Toc288205358)

[3.5 Running caTIES pipeline 43](#_Toc288205359)

[3.5.1 Running the Report Loader Server 49](#_Toc288205360)

[3.5.2 Running the Report De-identification Server 50](#_Toc288205361)

[3.5.3 Running the Report Concept-Code Server 51](#_Toc288205362)

[Chapter 4. Testing the System 54](#_Toc288205363)

[4.1 Running the Test Case Suite (API) 54](#_Toc288205364)

[4.1.1 Running caCORE CSM Enabled Test Case Suite 54](#_Toc288205365)

[4.1.2 Getting Test Suite Result and Detailed Report 55](#_Toc288205366)

[4.1.3 Source code of Test Cases 55](#_Toc288205367)

[4.2 Running Test Cases Suite (API) with HTTPS 55](#_Toc288205368)

[4.2.1 Configure Client for HTTPS 55](#_Toc288205369)

[Chapter 5. Private/Public Data Store 57](#_Toc288205370)

[5.1 Deploying the Public Database 57](#_Toc288205371)

[5.2 Deploying the Web Application on the Public Database 61](#_Toc288205372)

[5.3 Known Issues 61](#_Toc288205373)

[Chapter 6. Deploying caTissue caGRID Data Service 63](#_Toc288205374)

[6.1 Background 63](#_Toc288205375)

[6.2 Prerequisites 64](#_Toc288205376)

[6.3 Requesting a caGrid Host Certificate 64](#_Toc288205377)

[6.4 Deploying and Securing Globus 67](#_Toc288205378)

[6.5 Installing a caTissue service: 69](#_Toc288205379)

[6.6 Running the caGrid test queries: 71](#_Toc288205380)

[Appendix A – MySQL Case Sensitivity on Linux 74](#_Toc288205381)

[MySQL Case Sensitivity Issue on Linux 74](#_Toc288205382)

[Appendix B – Clinical Annotation Forms List 75](#_Toc288205383)

[Form Names 75](#_Toc288205384)

[Appendix C – Steps for deploying caTissue with IDP 77](#_Toc288205385)

[Idp Configuration 77](#_Toc288205386)

[Idp Deployment 79](#_Toc288205387)

[Appendix D - Redeploying the application after creating a Local Extension 80](#_Toc288205388)

[Creating a New Form 80](#_Toc288205389)

[Updating an Existing Form 80](#_Toc288205390)

[Appendix E - Verification of upgrade data 81](#_Toc288205391)

[Verification of MySQL Data 81](#_Toc288205392)

[Verification of Oracle Data 81](#_Toc288205393)

# Introduction

This guide is intended for the technical IT staff, it describes steps involved in deploying caTissue Suite (caTissue) application. The document covers:

1. Deploying the web application
2. caTIES-like component integration
3. Testing the application using API
4. Public-Private data store
5. Deploying caTissue caGrid Data Service

This chapter gives you a brief overview of caTissue Suite. It also describes the various deployment considerations and software prerequisites.

## Overview of caTissue Suite

caTissue Suite is the *next-generation* tissue banking application which integrates the functionality of the main components developed with the community under thecaBIG® TBPT (Tissue Banks and Pathology Tools) Workspace. The tool will make it easier for researchers to locate and analyze tissue specimens for use in cancer research that is based on tissue, clinical, and genomic characteristics. As part of the initial goal, the following three applications were developed in year 1 and year 2 of the program. These applications were subsequently rolled into caTissue Suite.

**caTissue Core:** This application is used to track multiple specimens from the same patient or participant, create and track refined materials (RNA, DNA) that are used for molecular analysis, and distribute specimens.

**caTissue Clinical Annotations(CA):** Stores and queries pathology annotations for breast, prostate, and melanoma cases. Future iterations would also cover other cancer types and information systems. For example: clinical pathology information systems, tumor registry, and so on.

**caTIES (v2.3):** Automates the process of coding, storing, and retrieving data from free-text pathology reports.

caTissue Suite is an integrated system that can seamlessly perform tasks across the three applications mentioned above.

Allthree applications have been independently developed and have their own UML(Unified Modeling Language) data models. In addition, while caTissue Core and CA are web based applications, caTIES is a Java thick client application. Thus, in the integrated solution, caTIES v2.3has been re-engineered to utilize the common object model and the existing caTIES user interface will be utilized for administrative functions. For example: QA or QC of de-identification and concept coding), and the essential functionality required for end user researchers will be provided through the integrated web interface.

### Document Text Conventions

The following table describes the various conventions used in this manual.

|  |  |
| --- | --- |
| Courier typestyle | Used for filenames, directory names, commands, file listings, source code examples, and anything that would appear in a Java program. For example: methods, variables, and classes. |
| **CAUTION:** | Used for cautioning the user before performing certain tasks. |
| **Note:** | Used for user attention. |

## Deployment Considerations

The following diagram outlines the steps involved in the deployment process.

1. **Pre-install**

Configure install.properties in case of fresh deployment and upgrade.properties in case of upgrade.

Configure the images and text on the home page.

Create database and user.

1. **Prerequisites**

JDK 1.6

JBoss 5.1.0 GA

MySQL 5.1.x or Oracle 10.2.0.2.0

Ant 1.7

**3. Install**

Deploy or upgrade caTissue.

Deploy caTIES Services.

**4. Post Install**

Configure caTissueCore\_Properties.xml.

Configure JBoss to run on HTTPS

Start JBoss.

The following are some important considerations in deploying caTissue Suite:

* Do you want to deploy the web application and database server on the same machine or different machines?
* Where to deploy the caTIES service (if installing).

Our recommendation is to deploy the web application and the database on different machines. If you are interested in deploying the caTIES services, it can be deployed on a third machine or on the same machine where the caTissue Suite database resides.

* Section Chapter 2 - Deploying the Web Applicationdescribes the process for caTissue web application deployment.
* Section Chapter 3 - caTIES Integration in Suite describes the process for caTIES services deployment.

## Software Prerequisites

The following table describes the technology stack supported by caTissue Suite.

Table : Software Prerequisites

| **Software Element Name** | **Version** | **Type** |
| --- | --- | --- |
| Windows | Microsoft Windows XP Professional Version 2002 Service Pack 2 | Server and Client |
| Linux | RedHat 9 or RedHat Enterprise ES/AS 2.1 or higher | Server |
| Ant | 1.7 | Build tool |
| JBoss | 5.1.0GA | Application server |
| JDK | 1.6 | Java |
| Oracle | 10.2.0.2.0 | Database – Server and client  **Note:** Oracle client to be installed on the machine which is hosting JBoss server.  Oracle database server supports only on Red Hat Enterprise ES/AS 2.1 or higher |
| MySQL | 5.1.x | **DatabaseNote**:Needto set systemvariable (max\_allowed\_packet) in **my.ini**file for supporting file type attributes in Dynamic extensions with file size greater than 1 GB etc.  For example: **max\_allowed\_packet = 1G** |
| Internet Explorer | 8.0 | Web browser |
| Mozilla Firefox | 3.6.3 | Web browser |
| Safari (Mac) | 5.0 | Web browser |
| MMTx |  | Needed for caTIES. Download from  http://catissuecore.wustl.edu/caties\_datafiles/ |
| NCI Metathesaurus |  | Needed for caTIES. Download from  http://catissuecore.wustl.edu/caties\_datafiles/ |

# Deploying the Web Application

This section describes the steps to deploy the web application and database. The section is divided into the following steps:

* Installation Prerequisites
* Downloading caTissue Suite
* Performing Pre-Installation Configuration
* Creating Database Scripts
* Deploying caTissue Suite
* Performing Post-Installation Configuration
* Configuring CSM
* Starting and Shutting down the Application Server
* Accessing the Web Application
* Deployment Errors

## Installation Prerequisites

The following table describes the installation prerequisites forcaTissue Suite.

Note: You only need to download the files for the corresponding Database (MySQL or Oracle) you will be using.

Table : Software Prerequisites

| **Software Name** | **Version** | **URL** |
| --- | --- | --- |
| Java (JRE and JDK) | 1.6 | http://www.oracle.com/technetwork/java/javase/downloads/index.html  **Note:**Most machines already have JRE installed on them. However, to deploy and run caTissue you need JDK and not only JRE. You can verify this by checking that the javac command runs successfully from the command line. |
| JBoss | 5.1.0 GA | <http://labs.jboss.com/jbossas/downloads/>  Note: The JBoss server is required only in case of upgrade. In case of fresh installation, it is provided in the installable zip. |
| MySQL  Database | 5.1.x | http://downloads.mysql.com/archives.php?p=mysql-5.1 |
| Oracle Database | 10.2.0.2.0 | [http://www.oracle.com](http://www.oracle.com/) |
| ANT | 1.7 | <http://ant.apache.org/> |

## Downloading caTissue Suite[TBD]

<https://gforge.nci.nih.gov/frs/?group_id=689>

Once you have downloaded the caTISSUE\_SUITE\_v12\_Installable\_Pack.zip, extract the contents of the file to any directory. This folder will contain two additional compressed files; caTissue\_Suite\_v1.2\_Installable and caTissue\_Suite\_API\_Client\_v1.2\_Installable.

Unzip the caTISSUE\_Suite\_v2.0\_Installable.zip, extract the contents of the file to any directory. This folder will be referred to as CATISSUE\_HOME in this document.

Unzip the caTissue\_Suite\_API\_Client\_v1.2\_Installable.zip, extract the contents of the file to any directory. This folder will be referred to as CATISSUE\_CLIENT\_HOME in this document.

If you are interested in buildingthe caTissue application from source code, please refer to documentation links on the caTissue Suite 1.2 tools page on the Tissue/Biospecimen Banking and Technology Tools wiki <https://cabig-kc.nci.nih.gov/Biospecimen/KC/index.php/CaTissueSuite1.2> .

The release files are available <https://gforge.nci.nih.gov/frs/?group_id=689>under *caTissue Suite v1.2* package.

**Caution:** Do not use directories with space in the name. For example: do not place the caTissue Suite directory in the /Program Files directory.

The following table lists the files in the installation folder and its description:

Table : Installation Folder Files

| **File Name** | **Description** |
| --- | --- |
| install.properties/upgrade.properties | Property file in which all the application configuration parameters are listed. |
| build.xml | ANT script used to deploy the application on a JBoss server. This also creates the database schema for Oracle or MySQL. |
| /modules | Folder containing various caTissue modules |
|  |  |
| /SQL | Folder containing SQL scripts required for deployment |
| README.txt | Readme file describing the installable files |
| bda-utils | Folder containing all the files related to BDA |
| common | Folder containing the common files |
| tools | Folder containing the installables e.g, JBoss server (in case of fresh installable zip) |
| project.properties | Property file in which project related configuration parameters are listed. E.g., MySQL version, Java version. |
| properties.template | Template describing the properties of the install.properties/upgrade.properties file. |

**Note:**Confirm that the environment variables ANT\_HOME and JAVA\_HOME are set, and that the system PATH includes the path for ANT\_HOME/binandJAVA\_HOME/bin.

## Performing Pre-Installation Configuration

Unzip the caTISSUE\_Suite\_v2.0\_Installable.zip, extract the contents of the file to any directory. This folder will be referred to as CATISSUE\_HOME in this document.

Before installing the application, all the parameters required for the process has to be defined by the user in the following files.

### New Installation

**File:**install.properties

**Location:**CATISSUE\_HOMEThe following table explains the parameters in this file, along with its default and permissible values.

Table : Installation configuration parameters

| **Property Name** | **Description** |
| --- | --- |
| application.base.path.linux | Description: This property should be set to the path where your application server should be installed on Linux.  Default Value: REPLACE\_VALUE  Permissible Values: N/A  For example:  application.base.path.linux=/user/local/jboss  **Note:** The path must be separated by ‘/’ and not ‘\’ |
| application.base.path.windows | Description: This property should be set to the path where your application server should be installed on windows.  Default Value: REPLACE\_VALUE  Permissible Values: N/A  For example:  application.base.path.windows=C:/apps/caTissue |
| authentication.type | Description: Used for setting up the login-config.xml.  Default Value: db  Permissible Values: : db or ldap  Note: No need to change the default value. This property is specific to CSM. If application uses the CSM for authentication then it is required to mention the authentication type which is used in application. |
| env.name | Description: The name of the environment where the application will be deployed. Do not need to change.  Default Value: local  Permissible Values: internal, external, dev, qa, stage or prod. |
| database.re-create | Description: This property when set will drop and recreate the database.user and database.name  Default Value: false  Permissible Values: true or false |
| database.system.user | Description: Required If database.re-create is set. This property refers to the database administrator who has the schema and user generation privileges on the database.  Default Value: REPLACE\_VALUE  Permissible Values: N/A  Note: Leave this property blank if no need to generate the schema. |
| database.system.password= | Description: The password used to authenticate the database administrator user given in the above property.  Default Value: REPLACE\_VALUE  Permissible Values:N/A  Note: Leave this property blank if no need to generate the schema. |
| database.system.url | Description: The JDBC connection url for the system user.  Default Value:  jdbc:${database.type}://${database.server}:${database.port}/  Permissible Values:N/A |
| database.drop-schema | Description: Set this property to true if you want to drop the schema. No need to change the value of this property.  Default Value: true  Permissible Values: : true or false |
| database.type | Description: The database type used in the application.  Default Value: N/A  Permissible Values: : ORACLE or MYSQL |
| database.server | Description: The hostname or IP where the database resides.  Default Value: N/A  Permissible Values: NA |
| database.port | Description: The port number to connect with the database server.  Default Value: N/A  Default port for Oracle: 1521  Default port for MySQL: 3306  Permissible Values:N/A |
| database.name | Description: The name of the database. Specify the same name that you have specified while creating the database.  Default Value: None  Permissible Values:N/A |
| database.user | Description: The user the application uses to connect to the database.  Default Value: None  Permissible Values:N/A |
| database.password | Description: The password used to authenticate the database user specified in the above property.  Default Value: None  Permissible Values: N/A |
| database.url | Description: JDBC url to connect to the database through application.  Default Value: None  Permissible Values:  For Oracle:  jdbc:oracle:thin:@${database.server}:${database.port}:${database.name}  For MySql databases:  jdbc:mysql://${database.server}:${database.port}/${database.name} |
| oracle.tns.name | Description: Specify the tns entry for the database used. This property is used only in case of Oracle database.  Default Value: None  Permissible Values: N/A |
| jboss.server.hostname | Description: Hostname or IP address of the machine on which the JBoss server will be running.  Default Value: None  Permissible Values: N/A |
| jboss.server.name | Description: Specify the server configuration name of jboss where application has to be deployed. This is useful if administrator wants to run multiple applications on same jboss. Note: The path must be separated by ‘/’ and not ‘\’  Default Value: default  By default it is set to 'default' configuration, the application will be installed in JBOSS\_HOME/server/default  Permissible Values: N/A |
| jboss.server.port | Description: This property specifies theport number on which JBoss server is running. It should not be specified in case the application is being deployed on an Apache Front ended server.  This property is also used for deploying Gate for caTIES. For deploying caTIES, this property needs to be specified even if the application is deployed on an Apache front ended server.  Default Value: 8080  Permissible Values: N/A |
| jboss.container.secure | Description: Should be set to yes if JBoss is deployed as HTTPS.  Default Value: N/A  Permissible Values: yes/no |
| deploy.cas.on.catissue.jboss | Description: This property specifies if the CAS server for authentication is to be deployed on the same server as caTissue application (specified by the property application.base.path.linux/application.base.path.windows). If this property is set to true, the CAS authentication server gets deployed on the same server as catissue application. If it is set to false, then the subsequent CAS related properties are required to specify the details of CAS server deployed on some other server.  Default Value: true  Permissible Values: true/false |
| cas.jboss.server.host | Description: This property specifies the host name of the server on which CAS application has been deployed. This property is required only when deploy.cas.on.catissue.jboss is set to false.  Default Value: N/A  Permissible Values: N/A |
| cas.jboss.port | Description: This property specifies the port number of the server on which CAS application is available. This property is required only when deploy.cas.on.catissue.jboss is set to false.  Default Value: N/A  Permissible Values: N/A |
| cas.jboss.container.secure | Description: This property specifies if the server on which CAS is available needs to be accessed using a secure link (an https URL).  Default Value: no  Permissible Values: yes/no |
| cas.server.keystore.path | Description: This property specifies the path to the keystore file if the server on which CAS is available needs to be accessed using a secure link (an https URL).  Default Value: N/A  Permissible Values: N/A |
| email.administrative.emailAddress | Description: Email address of the administrator. This could be an email alias if there is more than one administrator.  Default Value: N/A  Permissible Values: N/A |
| email.sendEmailFrom.emailAddress | Description: Email address used to send email messages from the deployment script.  Default Value: N/A  Permissible Values: N/A |
| email.mailServer | Description: The mail server used to send email messages.  Default Value: N/A  Permissible Values: N/A |
| use.email.commonpackage.config | Description: It should be set to true if you want to send email notification on server exceptions.  Default Value: false  Permissible Values:false,true |
| email.sendEmailTo.emailAddress | Description: Server exception’s email recipients. This could be an email alias or comma separated email addresses.  Default Value: N/A  Permissible Values: N/A |
| email.admin.support.emailAddress | Description: Email address of the administrator. This would be included in case email.sendEmailTo.emailAddress is empty or invalid  Default Value: N/A  Permissible Values: N/A |
| email.sendEmailFrom.name | Description: The sender name for server exception notification email.  Default Value: N/A  Permissible Values: N/A |
| email.exception.subject | Description: The subject line for server exception notification email.  Default Value: System Exception Occurred.  Permissible Values: N/A |
| session.timeout | Description: The web application inactivity timeout interval in minutes. Set this to a very high value (for example, 100) if you do not want your users to be timed out ever.  Default Value: 30  Permissible Values: Numeric Value |
| app.additional.info | Description: This property specifies additional information that needs to be displayed on the home page along with the application information, to identify the instance without referring to URls. This property can be used to put information like ‘SandBox Instance’, ‘Production Instance’  Default Value: N/A  Permissible Values: N/A |
| first.admin.department | Description: The first department that will be created and used in creating the first administrator user.  Default Value: N/A  Permissible Values: N/A |
| first.admin.institution | Description: The first institution that will be created and used in creating the first admin user.  Default Value: N/A  Permissible Values: N/A |
| first.admin.cancerresearchgroup | Description: The first cancer research group that will be created and used while creating the first admin user.  Default Value: N/A  Permissible Values: N/A |
| first.admin.emailAddress | Description:  Email address used in creating the first admin user. It will also be the login name for the first user.  Default Value: N/A  Permissible Values: N/A |
| first.admin.password | Description:  The password of the first admin user of the application  Default Value: N/A  Permissible Values: N/A |
| caties.mmtx.home | Description: The folder in which the MMTx library is installed on the machine on which you are going to run the caTIES concept coder service. This property is only needed when deploying the concept coder server of the caTIES-like component.  Default Value: N/A  Permissible Values: N/A |
| show.hide.forms.based.on.CPs | Description: This property is used during the execution of target 'show\_hide\_forms\_based\_on\_CPs', which is called in the deployment process. The task is executed only when the value of the property is set to true. The task is used to associate entities/categories with particular CPs. It takes an XML file 'Show\_Hide\_Forms.xml' as input. The XML has to be modified depending on the associations we want between entities/categories and CPs. The entities/categories when associated to a CP are not visible to any other CP.  Default Value: false  Permissible Values: true, false |
| Application.url | Description: The application URL is required for creation of Annotation Forms using form import feature of Dynamic Extensions. For importing the forms, it is necessary for the server to be up and running and the URL specified in this property is used for uploading the forms into the application.  Default Value: N/A  Permissible Values: Application URL on which the categories need to be created. (http://127.0.0.1:8080/catissuecore) |
| cacore.deployable.location | Description: Specify the folder where the caCORE API jars of DE model is located. The default DE model jars are available at CATISSUE\_HOME/cacore\_deployable folder  Default Value: ./cacore\_deployable  Permissible Values: N/A |
| exclude.entitygroup | Description: Comma separated Entity Group names which are to be excluded from generating the caCORE.  Default Value: None  Permissible Values: N/A |
| include.entitygroup | Description: Comma separated Entity Group names which are only to be included for generating the caCORE.  Default Value: N/A  Permissible Values: N/A |
| saved.query.owner | Description: This property specifies the login name of the user who should be made the owner of the old saved queries (pre v1.2).  This property is required to support the shared query feature (which is introduced in v1.2) for old saved queries.  Default Value: N/A  Permissible Values: N/A |

|  |  |  |
| --- | --- | --- |
| **IDP Properties** | | **Only needed if deploying caTissue with an ID Provider** |
| csm.database.type | | Description: The database type used in the application.  Default Value: N/A  Permissible Values: : mysql, oracle |
| csm.database.host | | Description: Thehostname or IP address of the machine on which the database server is running.  Default Value: localhost  Permissible Values: N/A |
| csm.database.port | | Description: Theport number to connect with the database server.  Default Value:  Default port for MySQL: 3306  Default port for Oracle: 1521  Permissible Values: N/A |
| csm.database.name | | Description: The name of the database. Specify the same name that you have specified while creating the database.  Default Value: N/A  Permissible Values: N/A |
| csm.database.username | | Description: The username used to connect to the database.  Default Value: N/A  Permissible Values: N/A |
| csm.database.password | | Description: The password used to authenticate the database user.  Default Value:N/A  Permissible Values: N/A |

**File: project.properties**

| **Property Name** | **Description** |
| --- | --- |
| mysql.minimum.version | Description: This property should be set to the version of MySQL installed on the deployment machine.  Default Value: REPLACE\_VALUE  Permissible Values: N/A |

### Upgrade

File:upgrade.properties

**Location:**CATISSUE\_HOMEThe following table explains the parameters in this file, along with its default and permissible values.

| **Property Name** | **Description** |
| --- | --- |
| application.base.path | Description: This property should be set to the path where your application server is residing on the machine.  Default Value: REPLACE\_VALUE  Permissible Values: N/A  For example:  application.base.path=/user/local/jboss  **Note:** The path must be separated by ‘/’ and not ‘\’ |
| jboss.home | Description: This property should be set to the home of your application server.  Default Value: REPLACE\_VALUE  Permissible Values: N/A  For example:  application.base.path =/user/local/jboss/jboss-5.1.0.GA |
| database.type | Description: The database type used in the application.  Default Value: N/A  Permissible Values: : ORACLE or MYSQL |
| database.server | Description: The hostname or IP where the database resides.  Default Value: N/A  Permissible Values: NA |
| database.port | Description: The port number to connect with the database server.  Default Value: N/A  Default port for Oracle: 1521  Default port for MySQL: 3306  Permissible Values:N/A |
| database.name | Description: The name of the database. Specify the same name that you have specified while creating the database.  Default Value: None  Permissible Values:N/A |
| database.user | Description: The user the application uses to connect to the database.  Default Value: None  Permissible Values:N/A |
| database.password | Description: The password used to authenticate the database user specified in the above property.  Default Value: None  Permissible Values: N/A |
| database.url | Description: JDBC url to connect to the database through application.  Default Value: None  Permissible Values:  For Oracle:  jdbc:oracle:thin:@${database.server}:${database.port}:${database.name}  For MySql databases:  jdbc:mysql://${database.server}:${database.port}/${database.name} |
| oracle.tns.name | Description: Specify the tns entry for the database used. This property is used only in case of Oracle database.  Default Value: None  Permissible Values: N/A |
| jboss.server.hostname | Description: Hostname or IP address of the machine on which the JBoss server will be running.  Default Value: None  Permissible Values: N/A |
| jboss.server.name | Description: Specify the server configuration name of jboss where application has to be deployed. This is useful if administrator wants to run multiple applications on same jboss. Note: The path must be separated by ‘/’ and not ‘\’  Default Value: default  By default it is set to 'default' configuration, the application will be installed in JBOSS\_HOME/server/default  Permissible Values: N/A |
| jboss.server.port | Description: This property specifies theport number on which JBoss server is running. It should not be specified in case the application is being deployed on an Apache Front ended server.  This property is also used for deploying Gate for caTIES. For deploying caTIES, this property needs to be specified even if the application is deployed on an Apache front ended server.  Default Value: 8080  Permissible Values: N/A |
| jboss.container.secure | Description: Should be set to yes if JBoss is deployed as HTTPS.  Default Value: N/A  Permissible Values: yes/no |
| deploy.cas.on.catissue.jboss | Description: This property specifies if the CAS server for authentication is to be deployed on the same server as caTissue application (specified by the property jboss.home). If this property is set to true, the CAS authentication server gets deployed on the same server as catissue application. If it is set to false, then the subsequent CAS related properties are required to specify the details of CAS server deployed on some other server.  Default Value: true  Permissible Values: true/false |
| cas.jboss.server.host | Description: This property specifies the host name of the server on which CAS application has been deployed. This property is required only when deploy.cas.on.catissue.jboss is set to false.  Default Value: N/A  Permissible Values: N/A |
| cas.jboss.port | Description: This property specifies the port number of the server on which CAS application is available. This property is required only when deploy.cas.on.catissue.jboss is set to false.  Default Value: N/A  Permissible Values: N/A |
| cas.jboss.container.secure | Description: This property specifies if the server on which CAS is available needs to be accessed using a secure link (an https URL).  Default Value: no  Permissible Values: yes/no |
| cas.server.keystore.path | Description: This property specifies the path to the keystore file if the server on which CAS is available needs to be accessed using a secure link (an https URL).  Default Value: N/A  Permissible Values: N/A |
| email.administrative.emailAddress | Description: Email address of the administrator. This could be an email alias if there is more than one administrator.  Default Value: N/A  Permissible Values: N/A |
| email.sendEmailFrom.emailAddress | Description: Email address used to send email messages from the deployment script.  Default Value: N/A  Permissible Values: N/A |
| email.mailServer | Description: The mail server used to send email messages.  Default Value: N/A  Permissible Values: N/A |
| use.email.commonpackage.config | Description: It should be set to true if you want to send email notification on server exceptions.  Default Value: false  Permissible Values:false,true |
| email.sendEmailTo.emailAddress | Description: Server exception’s email recipients. This could be an email alias or comma separated email addresses.  Default Value: N/A  Permissible Values: N/A |
| email.admin.support.emailAddress | Description: Email address of the administrator. This would be included in case email.sendEmailTo.emailAddress is empty or invalid  Default Value: N/A  Permissible Values: N/A |
| email.sendEmailFrom.name | Description: The sender name for server exception notification email.  Default Value: N/A  Permissible Values: N/A |
| email.exception.subject | Description: The subject line for server exception notification email.  Default Value: System Exception Occurred.  Permissible Values: N/A |
| session.timeout | Description: The web application inactivity timeout interval in minutes. Set this to a very high value (for example, 100) if you do not want your users to be timed out ever.  Default Value: 30  Permissible Values: Numeric Value |
| app.additional.info | Description: This property specifies additional information that needs to be displayed on the home page along with the application information, to identify the instance without referring to URls. This property can be used to put information like ‘SandBox Instance’, ‘Production Instance’  Default Value: N/A  Permissible Values: N/A |
| first.admin.department | Description: The first department that will be created and used in creating the first administrator user.  Default Value: N/A  Permissible Values: N/A |
| first.admin.institution | Description: The first institution that will be created and used in creating the first admin user.  Default Value: N/A  Permissible Values: N/A |
| first.admin.cancerresearchgroup | Description: The first cancer research group that will be created and used while creating the first admin user.  Default Value: N/A  Permissible Values: N/A |
| first.admin.emailAddress | Description:  Email address used in creating the first admin user. It will also be the login name for the first user.  Default Value: N/A  Permissible Values: N/A |
| first.admin.password | Description:  The password of the first admin user of the application  Default Value: N/A  Permissible Values: N/A |
| caties.mmtx.home | Description: The folder in which the MMTx library is installed on the machine on which you are going to run the caTIES concept coder service. This property is only needed when deploying the concept coder server of the caTIES-like component.  Default Value: N/A  Permissible Values: N/A |
| show.hide.forms.based.on.CPs | Description: This property is used during the execution of target 'show\_hide\_forms\_based\_on\_CPs', which is called in the deployment process. The task is executed only when the value of the property is set to true. The task is used to associate entities/categories with particular CPs. It takes an XML file 'Show\_Hide\_Forms.xml' as input. The XML has to be modified depending on the associations we want between entities/categories and CPs. The entities/categories when associated to a CP are not visible to any other CP.  Default Value: false  Permissible Values: true, false |
| Application.url | Description: The application URL is required for creation of Annotation Forms using form import feature of Dynamic Extensions. For importing the forms, it is necessary for the server to be up and running and the URL specified in this property is used for uploading the forms into the application.  Default Value: N/A  Permissible Values: Application URL on which the categories need to be created. (http://127.0.0.1:8080/catissuecore) |
| cacore.deployable.location | Description: Specify the folder where the caCORE API jars of DE model is located. The default DE model jars are available at CATISSUE\_HOME/cacore\_deployable folder  Default Value: ./cacore\_deployable  Permissible Values: N/A |
| exclude.entitygroup | Description: Comma separated Entity Group names which are to be excluded from generating the caCORE.  Default Value: None  Permissible Values: N/A |
| include.entitygroup | Description: Comma separated Entity Group names which are only to be included for generating the caCORE.  Default Value: N/A  Permissible Values: N/A |
| saved.query.owner | Description: This property specifies the login name of the user who should be made the owner of the old saved queries (pre v1.2).  This property is required to support the shared query feature (which is introduced in v1.2) for old saved queries.  Default Value: N/A  Permissible Values: N/A |
| **IDP Properties** | **Only needed if deploying caTissue with an ID Provider** |
| csm.database.type | Description: The database type used in the application.  Default Value: N/A  Permissible Values: : mysql, oracle |
| csm.database.host | Description: Thehostname or IP address of the machine on which the database server is running.  Default Value: localhost  Permissible Values: N/A |
| csm.database.port | Description: Theport number to connect with the database server.  Default Value:  Default port for MySQL: 3306  Default port for Oracle: 1521  Permissible Values: N/A |
| csm.database.name | Description: The name of the database. Specify the same name that you have specified while creating the database.  Default Value: N/A  Permissible Values: N/A |
| csm.database.username | Description: The username used to connect to the database.  Default Value: N/A  Permissible Values: N/A |
| csm.database.password | Description: The password used to authenticate the database user.  Default Value:N/A  Permissible Values: N/A |

**File: project.properties**

| **Property Name** | **Description** |
| --- | --- |
| mysql.minimum.version | Description: This property should be set to the version of MySQL installed on the deployment machine.  Default Value: REPLACE\_VALUE  Permissible Values: N/A |

**Configuring the display of forms:**

**File:** Show\_Hide\_Forms.xml

Location: CATISSUE\_HOME\modules\dynamic\_extensions\conf

Use this file to configure the display of clinical and pathology annotation forms (or any form created using dynamic extensions) that are loaded into the application during the deployment.

In caTissueSuite 1.0, many clinical and pathology forms were modeled based on the CAP checklists and loaded into caTissue using the Dynamic Extensions (DE) feature. But these data entry forms were tightly coupled with the model. The community requestedthat all the default forms be hidden so that users can create more user-friendly forms using the form upload feature of caTissue Suite 2.0. Using this feature, administrators can select attributes from the model and also specify the UI properties during form upload.

For reference, four such forms are loaded and displayed by default with caTissueSuite 2.0 for the following clinicalannotations- Chemotherapy,Radiation Therapy,Alcohol History Annotation,Smoking History.Refer toAppendix B – Clinical Annotation Forms List for more information.

**Configuring Label/Barcode Generator**

**File**: LabelGenerator.Properties

This file present at CATISSUE\_HOME/modules/caTissue/conf/catissuecore\_properties is used to configure automatic label or barcode generator for different objects of caTissue. By default it is set to default generators. Users can write customized generators and configure them here. Refer to technical guide for more details. In order to be able to enter the barcode or label manually,the values of the below parameters have to be blanked out.

| **Parameter Name** | **Details** |
| --- | --- |
| specimenLabelGeneratorClass | Label generator for Specimen |
| storageContainerLabelGeneratorClass | Label generator for Container |
| specimenBarcodeGeneratorClass | Barcode generator for Specimen |
| storageContainerBarcodeGeneratorClass | Barcode generator for Container |
| speicmenCollectionGroupLabelGeneratorClass | Label generator for Specimen Collection Group |
| speicmenCollectionGroupBarcodeGeneratorClass | Barcode generator for Specimen Collection Group |
| collectionProtocolRegistrationBarcodeGeneratorClass | Barcode generator for Collection Protocol Registration |
| protocolParticipantIdentifierLabelGeneratorClass | Unique number generator for PPI |

**Note:**These properties can be changed after deployment also. This file will be present at<JBOSS-HOME>/server/default/catissue-properties/. Once any property is changed, the server needs to be restarted.

To implement the auto label generation based on system tokens as described in the “Setting Up a Planned Collection Protocol” section of the Users Manual,the specimenLabelGeneratorClass needs to be changed as described below:

specimenLabelGeneratorClass=edu.wustl.catissuecore.namegenerator.DefaultTemplateBasedLabelGenerator

## Creating Database Scripts

The following are sample scripts, which are used to create the database and users for Oracle and MySQL and are available in CATISSUE\_HOME/db/MySql and CATISSUE\_HOME/db/Oracle. You can use your own script to create the database and users as per your requirements.

**Note:** The default script will create a user with all permissions. This will be required because this user will be used by the application to connect to the database. For features like Dynamic Extensions and Advance Query, the database is manipulated to add new tables. This is the reason for which the DB user with all permissions is created. Without creating or specifying a user, database, and table space (applicable only for Oracle), the ANT script will not be able to create the caTissue Suite schema for the application.

* **SQL Script for MySQL:** MySql\_DB\_Creation.sql

Before executing the SQL scripts, update the following parameters in MySql\_DB\_Creation.sql script.Table : MySQL Database Creation Parameters

|  |  |
| --- | --- |
| **Parameter Name** | **Description** |
| DATABASE\_NAME | Description: MySQL database name where the application data will be stored.  Default Value: N/A  Permissible Values: N/A |
| USERNAME | Description: The user name used to connect to the database.  Default Value: N/A  Permissible Values: N/A |
| PASSWORD | Description: The password used to authenticate the user name.  Default Value: N/A  Permissible Values: N/A |

If you are creating a user through your own script, in the case of MySQL, ensure that the mysql.user table has File\_priv=Y value for the user.To check this, run the query:

select user,File\_priv from mysql.user;

The user which you are using for caTissue deployment should have 'Y' for File\_priv.It is mandatory for successful database deployment.After executing the command for updating the File\_priv for the user,execute theflushprivilegescommand to reload the grant table for the user.

* **SQL Script for Oracle:**Oracle\_DB\_creation.sql

Before executing the SQL scripts update the following parameters in Oracle\_DB\_creation.sql script:

Table : Oracle Database Creation Parameters

|  |  |
| --- | --- |
| **Parameter Name** | **Description** |
| TABLESPACE\_NAME | Description: Oracle tablespace name where the application data will be stored.  Default Value: N/A  Permissible Values: N/A |
| TABLESPACE\_PATH | Description: The location or path of the data file where Oracle tablespace will be created.  Default Value: N/A  Permissible Values: N/A |
| USERNAME | Description: The user name used to connect to the database.  Default Value: N/A  Permissible Values: N/A |
| PASSWORD | Description: The password used to authenticate the user name.  Default Value: N/A  Permissible Values: N/A |

## Deploying caTissue Suite

Once the pre-installation set up is ready, you are ready to deploy the application. The following section provides different approaches to deploy the application.

**Note:**If you are using the ORACLE database, it is mandatory to install the Oracle client on the machine that is hosting the JBoss server. Ensure that the system variable ORACLE\_HOME is set properly and the system variable PATH contains ORACLE\_HOME/bin.

### New Installation

To deploy a new installation of caTissue Suite:

1. Go to the command prompt and change the directory to CATISSUE\_HOME/ folder.
2. Execute the command:

ant install

**Note:**

1. Fresh installation will install a new JBoss server at the location defined by the property application.base.path.windows/application.base.path.linux specified in install.properties. If a server is already present at that location, a user promt will be displayed confirming to proceed and overrite the JBoss server with the new one. To proceed, please enter ‘Y’ and press Enter. In case user does not want to overrite the JBoss server, please enter ‘N’ and press Enter. In this case the deployment will be terminated.
2. In case of MySQL database, the system prompts for the user input to override the database (specified by the property database.name in install.properties/upgrade.properties) and proceed. Please enter ‘Y’ and press Enter to proceed. In case user does not want to override the database, please press ‘N’ and presss Enter. In this case the deployment will be terminated.

### Upgrade

**Note**: It is strongly recommened that you perform and verify the upgrade process in a non-production environment first.

**Caution**: Always backup your database prior to performing an upgrade.

**Note**: caTissue Suite v2.0upgrade script supports only direct upgrades from caTissue Suite v1.2. If you want to upgrade from previous NCI versions of caTissue Suite, then you will need to first upgrade it to v1.2 using the appropriate version of caTissue installable, and then follow the upgrade step forcaTissue Suite v2.0

Upgrade Path 🡪

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| caTissue Core | caTissue Suite 1.0 | catissue Suite 1.1,  caTissue Suite 1.1.1,  caTissue Suite 1.1.2 | caTissue Suite 1.2 | caTissue Suite 2.0 |

To upgradeto caTissue Suite v2.0:

1. Go to the command prompt and change the directory to CATISSUE\_HOME/ folder.
2. To **upgrade** from caTissue Suite v1.2 execute the command :

ant upgrade

After the successful deployment of the application, a test mail will be sent to the administrative email address specified in the upgrade.properties file.

**Refer to** Appendix E - Verification of upgrade data**,** to verify the success of the upgrade process.

**Points to remember about Dynamic Extensions during deployment: [TBD]**

The deployment command does not deploy the caCORE API service of Dynamic Extension and Clinical Annotation model. If you wish to deploy this, run the following command:

ant deploy\_ca\_model\_war

In case of deployment errors, refer to the section Deployment Errors for details.

### Re-deploying the caTissue Application

If you need to make application changes (e.g. changes to the install.properties file or upgrade.properties file) you can safely re-deploy the application even though the database for the application has already been created. To do this the following target can be used:

1. Go to the command prompt and change the directory to CATISSUE\_HOME/ folder.
2. Execute the command:
   1. New Installation

ant install:jboss

**Note:**

This target will install a new JBoss server at the location defined by the property application.base.path.windows/application.base.path.linux specified in install.properties. If a server is already present at that location, a user promt will be displayed confirming to proceed and overrite the JBoss server with the new one. To proceed, please enter ‘Y’ and press Enter. In case user does not want to overrite the JBoss server, please enter ‘N’ and press Enter. In this case the deployment will be terminated

* 1. Upgrade

ant upgrade:jboss

**NOTE:**Before executing this command, please ensure that the pre installation steps have been followed precisely. Please refer to section 2.3.

## Performing Post Installation Configuration

The post installation configuration consists of:

1. Configuring JBoss
2. Configuring JBoss server to deploy HTTPS
3. Starting JBoss

### Configuring JBoss(Libraries, Steps to Minimize Security Risks, etc.)

Some of the JBoss-specific libraries might conflict with the jars used in the application as the version might differ.

* + **File:**hibernate3.jar

**Location:**JBOSS\_HOME/server/<jboss.server.name>/lib

**Default Location**: JBOSS\_HOME/server/default/lib

If hibernate3.jar is present, delete the file from the above location.

In order to minimize security risks for the application, we need to block the access to potentially harmful HTTP methods like “PUT, DELETE, etc.” This is done by adding the “<security-constraint>” to the web.xml at the specific location listed below.

* + **File:**web.xml

**Location:**JBOSS\_HOME/server/<jboss.server.name>/deploy/jboss-web.deployer/conf

Add the following configuration settings to the web.xml

<security-constraint>

<display-name>excluded</display-name>

<web-resource-collection>

<web-resource-name>No Access</web-resource-name>

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

<http-method>DELETE</http-method>

<http-method>PUT</http-method>

<http-method>HEAD</http-method>

<http-method>OPTIONS</http-method>

<http-method>TRACE</http-method>

</web-resource-collection>

<auth-constraint />

</security-constraint>

In order to change the logger settings, make the following changes:

* + **File:**jboss-log4j.xml

**Location:**JBOSS\_HOME/server/<jboss.server.name>/conf

Change the settings as below to avoid debug statements in the logger files to reduce the size of the log file. Add a new param tag with Threshold value as WARN

1.<appendername="FILE" class="org.jboss.logging.appender.DailyRollingFileAppender">

<errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>

<param name="File" value="${jboss.server.home.dir}/log/server.log"/>

<param name="Append" value="false"/>

<param name="Threshold" value="WARN"/>

</appender>

* + **File:** log4j.properties

**Location:**JBOSS\_HOME/server/default/catissuecore-properties/

This file contains caTissue specific logging settings. Change the values of following properties to WARN as shown below to avoid debug statements in the logger files to reduce the size of the log file.

log4j.rootCategory=WARN, CONSOLE

log4j.appender.CONSOLE.Threshold=WARN

log4j.appender.LOGFILE.Threshold=WARN

log4j.appender.HIBERNATE.Threshold=WARN

log4j.logger.org.hibernate=WARN,HIBERNATE

log4j.logger.edu.wustl.catissuecore=WARN, LOGFILE

**Note:**Log4j is an open source application for logging application messages. Please refer to <http://logging.apache.org/log4j/docs/> for more detail about loggers.

* + **File:** caTissueCore\_Properties.xml

**Location:**JBOSS\_HOME/server/<jboss.server.name>/catissuecore-properties/

This file contains the configuration properties to run caTissue. The following table describes all the properties, their default values, and their permissible values.

Table : Configuring caTissueCore\_Properties.xml

| **Parameter Name** | **Details** | | |
| --- | --- | --- | --- |
| email.administrative.emailAddress | Description: Email address of the application administrator. This could be an email alias if there is more than one administrator.  Default Value: N/A  Permissible Values: N/A | | |
| email.sendEmailFrom.emailAddress | Description: Email address used to send emails from the application.  Default Value: N/A  Permissible Values: N/A | | |
| email.mailServer | Description: The mail server used to send emails.  Default Value: N/A  Permissible Values: N/A | | |
| Institution.name | Description: The Name of the institution deploying the application.  Default Value: N/A  Permissible Values: N/A | | |
| use.proxy.server | Description: A Boolean parameter use to check whether the host is behind the proxy server or firewall. This is used while downloading the CDEs from caDSR.  Default Value: N/A  Permissible Values: true, false | | |
| proxy.host | Description: The proxy server address for your host.  Default Value: N/A  Permissible Values: N/A | | |
| proxy.port | Description: The proxy server port for your host.  Default Value: N/A  Permissible Values: N/A | | |
| proxy.username | Description: The username required to connect to the proxy server.  Default Value: N/A  Permissible Values: N/A | | |
| proxy.password | Description: The password required for proxy server authentication.  Default Value: N/A  Permissible Values: N/A | | |
| barcode.isEditable | Description: Barcode field on all UI pages can be made non editable by using this parameter  Default Value: True(Barcode will be editable)  Permissible Values: True, False | | |
| session.time.out | Description: Session timeout  Default Value: N/A  Permissible Values: N/A | | |
| app.additional.info | Description: This property specifies additional information that needs to be displayed on the home page along with the application information, to identify the instance without referring to URls. This property can be used to put information like ‘SandBox Instance’, ‘Production Instance’  Default Value: N/A  Permissible Values: N/A | | |
| **Password Security Settings** | | | |
| Password.not\_same\_as\_ last\_n | Description: If set to a number n, the system will not allow the users to set a password that is the same as one of their previous n passwords.  Default Value: N/A  Permissible Values: N/A | | |
| MinimumPasswordLength | Description: The minimum length for password.  Default Value: N/A  Permissible Values: N/A | | |
| DaysCount | Description: The number of days within which the password cannot be changed.  Default Value: N/A  Permissible Values: N/A | | |
| password.expire\_after\_n\_days | Description: The number of days after which the password will expire.  Default Value: N/A  Permissible Values: N/A | | |
| **Default CDE/Enumerated value setting**  **Note:** The default value should be one from the permissible CDE list. If an invalid value is provided, the respective dropdown on the web application will be shown without any value selected. | | | |
| defaultTissueSite | Description: The default value for the Tissue Site.  Default Value: Not Specified  Permissible Values: N/A | | |
| defaultClinicalStatus | Description: The default value for the Clinical Status.  Default Value: Not Specified  Permissible Values: N/A | | |
| defaultGender | Description: The default value for the Gender.  Default Value: Unspecified  Permissible Values: N/A | | |
| defaultGenotype | Description: The default value for the Genotype.  Default Value: Unknown  Permissible Values: N/A | | |
| defaultSpecimen | Description: The default value for Specimen Class.  Default Value: N/A  Permissible Values: N/A | | |
| defaultTissueSide | Description: The default value for Tissue Side.  Default Value: Not Specified  Permissible Values: N/A | | |
| defaultPathologicalStatus | Description: The default value for Pathological Status.  Default Value: Not Specified  Permissible Values: N/A | | |
| defaultReceivedQuality | Description: The default value for the Received Quality.  Default Value: Not Specified  Permissible Values: N/A | | |
| defaultFixationType | Description: The default value for Fixation Type.  Default Value: Not Specified  Permissible Values: N/A | | |
| defaultCollectionProcedure | Description: The default value for the Collection Procedure.  Default Value: Not Specified  Permissible Values: N/A | | |
| defaultContainer | Description: Thedefault value for the Container.  Default Value: Not Specified  Permissible Values: N/A | | |
| defaultMethod | Description: The default value for the Method.  Default Value: Not Specified  Permissible Values: N/A | | |
| defaultEmbeddingMedium | Description: The default value for the Embedding Medium.  Default Value: Not Specified  Permissible Values: N/A | | |
| defaultBiohazard | Description: The default value for Biohazard.  Default Value: N/A  Permissible Values: N/A | | |
| defaultSiteType | Description: The default value for the Site Type.  Default Value: N/A  Permissible Values: N/A | | |
| defaultSpecimenType | Description: The default value for the Specimen Type.  Default Value: N/A  Permissible Values: N/A | | |
| defaultEthnicity | Description: The default value for Ethnicity.  Default Value: Unknown  Permissible Values: N/A | | |
| defaultRace | Description: The default value for Race.  Default Value: Unknown  Permissible Values: N/A | | |
| defaultClinicalDiagnosis | Description: The default value for Clinical Diagnosis.  Default Value: Not Specified  Permissible Values: N/A | | |
| defaultStates | Description: The default value for States.  Default Value: N/A  Permissible Values: N/A | | |
| defaultCountry | Description: The default value for the Country.  Default Value: United States  Permissible Values: N/A | | |
| defaultHistologicalQuality | Description: The default value for Histological Quality.  Default Value: Not Specified  Permissible Values: N/A | | |
| defaultVitalStatus | Description: The default value for Vital Status.  Default Value: Unknown  Permissible Values: N/A | | |
| **Participants matching parameters**  Change these parameters to alter the way the matching participants are found during participants’ insertion.For more information on the algorithm, refer to[ParticipantMatchingAlgorithm](http://gforge.nci.nih.gov/plugins/scmcvs/cvsweb.php/catissuecore/Version1.1/WashU/TechnicalGuide/MPI%20Matching%20Algorithm.doc?cvsroot=cacorecvs). | | | |
| SSNExact | | Description: Exact match for SSN field  Default Value: 30 | |
| SSNPartial | | Description: Partial match for SSN field  Default Value: 20 | |
| DOBExact | | Description: Exact match for date of birth field  Default Value: 20 | |
| DOBPartial | | Description: Partial match for date of birth field  Default Value: 15 | |
| LastNameExact | | Description: Exact match for last name field  Default Value: 20 | |
| LastNamePartial | | Description: Partial match for last name field  Default Value: 15 | |
| FirstNameExact | | Description: Exact match for first name field  Default Value: 10 | |
| FirstNamePartial | | Description: Partial match for first name field  Default Value: 5 | |
| MiddleNameExact | | Description: Exact match for middle name field  Default Value: 10 | |
| MiddleNamePartial | | Description: Partial match for middle name field  Default Value: 5 | |
| GenderExact | | Description: Exact match for gender field  Default Value: 5 | |
| RaceExact | | Description: Exact match for race field  Default Value: 10 | |
| RacePartial | | Description: Partial match for race field  Default Value: 5 | |
| Bonus | | Description: Given for LastName, FirstName and DOB exact match  Default Value: 15 | |
| **Help links –** By default the help documents are present within the application. If there are updated versions which users would like to point to, then it can be configured by using below parameters. | | | |
| userguide.link | | | Description: User manual link  Default Value:caTissue\_Suite\_User\_Manual.doc |
| technicalguide.link | | | Description: Technical guide link  Default Value:caTissueSuite\_v1\_2\_\_Technical\_Guide.doc |
| trainingguide.link | | | Description: Training materials link  Default Value:<http://cabigtrainingdocs.nci.nih.gov/caTissue/index.html> |
| umlmodel.link | | | Description: UML model diagrams link  Default Value:caTissueSuiteV11\_UML.zip |
| knowledgecenter.link | | | Description: KnowledgeCenter Link  Default Value:  <https://cabig-kc.nci.nih.gov/Biospecimen/KC/index.php/Main_Page> |
| knowledgecenterforum.link | | | Description: KnowledgeCenter Forum Link  Default Value:<https://cabig-kc.nci.nih.gov/Biospecimen/forums/> |
| **KeywordSearch -**Change this parameter to make KeywordSearchenable or disable Indexing of attributes at run time. | | | |
| KeywordSearchConfigured | | | Description: Determined is KeywordSearch Configured  Default Value:false |
| **caCORE properties –** Set the value for caCORE query execution. | | | |
| maxRecordsPercaCOREQueryAllowed | | | Description: Maximum records to be fetched during caCORE/caGrid query execution  Default Value: 10000 |

### Configuring caTissue Suite for KeywordSearch

A Lucene-based search tool, Keyword Search, is introduced in caTissue v2.0. Keyword Search allows you to search by typing keywords and displays data that matches the search strings within a range of attributes. For more information on searching with Keyword Search, refer to the *caTissueEndUser Manual*.

To use the search functionality, index the attributes you want to search. Once the application is deployed,create the attributes indexes on the Keyword Search.

For creating the index, change the directory to CATISSUE\_HOME/ folder and execute the following command:

ant runKeywordSearchIndexer

**Note:** Ensure that the DB properties are set appropriately in install.properties or upgrade.properties.

To enable run time Keyword Search indexing set value of KeywordSearchConfigured property to ‘true’ in caTissueCore\_Properties.xml. This file is available at JBOSS\_HOME/server/<jboss.server.name>/catissuecore-propertiesfolder and restart the JBoss server.

### Configuring JBoss Server to deploy caTissue using HTTPS

**IMPORTANT:**  Since caTissue contains patient identified information, it is important to deploy it in a secure environment. For more information see <http://en.wikipedia.org/wiki/Https>. Even if the application is installed within secure firewall, it is highly recommended to deploy the application using HTTPS to avoid any unauthorized access.**Please note that caTissue Suite v.2.0 deployment is not supported on HTTPS unsigned or self signed certificates.**

To deploy caTissue as a HTTPS-based secure web application, you need to perform the following steps:

1. Obtain a server certificate from a certificate authority.
2. Move the certificate to the appropriate JBoss directory.
3. Edit the JBoss configuration file to turn on SSL.

**CAUTION:**  The JBoss server should be shut down during this process.

**Step 1: Moving the Keystore File**

1. Rename the server certificate file to chap8.keystore.
2. Copy this keystore file to the conf/ directory of your JBoss installation. For example:

JBOSS\_HOME/server/<jboss.server.name>/conf.

**Step 2:** Editing the JBoss Configuration File

1. Open file JBOSS\_HOME/server/<jboss.server.name>/deploy/jboss-web.deployer/server.xml.
2. Inside this file, there should be a tag as in the following:

<!--

<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"

maxThreads="150" scheme="https" secure="true"

clientAuth="false" sslProtocol="TLS" />

-->

1. In the tag from step 2 (above), perform the following changes:
   1. Uncomment the block if it is commented.
   2. Replace the content with the tag below

<Connector port="443" address="${jboss.bind.address}" SSLEnabled="true" maxThreads="100" strategy="ms" maxHttpHeaderSize="8192" emptySessionPath="true" scheme="https" secure="true" clientAuth="false" keystoreFile="${jboss.server.home.dir}/conf/chap8.keystore" keystorePass="rmi+ssl" sslProtocol = "TLS" />

* 1. Change the port to 443 or to any desired port on which you want to run the application on.
  2. Change the keystore password in parameter keystorePassto the password specified while creating the certificate.

1. After the changes, the block would look like this:

<!-- SSL/TLS Connector configuration using the admin devl guide keystore -->

<Connector port="443" address="${jboss.bind.address}" SSLEnabled="true" maxThreads="100" strategy="ms" maxHttpHeaderSize="8192" emptySessionPath="true" scheme="https" secure="true" clientAuth="false" keystoreFile="${jboss.server.home.dir}/conf/chap8.keystore" keystorePass="changeit" sslProtocol = "TLS" />

1. Comment the section where the HTTP connector tag is defined so that the application is not accessible through HTTP.
2. Start the JBoss server and access the application by using the following URL: <https://machine_name:port/catissuecore/>

## Configuring CSM

**CAUTION:** Skip this section if you have used the install/upgrade or the install:jboss/upgrade:jboss target to deploy the application.

**File:**properties-service.xml

**Location:**JBOSS\_HOME/server/<jboss.server.name>/deploy/

Add the following entries under the <mbean> tag for SystemProperties in the above file.

<attribute name="Properties"><!-- could already exist -->

    gov.nih.nci.sdk.remote.catissuecore.securityLevel=0

    gov.nih.nci.sdk.applications.session.timeout=3000

    gov.nih.nci.security.configFile=APP\_SEQURITY\_CONFIG\_PATH

    app.propertiesFile=APP\_PROPERTIES\_PATH

</attribute><!-- could already exist -->

Table : Configuring properties service.xml

|  |  |
| --- | --- |
| Parameter Name | Details |
| APP\_SEQURITY\_CONFIG\_PATH | Description: Name of the property specifying the application security.  Value:JBOSS\_HOME/server/<jboss.server.name>/catissuecore-properties/ApplicationSecurityConfig.xml |
| APP\_PROPERTIES\_PATH | Description: Name of the property specifying the application configuration.  Value:JBOSS\_HOME/server/<jboss.server.name>/catissuecore-properties/caTissueCore\_Properties.xml |

**Note:**

* gov.nih.nci.security.configFileis the name of the property file, which points to the fully qualified path where ApplicationSecurityConfig.xmlhas been placed. The name of the property should be gov.nih.nci.security.configFile, and should not be modified as it is a system-wide property.
* Please note that the path must be separated by UNIX style path separator "/".

**File:**login-config.xml

**Location:** JBOSS\_HOME/server/<jboss.server.name>/conf/

Add the following entries to the file above.

<application-policy name=**APPLICATION\_POLICY**>

  <authentication>

    <login-module code = "gov.nih.nci.security.authentication.loginmodules.RDBMSLoginModule" flag = "required" >

        <module-option name="driver">DRIVER\_CLASS\_NAME</module-option>

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

        <module-option name="user">USER\_NAME</module-option>

        <module-option name="passwd">PASSWORD</module-option>

        <module-option name="query">select \* from csm\_user where login\_name=? and password=? </module-option>

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

      </login-module>

    </authentication>

</application-policy>

The login-module is a CSM Login module class used to perform the authentication task. In this case it is

gov.nih.nci.security.authentication.loginmodules.RDBMSLoginModule.

Update the following parameters in the above setting:

Table : Configuring login-config.xml

| **Parameter Name** | **Details** |
| --- | --- |
| APPLICATION\_POLICY | Description: Specifies the application for which you are defining the authentication policy.  Default Value: catissuecore  Permissible Values: N/A |
| DRIVER\_CLASS\_NAME | Description: The database driver loaded in memory to perform database operations.  Default Value: N/A  Permissible Values:  MySQL - org.gjt.mm.MySQL.Driver  Oracle –oracle.jdbc.driver.OracleDriver |
| URL | Description: The URL used to locate and connect to the database. Specify the database URL that you used during the creation of the database.   * + **MySQL:**   jdbc:MySQL://database.host:database.port/database.name   * + **Oracle:**   jdbc:oracle:thin:@database.host:database.port:database.name  Default Value: N/A  Permissible Values: N/A |
| USER\_NAME | Description: The username used to connect to the database.  Default Value: catissue\_core  Permissible Values: N/A |
| PASSWORD | Description: The password used to authenticate the username.  Default Value: catissue\_core  Permissible Values: N/A |

## Starting and shuttingdown the Application Server

**Note:** It is important that you review your Jboss configuration against your institutional security policies prior to using caTissue for real work - particularly where protected health information or other sensitive data will managed in caTissue. See

<https://wiki.nci.nih.gov/display/BuildandDeploymentAutomation/Securing+JBOSS+Console+Apps>for details on how to secure the JBoss Console. For more information see the caBIG® Enterprise Security Program web page

<https://wiki.nci.nih.gov/pages/viewpage.action?pageId=24276546>

Before starting the server, make the following changes in the JBoss startup file (run.conf.bat on Windows, run.conf on Unix/Linux).

**For Windows:** Open the fileJBOSS\_HOME/bin/run.conf.bat and search for the line set JAVA\_OPTS and modify it to appear like:

set "JAVA\_OPTS=-Xms128M -Xmx1024M -XX:MaxPermSize=256M -Dorg.apache.jasper.compiler.Parser.STRICT\_QUOTE\_ESCAPING=false"

**For Linux and Unix:** Open file JBOSS\_HOME/bin/run.conf and search for the line starting with JAVA\_OPTS and modify it to appear like: “

JAVA\_OPTS="-Xms128M -Xmx1024M -XX:MaxPermSize=256M -Dorg.apache.jasper.compiler.Parser.STRICT\_QUOTE\_ESCAPING=false Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000"

Now, start the JBoss server as per the instructions below:

**Note:** On successful execution of the the targets install/install:jboss, the server starts automatically. User needs to kill the server and then restart it as specified below.

**CAUTION:** Line for setting parameters for JAVA\_OPTS ideally will already be present in the run.bat/run.conf, in which case you just need to change the heap space and PermSize settings similar to above.

Table : Server Startup and Shutdown

| **Operation** | **Windows** | **Linux and Unix** |
| --- | --- | --- |
| **Start** | JBOSS\_HOME/bin/run.bat -b<host or ip> | nohup $JBOSS\_HOME/bin/run.sh-b <host or ip> |
| **Stop** | JBOSS\_HOME/bin/shutdown.bat | JBOSS\_HOME/bin/shutdown.sh |

**Note:**The –b option inserver start up command is optional. The host name or IP address specified with this option is used as a bind address to all jboss services. If the –b option is not specified, the server will be accessible only from the machine where it is deployed i.e. the server is accessible by URL http://localhost:port\_number

## Accessing the Web Application

Once the JBoss server is started, you can access the application using a web browser. The following is the URL pattern used for the application: <http://machine_name:port_number/catissuecore>

When the browser displays the home page of caTissue Suite, you can log on to the application by specifying the values which were configured in first.admin.emailAddress andfirst.admin.passwordof caTissueInstall.properties before deployment.

## Deployment Errors

The common deployment errors are:

**Error:** ANT\_HOME is set incorrectly or ant could not be located. Please set ANT\_HOME.

**Cause:** ANT\_HOME is not set or set incorrectly.

**Solution:** Set the ANT\_HOMEenvironment variable and retry the deployment.

**Error:** ant is not recognized as an internal or external command, operable program, or batch file.

**Cause:** ANT\_HOME/bin is not in the system path.

**Solution:** Make sure that ANT\_HOME is set to the appropriate folder and ANT\_HOME/bin is present in the system path variable.

**Error:** The following error occurred while executing this line: D:\SuiteDep\deploy.xml:501: java.sql.SQLException: Unable to connect to any hosts due to exception: java.net.ConnectException: Connection refused: connect

**Cause:** Database name, user name, or password is not configured correctly.

**Solution:** Crosscheck the name by logging to the database and by using the database client.

**Error:** Cannot accesshttp://machine\_name:port\_number/catissuecore

**Cause:** Check if thejboss.home parameter in install.properties file is set correctly.

**Solution:** Correct the jboss.home parameter if set incorrectly. You should use the Unix path separator "/"to separate folders in the path in place of the Windows forward slash. For example: use D:/jboss and not D:\jboss.

**Error:** Deployment log shows email could not be sent.

**Cause:** The configuration parameters in the install.properties file are not set correctly. The email server is not accessible from this machine.

**Solution:** There is no need to run the deployment again if this is the only error you encountered. However, if you do not correct this problem, caTissue will not send email notifications. To correct this problem, correct the configuration parameters in JBOSS\_HOME/server/<jboss.server.name>/catissuecore-properties/caTissueCore\_propeties.xml

**Error:** java.io.IOException: java.io.IOException: sqlldr: not found

**Cause: S**ystem variables ORACLE\_HOME and PATH not set properly.

**Solution**: Set the system variable to ORACLE\_HOME and ensure that the system variable ‘PATH’ points to ORACLE\_HOME/bin.

**Error:**java.sql.SQLException: Invalid authorization specification message from server: "Access denied for user 'username'@'%' (using password: YES)"

**Cause:**No file privileges were given to the database user. Occurs while importing CA model data of caTissue Suite Application to MySQL Database.

**Solution:**Ensure that the mysql.user table hasFile\_priv= ‘Y’ value for the user.To check whether it’s set to ‘Y’, execute the following query.

SELECT User, file\_priv from mysql.user;

If its ‘N’, you need to set the file privilege to ‘Y’ using following sql.

USE mysql;

UPDATE User SET File\_priv = 'Y' where User='username';

FLUSH PRIVILEGES;

After setting the appropriate file privileges, redeploy the application for creation of the database.

**Known issues:**

Bug 9885: DE warnings at the time of deployment – These are just warnings and do not affect the functionality, so can be ignored.

Bug 9696 - warnings at the time of jboss server starting - These are just warnings and do not affect the functionality, so can be ignored. This is specific to deployment with Oracle as the driver.

5938 -Searching on participant last name as scientist returns the result – Workaround for this to keep track of the audit trail of the API queries. This can be achieved by DB queries on table catissue\_audit\_event\_query\_log(contains queries executed by the user)and catissue\_audit\_event(value of comment field in this table is ‘APIQueryLog’ to indicate that it is API query).

# caTIES(v2.3)Integration in Suite

The following section describes how to load the surgical pathology reports and associate them to the Participants and Specimen collection groups in the Suite database. The caTIES pipeline consists of three services as described below:

Table : Three Services of caTIES

|  |  |
| --- | --- |
| Report loader | This service loads the HL7 formatted reports into the caTissue Suite database. This service takes care of creating the corresponding participants and specimen collection groups which are not present. It also reports conflicts, if any. Please refer to the user manual for details. |
| Report de-identifier | This service de-identifies the SPRs. Users have the option to select one of the three de-identifier options described in Table 13. |
| Concept coder | This service concept-codes the de-identified SPRs with the concepts from the NCI Metathesaurus. |

Table : De-identifier options

|  |  |
| --- | --- |
| DeID de-identifier | DeID de-identifier uses the DeID de-identification tool to de-identfy SPRs. The DeID tool is available only for the Windows platform. The DeID tool must be installed on the machine prior to deploying DeID de-identifier. |
| Harvard Scrubber  de-identifier | Users may choose this option to use Harvard Scrubber as the de-identification tool. This is an open source tool andis bundled with the de-identifier hence not required to install prior to deploying the de-identifier service. |
| Do-Nothing  de-identifier | By using this option, users might choose not to de-identify the reports. |
| Custom de-identifier | This option is available for users who wish to use their own de-identifier to de-identify SPRs. |

The following deployment options are available when deploying the caTIES services:

Table : Deployment Options for caTIES Services

|  |  |
| --- | --- |
| Install Suite with zero caTIES services | This means that the caTissue Suite will not support the caTIES pipeline. |
| Install Suite with only report loader | This means that caTissue Suite will only support identified report. The scientist will not have access to the de-identified SPRs. |
| Install Suite with only report loader and de-identifier services | This means that caTissue Suite will support both identified and de-identified reports and the scientist will have access to the de-identified SPRs. But, the reports will not be concept-coded and therefore users can perform only free-text based queries. |
| Install Suite with report loader,  de-identifier, and conceptcoder services ( all three services) | This means that the caTissue Suite will support both identified and de-identified reports and the reports will be concept-coded. Therefore users can perform free-text as well as concept code based queries. |

To run the caTIES pipeline, you need to:

1. Download MMTx library and data.zip.
2. Deploy the Gate library on the caTissue Suite web server.
3. Configure and deploy caTIES.
4. Run caTIES pipeline.

## Downloading MMTx Library and data.zip

**Note:** You can ignore this section if you are not going to deploy the caTIES concept coder pipeline.

1. Download the MMTx library (MMTx-2.4b.zip) and ncimeta-0601.zip from <http://catissuecore.wustl.edu/caties_datafiles/> on the machine where you will deploy the caTIES concept coder service.
2. Unzip the MMTx-2.4b.zip to a folder. This folder will be referred to as MMTX\_ROOT henceforth.
3. Unzip ncimeta-0601.zipand extract to the MMTX\_ROOT\nls\mmtx\data\ folder.
4. Update themmtxRegistry.cfg file present at MMTX\_ROOT\nls\mmtx\config.
5. Set various properties for MMTx including MMTX\_ROOT

## Deploying Gate Library

1. On the machine where the caTissue Suite web server is deployed, change the directory to CATISSUE\_HOME/.
2. Configure the following parameters in CATISSUE\_HOME/install.properties or CATISSUE\_HOME/upgrade.properties.
   1. caties.mmtx.home
   2. jboss.server.host
   3. jboss.container.secure
3. Run ant deploy:gate

## Configuring caTIES

Configure the following parameters in the CATISSUE\_HOME/modules/caTIES/conf/deploycaties.properties file**.**

Table : caTIES Installation configuration parameters

| **Property Name** | **Description** |
| --- | --- |
| keystore.file.path | Description: Path to the keystore file if the caTissue is deployed as HTTPS.  Default Value: N/A  Permissible Values: N/A |
| **Report Loader Server Settings** | |
| add.default.collection.protocol | Description: Should be set to yesif if you want to add a default collection protocol to the system. The default collection protocol will be used by report loader to the associated reports.  Default Value: None  Permissible Values: yes, no |
| install.report.loader.server | Description: Should be set to yes if you want to install a report loader server.  Default Value: None  Permissible Values: yes, no |
| report.loader.installation.dir | Description: If you opt to install a report loader server, then this property should be used to specify the location where report loader server will be deployed.  Default Value: N/A  Permissible Values: N/A |
| input.files.dir | Description: Path of the directory from which the report loader will pick up the files containing HL7 format report for providing input to the caTIES pipeline.  Default Value: N/A  Permissible Values: N/A |
| bad.files.dir | Description: Path of the directory where the files containing invalid HL7 format reports will be placed.  Default Value: N/A  Permissible Values: N/A |
| process.file.dir | Description: Path of the directory where successfully-processed HL7 format files will be placed.  Default Value: N/A  Permissible Values: N/A |
| collection.protocol.title | Description: The title of the collection protocol which, will be used by the report loader to associated reports. If you have set add.default.collection.protocol to yes,then the deployment process will create a new collection protocol by this name. If it is set to no, then the deployment process will assume that a collection protocol by this name already exists in the database.  Default Value: N/A  Permissible Values: N/A |
| site.name | Description: The name of the site which will be used by the report loader to associate with the default collection protocol. This property is mandatory If you have set add.default.collection.protocol to yes. A new site is created with the specified name if one does not already exist.  Default Value: N/A  Permissible Values: N/A |
| **Report Concept Code Server Settings** | |
| install.concept.code.server | Description: Should be set to yes if you want to install report concept code server.  Default Value: None  Permissible Values: yes, no |
| concept.code.installation.dir | Description: If you opt to install a report concept-code server then this should be used to specify the location where the report concept-code server will be deployed.  Default Value: N/A  Permissible Values: N/A |
| **Report De-identification Server Settings** | |
| install.deidentifier.server | Description: Should be set to yes if you want to install a report de-identification server.  Default Value: None  Permissible Values: yes, no |
| deididentifier.installation.dir | Description: If you opt to install report a de-identification server, then this property should be used to specify the location where the report de-identification server will be deployed.  Default Value: N/A  Permissible Values: N/A |
| install.deid.deididentifier | Description: Should be set to yes if you want to install report de-identification server based on DeID tool. DeID is a third-party report de-identification tool which should be installed on the system to de-identify reports using this tool.  Default Value: None  Permissible Values: yes, no |
| install.harvardscrubber.deididentifier | Description: Should be set to yes if you want to install report de-identification server based on the HarvardScrubber tool. Harvard Scrubber is an open source tool for de-identification.  Default Value: None  Permissible Values: yes, no |
| install.donothing.deididentifier | Description: Should be set to yes if you do not want to de-identify report text but want to use concept coder server. Donothing identifier copies identified report text as it is to de-identified report.  Default Value: None  Permissible Values: yes, no |
| install.custom.deididentifier | Description: Should be set to yes if you wish to de-identify report text using your own custom de-identification tool.  Default Value: None  Permissible Values: yes, no |
| deid.home | Description: This property should be set only if install.deid.deididentifier property is set to yes. This property should point the DeID tool installation directory.  Default Value: None  Permissible Values: N/A |
| deid.dny.folder | Description: This property should be set only if install.deid.deididentifier property is set to yes. This property should point the folder where the DeID tool’s data dictionary is available.  Default Value: None  Permissible Values: N/A |
| deididentifier.class.name | Description: This property should be set only if install.custom.deididentifier property is set to yes. This property specifies the absolute Class name of the custom de-identifier.  Default Value: None  Permissible Values: N/A |

Configuring CATISSUE\_HOME/modules/caTIES/sites\_configuration.xml

**Note:** The properties with characters beginning and ending with @@ will be automatically replaced with the corresponding values from the CATISSUE\_HOME/install.properties or CATISSUE\_HOME/upgrade.properties file. If you are deploying caTIES services on the same machine as caTissue web application, then you need not set these parameters. If you are deploying caTIES on a different machine, then you can copy the fully configured CATISSUE\_HOME/install.properties or CATISSUE\_HOME/upgrade.properties file from the machine on which you have deployed the caTissue web application. You can also set these parameters directly in this file.

The HL7 reports contain a site abbreviation code, which may or may not map directly to the actual site name used in Suite. Therefore, for every site expected in the caTIES reports, the sites configuration file should contain:

<sites>

<site>

<SITE\_NAME><site name in caTissue></SITE\_NAME>

<SITE\_ABBRIVIATION><site name in the HL7 reports></SITE\_ABBRIVIATION>

</site>

</sites>

In this configuration file, multiple caTissue sites can be mapped to single HL7 site abbreviation. The caTissue site names mentioned here should be names of valid and active sites in caTissue. Following is an example configuration.

<sites>

<site>

<SITE\_NAME>BJH</SITE\_NAME>

<SITE\_NAME>Siteman Cancer Site</SITE\_NAME>

<SITE\_NAME>BJC</SITE\_NAME>

<SITE\_ABBRIVIATION>RY</SITE\_ABBRIVIATION>

</site>

</sites>

Here, BJH, Siteman Cancer Site, BJC are existing sites in caTissue application and the abbreviation RY is the name of the site used in HL7 forms.

Configuring CATISSUE\_HOME/modules/caTIES/conf/application-config-client-info.xml.xml

The property serviceUrl in bean id catissue\_cacoreApplicationService needs to be modified to point to the caTissue application URL. For instance, if the URL of caTissue is <https://ps2222:8443/catissuecore> then the serviceUrl value will be

<https://ps2222:8443/catissuecore>/http/applicationService

The property serviceUrl in bean id catissue\_cacoreAuthenticationManager needs to be modified to point to the caTissue application URL. For instance, if the URL of caTissue is <https://ps2222:8443/catissuecore> then the serviceUrl value will be

<https://ps2222:8443/catissuecore>/http/remoteAuthenticationManager

## Deploying caTIES

To deploy caTIES:

1. From the command prompt, go to the CATISSUE\_HOME/ folder.
2. Run the ANT task as follows: ant deploy:caties.

## Running caTIES pipeline

After deployment of the caTies server, the user has to verify each of the caTIES.properties file present within respective server’s caTies\_conf folder for:

1. Running Report Loader server
2. Running De-identification server
3. Running Concept code server.

Note: The JBoss server on which caTissue Suite is deployed should be up and running for starting the above-mentioned servers.

The table below describes the properties present within<caTIES\_server>/caTIES\_conf/caTIES.properties file which needs to be set appropriately before running the respective servers.

| **Property Name** | **Description** |
| --- | --- |
| keystoreFilePath | Description: Path to the keystore file if the caTissue is deployed as HTTPS.  Default Value: N/A  Permissible Values: N/A |
| userName | Description: Username of the user having administrator privileges to run the caTIES server. The token will be automatically replaced with the value specified in caTissueInstall.properties for the property first.admin.emailAddress.  Default Value: @@ADMIN\_EMAIL@@  Permissible Values: None |
| password | Description: Password of the user whose user name is specified in property username. The token will be automatically replaced with the value specified in caTissueInstall.properties file for the property first.admin.password.  Default Value: @@ADMIN\_PASSWORD@@  Permissible Values: None  **Note:** If you are a new user of caTissue web application, you must change the password at first login and update that new password here |
| **Report Loader Server Settings** | |
| inputDir | Description: Path of the directory from which the report loader will pick up the files containing HL7 format report for providing input to the caTIES pipeline. The token will be automatically replaced with the value specified in deploycaties.properties for the property input.file.dir.  Default Value: @@INPUT\_FILES\_DIR@@  Permissible Values: N/A |
| badFilesDir | Description: Path of the directory where the files containing invalid HL7 format report will be placed. The token will be automatically replaced with the value specified in deploycaties.properties for the property bad.file.dir.  Default Value: @@BAD\_FILES\_DIR@@  Permissible Values: N/A |
| processFileDir | Description: Path of the directory where successfully processed HL7 format files will be placed. The token will be automatically replaced with the value specified in deploycaties.properties for the property process.file.dir.  Default Value: @@PROCESS\_FILES\_DIR@@  Permissible Values: N/A |
| filePollerSleepTime | Description: Sleep time in millisecond of report loader thread and report queue processor thread.  Default Value: 100000  Permissible Values: any positive value <= Long.MAX |
| siteInfoFileName | Description: Path of the file containing site configuration information.  Default Value: ./caTIES\_conf/sites\_configuration.xml  Permissible Values: N/A |
| sectionHeaderPriorityFileName | Description: Path of the file containing report section header priority information.  Default Value: ./caTIES\_conf/SectionHeaderConfig.txt  Permissible Values: N/A |
| filePollerPort | Description: Port number where the report loader server is running.  Default Value: 3030  Permissible Values: N/A |
| collectionProtocolTitle | Description: The title of the collection protocol which, will be used by the report loader to associated reports. The token will be automatically replaced with the value specified in deploycaties.properties for the property collection.protocol.title.  Default Value: @@COLL\_PROT\_TITLE@@  Permissible Values: N/A |
| **Report Concept Code Server Settings** | |
| caties.coder.version | Description: Coder version name used by the concept code server.  Default Value: UMLS2004  Permissible Values: N/A |
| caties.gate.home | Description: Path to the gate home directory. The token will be automatically replaced on deployment.  Default Value: @@GATE\_HOME@@  Permissible Values: N/A |
| caties.creole.url.name | Description: Path to the creole.xml, which specifies values of properties required by gate for concept coding process.  Default Value: http://@@HOST@@:@@PORT@@/gate/gate\_3\_1/application/plugins/caTIES  Permissible Values: N/A |
| caties.case.insensitive.gazetteer.url.name | Description: Path to the case insensitive gazetteer definition file.  Default Value: http://@@HOST@@:@@PORT@@/gate/gate\_3\_1/application/plugins/caTIES/CaTIES\_CaseInSensitiveGazetteer.def  Permissible Values: N/A |
| caties.case.sensitive.gazetteer.url.name | Description: Path to the case sensitive gazetteer definition file.  Default Value: http://@@HOST@@:@@PORT@@/gate/gate\_3\_1/application/plugins/caTIES/CaTIES\_CaseSensitiveGazetteer.def  Permissible Values: N/A |
| caties.section.chunker.url.name | Description: Path to the section chunker.  Default Value: http://@@HOST@@:@@PORT@@/gate/gate\_3\_1/application/plugins/caTIES/CaTIES\_Chunker.jape  Permissible Values: N/A |
| caties.concept.filter.url.name | Description: Path to the concept filter.  Default Value: http://@@HOST@@:@@PORT@@/gate/gate\_3\_1/application/plugins/caTIES/CaTIES\_ConceptFilter.jape  Permissible Values: N/A |
| caties.neg.ex.url.name | Description: Path to the negative expression processor.  Default Value: http://@@HOST@@:@@PORT@@/gate/gate\_3\_1/application/plugins/caTIES/CaTIES\_NegEx.jape  Permissible Values: N/A |
| caties.concept.categorizer.url.name | Description: Path to the concept categorizer.  Default Value: http://@@HOST@@:@@PORT@@/gate/gate\_3\_1/application/plugins/caTIES/CaTIES\_ConceptCategorizer.jape  Permissible Values: N/A |
| conceptCoderSleepTime | Description: Sleep time in millisecond of report concept code pipe line manager thread.  Default Value: 86400000  Permissible Values: any positive value <= Long.MAX |
| conceptCoderPort | Description: Port number where report concept code server is running.  Default Value: 3050  Permissible Values: N/A |
| save.binary.content | Description: Should be set to true if you want to save binary content.  Default Value: false  Permissible Values: N/A  **Note**: When the concept-coder concept codes, it creates a binary-equivalent document tothe report text. The binary output contains the information related to concepts appearing in the report text. Since the size of the binary output is very large, you can discard it using this property. |
| save.xml.content | Description: Should be set to true if you want to save XML content.  Default Value: false  Permissible Values: N/A  **Note**:When the concept-coder concept codes the report text, it creates an XML-equivalent document to the report text. The XML output contains the information related to concepts appearing in the report text. Since the size of the XML output is very large, you can discard it using this property. |
| **Report De-identification Server Settings** | |
| deidentfierClassName | Description: Since there are different options for the de-identification tool,a class name can be provided to the de-identification server to select a specific tool to properly de-identify the report text. This tool is specified in the form of a class name (a software component). The token is automatically replaced by respective class name as per the de-identifier selected. The token will be automatically replaced with the value specified in deploycaties.properties for the property deidentifier.class.name if the property install.custom.deidentifier is set to yes.  Default Value: @@DEIDENTIFIER\_CLASS\_NAME@@  Permissible Values: N/A |
| maxThreadPoolSize | Description: Max pool size, which means the maximum number of threads in the thread pool.  Default Value: 20  Permissible Values: N/A |
| deididentifierPort | Description: Port number where report de-identification server is running.  Default Value: 3040  Permissible Values: N/A |
| deididentifierSleepTime | Description: Sleep time in millisecond of report de-identification pipe line manager thread.  Default Value: 86400000  Permissible Values: any positive value <= Long.MAX |
| deidHome | Description: Path of the home directory where the DeID software is installed. This property is available if and only if the user has opted to install DeID de-identification server.  Default Value: N/A  Permissible Values: N/A |
| deidDnyFolder | Description: Path of the directory containing dictionary for de-identification of report. This property is available if and only if the user has opted to install DeID de-identification server.  Default Value: N/A  Permissible Values: N/A |
| deidConfigFileName | Description: Configuration file name required for report de-identification native call. This property is available if and only if the user has opted to install DeID de-identification server.  Default Value: deid.cfg  Permissible Values: N/A |
| deidDTDFilename | Description: Data type definition file name required to extract text from the XML file returned from the de-id native method call. This property is available if and only if the user has opted to install DeID de-identification server.  Default Value: Dataset.dtd  Permissible Values: N/A |
| harvardScrubberDTDFileName | Description: Data type definition file name required to extract text from the XML file returned from the Harvard Scrubber de-identification tool. This property is available if and only if the user has opted to install Harvard Scrubber de-identification server.  Default Value: <DEIDENTFIER\_SERVER\_INSTALLATION\_DIR>/Scrubber.dtd  Permissible Values: N/A |
| harvardScrubberConfigFileName | Description: Configuration file name required for report de-identification. This property is available if and only if the user has opted to install Harvard Scrubber de-identification server.  Default Value: <DEIDENTFIER\_SERVER\_INSTALLATION\_DIR>/caTIES\_conf/ScrubberConfiguration.xml  Permissible Values: N/A |

### Running the Report Loader Server

The report loader server of caTIES takes HL7 format reports through input files, parses them, and then populates the caTissue Suite datastore. Along with creating identified Surgical Pathology Reports (SPR), the report loader server also handles cases like participant conflict, and SCG conflict.

|  |  |
| --- | --- |
| **Participants matching parameters**  Change these parameters to alter the way the matching participants are found during participants’ insertion. | |
| SPRSSNExact | Description: Exact match for SSN field  Default Value: 40 |
| SPRSSNPartial | Description: Mismatch for SSN field  Default Value: -140 |
| SPRPMIExact | Description: Exact match for medical record number field  Default Value:40 |
| SPRPMIPartial | Description: Mismatch for medical record number field  Default Value:-140 |
| SPRDOBExact | Description: Exact match for date of birth field  Default Value: 30 |
| SPRDOBPartial | Description: Mismatch for date of birth field  Default Value: -140 |
| SPRLastNameExact | Description: Exact match for last name field  Default Value: 20 |
| SPRLastNamePartial | Description: Mismatch for last name field  Default Value: -140 |
| SPRNameExact | Description: Exact match for first name field  Default Value: 10 |
| SPRNamePartial | Description: Mismatch for first name field  Default Value: -140 |

To alter the properties, change the parameters present in the file caTissueCore\_Properties.xml at <JBOSS-HOME>/server/default/catissue-properties/and report.loader.installation.dir/caTIES\_conf/conf.

For more information on the algorithm, refer to [SPR\_ParticipantMatchingAlgorithm](https://gforge.nci.nih.gov/docman/view.php/18/15154/ParticipantMatchingForSPRLoading.xls)

Steps for report loading:

1. Ensure the presence of a collection protocol in the database with the title specified in the caTIES.properties file present at report.loader.installation.dir in caTIES\_conf/caTIES.properties file.
2. Switch to the folder where the report loader is deployed (specified by the report.loader.installation.dir in caTIES\_conf/caTIES.properties file).
3. Run the following ANT task at the command prompt:

ant run\_report\_loader\_server

1. To stop the report loader server, run the following ANT task at the command prompt:

ant stop\_report\_loader\_server

### Running the Report De-identification Server

The de-identification server of caTIES is responsible for generating de-identified SPRs. De-identification is the process by which patient’s personal information is removed from the report text.

caTissue suite provides users with four different options for de-identification of reports:

1. DeID de-identifier
2. Harvard Scrubber de-identifier
3. Do-Nothing de-identifier
4. Custom de-identifier

The user can use any one of the following options for the de-identification process.

1. The DeID de-identifier:The DEID de-identification server takes help of a third party software called ‘De-ID’ to de-identification report text.**This option is supported on the Windows OS only**. Pre-requisites to use deid de-identifier are:

1. De-identification software must be pre-installed. The De-ID software is downloadable from <http://cliniscience.grouphub.com/> site.
2. Installation of the De-ID software automatically adds the home directory of De-ID to system’s path environmental variable. Verify that the path variable contains the path of De-ID’s home directory. If not, then add it explicitly.

Note: If Oracle server or Oracle client application and De-ID are installed on same machine, then the De-ID’s home path should be set before the Oracle application path in system’s path environment variable. Example: PATH = %DEID\_HOME%;%ORACLE\_HOME%

2. The Harvard Scrubber de-identifier: the de-identification server uses Harvard Scrubber, an open source tool for report text de-identification.

3. The Do-nothing de-identifier: the de-identification server simply copies identifier report text as it is to the de-identified report text. Report text is not de-identified with this option.

4. Custom de-identifier: If usershave written their own program to de-identify report text, they can use it with the de-identification server to de-identify the report text. Pre-requisitesto use custom de-identifier are:

1. The class implementing custom de-identification must extend the class caTissue Suite class edu.wustl.catissuecore.deidentifier.AbstractDeidentifier and should implement all the abstract methods specified in the above class.
2. Create a jar of source code and copy it to the <DEIDENTIFIER\_INSTALLATION\_DIR>/lib directory.
3. Verify the deidentifierClassName property in the <DEIDENTIFIER\_INSTALLATION\_DIR>/caTIES\_conf/caTIES.properties file.
4. Copy required resources to <DEIDENTIFIER\_INSTALLATION\_DIR> directory.

Where, <DEIDENTIFIER\_INSTALLATION\_DIR> is the path specified by the deid.installation.dir in the CATISSUE\_HOME/modules/caTIES/conf/caTIES\_conf/caTIES.properties file.

To run the report de-identification server, perform the following steps:

1. Switch to the folder where report de-identification server is deployed. This is specified by deidentifier.installation.dirinCATISSUE\_HOME/caTIES\_conf/caTIES.properties file.
2. To run the report de-identification server, run the following ANT task from the command prompt:

ant run\_deidentifier\_server

1. To stop report de-identification server, run the following ANT task from the command prompt:

ant stop\_deidentifier\_server

### Running the Report Concept-Code Server

The concept-coder server is responsible for concept-coding the report text. Concept coding is the process of semantic classification of medically-related words that appear in the report text.

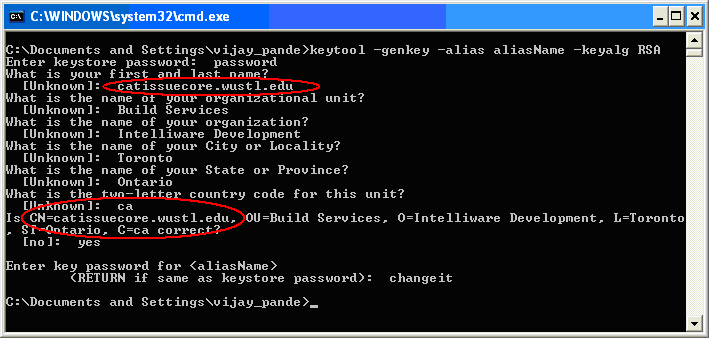
1. Verify the server name, server port, and data dictionary name in crole.xml file at:

http://<JBOSS\_HOST>:<JBOSS\_PORT>/gate.war/gate\_3\_1/application/plugins/caTIES/crole.xml

where JBOSS\_HOST and JBOSS\_PORT are the hostname and port number respectively on the JBoss server.

1. If JBoss is running on HTTPS then make sure that keystore file created contains a Common Name (CN) identical to the machine name on which JBoss is running. For more information please refer to the topic “Creating self signed certificate” explained at[Creating self signed certificate](#_Configuring_JBoss_Server)

Example: the keystore file for a concept-coder pointing to the demo site can be created as shown below:



1. Switch to the folder where the concept coder server is deployed (specified by concept.code.installation.dir in caTIES\_conf/caTIES.properties file).
2. Run the following ANT task at the command prompt:

ant run\_concept\_code\_server

1. To stop report concept coder server, run the following ANT task at command prompt:

ant stop\_concept\_code\_server

**Note:** Refer to the following summary table to start and run caTIES pipeline.

|  |  |  |
| --- | --- | --- |
| **Service** | **Start** | **Stop** |
| Report Loader | ant run\_report\_loader\_server | ant stop\_report\_loader\_server |
| De-identifier | ant run\_deidentifier\_server | ant stop\_deidentifier\_server |
| Concept Coder | ant run\_concept\_code\_server | ant stop\_concept\_code\_server |

# Testing the System

Once the JBoss server starts, you can access the application using a web browser. The URL pattern for the application is:<http://machine_name:port_number/catissuecore>

When the browser displays the home page of caTissue Suite, log on to the application by specifying the login credentials first.admin.emailAddressandfirst.admin.password.

## Running the Test Case Suite (API)[TBD]

This section describes the process of configuring and running the test suite of a CSM-enabled caCORE.

The CATISSUE\_CLIENT\_HOME/caTissueSuite\_Client/ directory contain files and libraries required to compile and run the caCORE CSM enabled client program.

Before using the caCORE API, define all required parameters in caTissueInstall.properties

(Refer section 2.3) present at location CATISSUE\_CLIENT\_HOME/caTissueSuite\_Clientand configure the host property in

CATISSUE\_CLIENT\_HOME/caTissueSuite\_Client/conf/remoteService.xml file.

**NOTE**: The label generator should be “On” for running API test cases.

**File:** remoteService.xml

Table : Configure remoteService.xml

|  |  |
| --- | --- |
| **Parameter** | **Details** |
| @@HOST@@:@@PORT@@ | Description: Host URL of the application to which the caCORE client will connect. If the tokens are not automatically replaced, replace the host parameter with host:port at which the server is configured. For example: catissuecore.wustl.edu:8080to access the caTissue Suite public demo site.  Default Value: N/A  Permissible Values: N/A |

### Running caCORE CSM Enabled Test Case Suite

To compile and run test suite:

1. Go to the command prompt and change your current directory to CATISSUE\_CLIENT\_HOME/caTissueSuite\_Client.
2. Edit the demo client program caTissueBaseTestCase.javalocated in package to caTissueSuite\_Client\src\edu\wustl\catissuecore\api\testand set the appropriate username, password, jbossURL and keyStorePath variables values.

The jbossURL variable value will be the same as mentioned in remoteService.xml file. If the jboss is secured, then mention the absolute path of the keystore variable.

Eg:

loginName = "admin@admin.com";

password = "Test123";

jbossURL = "https://localhost:8443/catissuecore";

keyStorePath = “G:/jboss-4.2.2.GA/server/default/conf/chap8.keystore”;

1. Run ANT task at the command prompt to compile and run the client program. The syntax of ANT task is: ant <target\_name>.

The following table describes the different targets of the ANT script:

Table : API related ANT targets

|  |  |
| --- | --- |
| **Task** | **Description** |
| compile\_junit\_TestCases | Compiles all the necessary classes to run the test case suite. |
| runNightlyBuild | Runs the test case suite and creates test case reports. |

**Note:**If you create any new test cases, invokesetLogger() inside your test case.

For example:

private static void setLogger(Object object)

{

Logger.out = org.apache.log4j.Logger.getLogger(object.getClass());

}

FluidSpecimen specimen = new FluidSpecimen ();

**setLogger(specimen);**

List specimenList = appService.search(FluidSpecimen.class, specimen);

### Getting Test Suite Result and Detailed Report

After executing the test suite you can see the results in the reports directory. The test suite will create a report directory named

.CATISSUE\_CLIENT\_HOME/caTissueSuite\_Client/Nightly\_Build\_Report/.

### Source code of Test Cases

If you want to write your own API test program or edit the existing one, look for the source code at:CATISSUE\_CLIENT\_HOME\caTissueSuite\_Client\src\edu\wustl\catissuecore\api\test

## Running Test Cases Suite (API) with HTTPS

If the JBoss server is configured with HTTPS, perform the following steps to run Test Case Suite (API). To configure JBoss server with HTTPS, please refer

### Configure Client for HTTPS

Perform the following steps to configure client with HTTPS:

1. Generate the keystore file on the client machine. To generate the keystore file,perform the first step mentioned in the section [Configuring JBoss Server to deploy caTissue using HTTPS](#_Configuring_JBoss_Server).
2. Update the source code in the authentication section and set the path of the keystore file to the system property javax.net.ssl.trustStore.

For example, if the keystore created is stored in c://catissue/catissue.keystore, add the following line of code along with the authentication

ClientSession cs = ClientSession.getInstance();

System.setProperty("javax.net.ssl.trustStore","c://catissue/catisue.keystore"); cs.startSession(<login-name>,<password>);

1. In theremoteService.xmlfile, set the value for the propertyserviceURLto[http**s**://@@HOST@@:@@PORT/catissuecore/http/remoteService](https://@@HOST@@:@@PORT/catissuecore/http/remoteService)

where @@HOST:@@PORT should be replaced with host URL as shown in

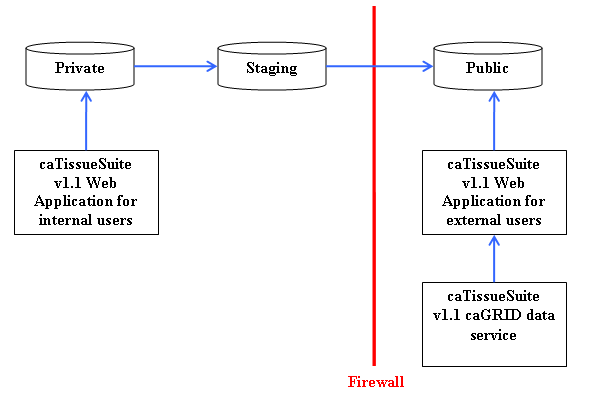
Table 15: Configure remoteService.xml.

# Private/Public Data Store

caTissue Suite contains patient/participant-identifying data. Since the database containing identified data should always reside within the institutional IT firewall, itcan be challenging to get caBIG™ data sharing commitments approved by the IRB.

The private/public data store functionality of Suite solves this problem by providing a mechanism by which a caTissue database is de-identified and then copied into a database which is publicly accessible. This public database can now be made freely accessible to authorized researchers worldwide outside the institutional firewall.

The following diagram depicts the data flow from private to public databases.



Please note the following about the public database:

1. You can deploy caTissue Suite v1.1 on the public database.
2. The public database is mainly targeted towards researchers who want to query your instance of caTissue. They can query by either using the Query interface,caTissue Suite API or caTissue caGrid data service.
3. The public database is currently read-only and will be overwritten the next timeit issynchronized with the private database.
4. Future versions of caTissue might allow users to create or accept biospecimen distribution request on the public database.

## Deploying the Public Database

To deploy the public database, you need to:

1. Configure the privatePublic.properties file.
2. Configure mask.sql to perform additional data masking as per individual needs.
3. Generate the commands.properties file.
4. Schedule the ANT task migrate to run daily by using Scheduled Task on Windows and cron job on Linux.

**Configuring the privatePublic.properties file**

**Location:**CATISSUE\_HOME/modules/public\_private\_migrator/privatePublic.properties

Table : Private Public Datastore Configuration Parameter

| **Property Name** | **Description** |
| --- | --- |
| databaseType | Description : Database type used (mysql/oracle)  Default Value: N/A  Permissible Values: N/A |
| privateDBPort | Description: Port used by private caTissue Suite instance  Default Value: N/A  Permissible Values: N/A |
| privateDBName | Description: Database name of private caTissue Suite instance.  Default Value: N/A  Permissible Values: N/A |
| privateDBHost | Description: Database Host name of private caTissue Suite instance.  Default Value: N/A  Permissible Values: N/A |
| privateDBUserName | Description: Database user name of private caTissue Suite instance.  Default Value: N/A  Permissible Values: N/A |
| privateDBPassword | Description: Database password of private caTissue Suite instance.  Default Value: N/A  Permissible Values: N/A |
| privateDBTablespace | Description: Tablespace ofdatabase of private caTissue Suite instance. This is required only for Oracle.  Default Value: N/A  Permissible Values: N/A |
| privateDBIndexTablespace | Description:Index tablespace ofdatabase of private caTissue Suite instance. This is required only for Oracle.  Default Value: N/A  Permissible Values: N/A |
| stagingDBName | Description: Database name of staging database.  **Note:** If a database with this name does not exist, the private-public data synchronizer will create a database by this name. If it exists, then it will drop the database and recreate the database.  Default Value: N/A  Permissible Values: N/A |
| stagingDBUserName | Description: Database user name of staging database.  Default Value: N/A  Permissible Values: N/A |
| stagingDBPassword | Description: Database password of staging database.  Default Value: N/A  Permissible Values: N/A |
| stagingSystemUserName | Description:Administrator user name ofdatabase of staging caTissue Suite instance. This is required only for Oracle. This should be ‘system’ or the user having rights equal to user ‘system’.  Default Value: N/A  Permissible Values: N/A |
| stagingSystemPassword | Description: Administrator password ofdatabase of staging caTissue Suite instance. This is required only for Oracle. (Corresponding password for the user name provided in the property stagingSystemUserName).  Default Value: N/A  Permissible Values: N/A |
| publicDBName | Description: Database name of public caTissue Suite instance.  **Note:** If adatabase with this name does not exist, the private-public data synchronizer will create a database by this name. If it exists, then it will be completely overwritten.  Default Value: N/A  Permissible Values: N/A |
| publicDBHost | Description: Database host name of public caTissue Suite instance.  Default Value: N/A  Permissible Values: N/A |
| publicDBUserName | Description: Database user name of public caTissue Suite instance.  Default Value: N/A  Permissible Values: N/A |
| publicDBPassword | Description: Database password of public caTissue Suite instance.  Default Value: N/A  Permissible Values: N/A |
| publicDBPort | Description : Port used for public caTissue Suite instance.  Default Value: N/A  Permissible Values: N/A |
| publicSystemUserName | Description: Administrator user name ofdatabase of public caTissue Suite instance. This is required only for Oracle. This should be ‘system’ or the user having rights equal to user ‘system’.  Default Value: N/A  Permissible Values: N/A |
| publiceSystemPassword | Description: Administrator password ofdatabase of public caTissue Suite instance (required only for Oracle). (Corresponding password for the user name provided in the property publicSystemUserName).  Default Value: N/A  Permissible Values: N/A |
| publicDBTNSName | Description: Database TNS name of public caTissue Suite instance (required only for Oracle).  Default Value: N/A  Permissible Values: N/A |

**Note:**Staging and Public databases will not be dropped and recreated if they are in use during migration. Make sure that the JBoss server hosting the public instance is stopped before the migration.

**Configure mask.sql**

User can configure additional SQL statements to perform additional data de-identification. For this add/modify the SQL statements in mask.sql located at CATISSUE\_HOME/modules/public\_private\_migrator/sql

**Generating thecommand.properties File**

To generate the command.propertiesfile:

* Switch to the CATISSUE\_HOME/modules/public\_private\_migrator directory and run the following ANT task:

ant generate\_command\_file

**Running the migration tool**

To run the migration tool, run the following command:

ant migrate

**Running the tool for only data de-identification**

User can execute only data de-identification target. This command does not create the staging and/or public database as carried by target migrate mentioned above. To run only de-identification:

* Configure privatePublic.properties.The data de-identification runs only on staging database.
* Switch to the CATISSUE\_HOME/modules/public\_private\_migrator directory and run the following ANT task:

ant maskData

**CAUTION:** The only de-identification command is executed only on staging database. Thus ensure that the staging database properties are appropriately configured in privatePublic.properties

**Synchronizing the public database on a regular basis**

You can set this task to be run as a scheduled task. For example: nightly, weekly, and so on. You can use the Operating System’s task scheduler like Unix cron(8) or the Windows Scheduler to execute themigratecommand.

**CAUTION:** Every time the migrate command is executed, the public database is recreated, and any updates to the public database are lost. The synchronization tool does not make a backup of the public database. Optionally, you can also add a database backup creation in the scheduled task. For oracle once dump file is created the tool does not automatically delete or overwrite the private database dump. The dump file for private database dump should be manually removed or cron job should be configured to delete dump file before next run of the tool.

**NOTE:** For oracle, the public database created will always have ‘users’ as default tablespace. This is because we are using imp/exp command of the oracle to import/export dump which only works with ‘users’ tablespace which is the default tablespace for oracle. Also, for oracle it is mandatory that the tool should run on the machine where private database is located.

## Deploying the Web Application on the Public Database

The process of deploying the web application on the public database is similar to deploying it on the private database. Please follow the instructions in Section 2 - Deploying the Web Application.

## Known Issues

The private->public migrator does not work on MySQL->LINUX set up. Once the command.properties file is generated, follow the below steps:

1. Create a new file at CATISSUE\_HOME/modules/public\_private\_migratorby name private.sh with below contents:

mysql -u ***stagingDBUserName***-p***stagingDBPassword*** -h ***stagingDBHost*** -P ***stagingDBPort*** <./sql/drop\_staging\_db\_mysql.sql

mysqldump -u ***privateDBUserName*** -p***privateDBPassword*** -h ***privateDBHost*** -P ***privateDBPortprivateDBName***|mysql -u ***stagingDBUserName***-p***stagingDBPassword*** -h ***stagingDBHost*** -P ***stagingDBPortstagingDBName***

**Note:** Replace the parameters(marked in italics and bold) in above two statements by actual values.

1. Create a new file by name public.sh with below contents at CATISSUE\_HOME/modules/public\_private\_migrator

mysql -u ***publicDBUserName***-p***publicDBPassword*** -h ***publicDBHost***-P ***publicDBPort***< ./sql/drop\_public\_db\_mysql.sql

mysqldump -u ***stagingDBUserName*** -p***stagingDBPassword*** -h ***stagingDBHost*** -P ***stagingDBPortstagingDBName***|mysql -u ***publicDBUserName*** -p***publicDBPassword*** -h ***publicDBHost*** -P ***publicDBPortpublicDBName***

**Note:** Replace the parameters(marked in italics and bold) in above two statements by actual values.

1. Replace the command.properties file contents with below contents:

#Commands file for public private DB

#properties related to private database

createPrivateDump=./private.sh

dropStagingDB=

createStagingDB=

#properties related to public database

createStagingDump=./public.sh

dropPublicDB=

createPublicDB=

1. Run the migration tool using the following command:

antmigrate

# Deploying caTissuecaGRID Data Service [TBD]

caTissue Suite v1.2 is grid-enabled wherein all the caGrid queries are routed through a single designated caTissue user account(i.e. a caGrid user will not be directly provisioned in caTissue). In a future release of caTissue, one will be able to provision the caGrid users in caTissue.

**CAUTION:** The single designated caTissue user should be a user with role scientist who does not have access to any identified data AND is not a principal investigator or coordinator of any collection protocol.

The caTissue instance to which the grid service is going to point, should be deployed on a HTTP JBoss instance.In this release, the grid service is not supported if caTissue is deployed on a secure JBoss instance. Please refer to the Private/Public Data Store chapter for details on how to de-identify caTissue data.

## Background

The caTissue caGrid data service requires that a caGrid service (or grid node) is deployed and functioning. Below are detailed instructions for deploying the service but assumptions are made that might not fit your environment or deployment needs. For more information or assistance with deploying the caGrid service please see the caGrid Knowledge Center<https://cabig-kc.nci.nih.gov/CaGrid/KC/index.php/Main_Page>. Multiple data services can use one grid node but require separate JBoss containers. Please see the TBTT FAQ on changing the JBoss default ports to run multiple JBoss containers from the same machine <https://cabig-kc.nci.nih.gov/Biospecimen/KC/index.php/Changing_JBoss_default_ports>. By changing the default ports you can deploy multiple versions of JBoss without port conflicts.

The deployment steps for the caGrid Service are:

1. Deploy JBoss 4.0.4
2. Deploy Java JDK 1.5
3. Deploy the caGrid 1.2 service (using the caGrid installer will automatically download Ant and Globus)
4. Create the secure caGrid service

The deployment steps for the caTissue data services are:

1. Delpoy Globus into the JBoss for the desired caTissue service
2. Delpoy and configure the desired caTissue service (Installing a caTissue service:)
3. Start the caTissue data service JBoss

## Prerequisites

Note: Please skip this step if you already have a caGrid service installed on the same machine where you are deploying caTissue caGRID data service.

| **Software Name** | **Version** | **URL** | **Install Guide** |
| --- | --- | --- | --- |
| caGrid | 1.2 | <http://gforge.nci.nih.gov/frs/download.php/3738/caGrid-installer-1.2.zip> | http://gforge.nci.nih.gov/plugins/scmcvs/cvsweb.php/~checkout~/cagrid-1-0/Documentation/docs/installer/caGrid-1-2\_Installer\_Guide.pdf?rev=HEAD;content-type=application%2Foctet-stream;cvsroot=cagrid-1-0 |
| Globus Toolkit | 4.0.3 | <http://gforge.nci.nih.gov/frs/download.php/1334/ws-core-enum-4.0.3.zip> | <http://www.globusconsortium.org/tutorial/ch6/> |
| Jboss | 4.0.4 GA | <http://labs.jboss.com/jbossas/downloads/> |  |

**Steps to deploy prerequisites:**

* Install JDK 1.5and set the environment variableJAVA\_HOME pointing to the location where Java is installed.
* Install JBoss and set the environment variableJBOSS\_HOME pointing to the location where JBoss is installed.

Note: “JBOSS\_HOME” should be set “temporarily” for the session using the below mentioned command

For Windows

set JBOSS\_HOME=path\_to\_the\_correct\_jboss

For Linux

export JBOSS\_HOME=path\_to\_the\_correct\_jboss

* Make sure that JBoss bin/ directory is present in the system PATH variable.
* Make sure that Apache Ant is installed on your machine and environment variable ANT\_HOMEis set properly.
* Download and install Globusand set environment variableGLOBUS\_LOCATIONpointing to the installation directory.
* Download and install caGrid 1.2and set environment variableCAGRID\_LOCATIONpointing to the installation directory. Follow the caGrid 1.2 installation guide.

## Requesting a caGrid Host Certificate

**Note**: You will only need to do this step once per caGrid service per machine. The steps here assume the production IndexService as a target grid. If the caGrid installed is pointing to training grid, you may need to configure caGrid re-targeting the official production grid. For more information please visit <http://www.cagrid.org/wiki/CaGrid:How-To:ChangeTargetGrids> on how to change the target grid.

1. To request a host credential complete the following steps:
   1. Launch the GAARDS UI by running following commands:
      1. cd CAGRID\_LOCATION
      2. ant security
   2. If you do not have a caGrid user account, you have to first create one. To request a user account please complete the following steps:
      1. Launch the GAARDS UI
      2. From the top menu bar in the GAARDS UI select *User Management* =>*Local Account* =>*Registration*.
      3. From the *Service* drop down, select <https://cagrid-dorian.nci.nih.gov:8443/wsrf/services/cagrid/Dorian>
      4. Specify a username and password; this will be the username and password that you use to authenticate with the Dorian IdP.

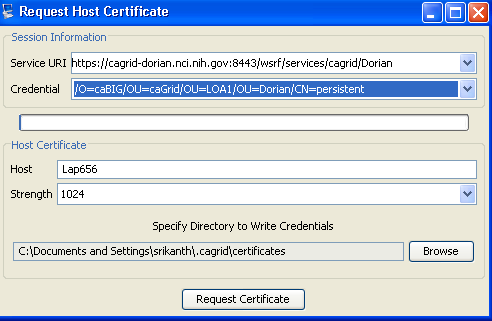
**NOTE:** The password must contain at least 10 and at most 20 characters, as well as contain at least one capital letter, one number, one non-alphanumeric symbol, and not contain any dictionary words.

* + 1. Finally enter your personal information and click the *Apply* button.

In most case your account will be immediately approved and you should be able to create grid credentials.

**Note:** If you encounter the error "[Caused by: Failure unspecified at GSS-API level [Caused by: Unknown CA]]” then you can run "ant syncWithTrustFabric" from the caGrid directory to synchronize the trust fabric. Retry registering the user after running this command.

* 1. Login to caGrid:
     1. Click the Login button on the toolbar in the GAARDS UI to open the Create Proxy or login window.
     2. To login, first specify the Dorian that maintains your grid user account by selecting the Dorian URI from the Identity Federation Service drop down.
     3. The GAARDS UI is pre-configured with a list of Dorians through its configuration file. If the Dorian you wish to select is not in the list, enter it.
     4. Select the lifetime of your grid proxy; this is how long your credentials are good for. Select from the Lifetime drop down for hours, minutes, and seconds.
     5. Specify how many times your credentials can be delegated. Delegating your credentials gives another party the ability to act on your behalf or as you. For example if you allow a delegation path length of 1, you allow a grid service you connect with to connect to another grid service as you. However, the second grid service would not be able to connect to another grid service as you. By default the delegation path length is set to 0, and in most cases it will not need to be increased. If you wish to increase the delegation path length, change the Delegation Path Length text field.
     6. Specify the Authentication Service you wish to authenticate with by selecting the URI from the Authentication Service drop down. After selecting the Authentication Service you will be supplied with input fields to enter in the information you need to authenticate. Provide the information requested. In the example below a Dorian IdP is selected. Since the Dorian IdP requires a user id and password, input fields display. The GAARDS UI is pre-configured with a list of Identity Providers if your Identity provider is not listed in the Identity Provider drop down. Add it by editing the GAARDS UI configuration file.
     7. Once you have entered the required IdP Authentication Information, click the Authenticate button to 1) authenticate you with your Identity Provider, 2) obtain a SAML Assertion from your Identity Provider, and 3) contact Dorian using the SAML Assertion to facilitate the creation of a grid proxy. Once the grid proxy is created the Create Proxy window closes and the Proxy Manager window opens with the newly created proxy shown. The Proxy Manager window allows the management of grid proxies or grid credentials that you locally created.
     8. Click the *Set Default* button.
  2. Request for host certificate:
     1. From the *MyAccount* menu select *Request a Host Certificate*, this will launch the *Request a Host Certificate* window.



* + 1. From the Service URI drop down select

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

* + 1. In the Host text box, enter the name of the host you are requesting a host credentials for.
    2. Specify the directory on the file system where the host credentials should be written to, this can be done by clicking the *Browse* button.
    3. Click the Request Certificate button.
  1. Close the GAARDS UI
  2. After clicking the *Request Certificate* button, the UI will submit the host certificate request to Dorian. Upon receiving the request Dorian will immediately approve the request and the host credentials (certificate and private key) will be written to the directory specified. The file containing the certificate will be named *THE\_HOSTNAME\_YOU\_ENTERED-cert.pem*, the file containing the private key will be named *THE\_HOSTNAME\_YOU\_ENTERED-key.pem*. You will need these two files in step 2.c

Note: Host credentials are issued only once. Backup these two files, in case if they are lost.

## Deploying and Securing Globus

Note: This setp must be performed for each caTissue data service deployed.

1. If you are behind a proxy server, then set the ANT\_OPTS environment variable to:

-Dhttp.proxyHost=pchtroxy\_server\_url -Dhttp.proxyPort=proxy\_server\_port -Dhttp.proxyUser=username -Dhttp.proxyPassword=password

Please replace the parameters proxy\_server\_url, proxy\_server\_port, username and password with the corresponding values.

1. Open file CAGRID\_LOCATION/antfiles/jboss/jboss.xml and perform the following changes
   * Search for string: <target name="deploySecureJBoss"
   * Update the defaultPort to the appropriate port used for the caGrid service
   * After editing, the section should look like as follows:

<property name="defaultPort" value="SERVICE\_PORT"/>

<property name="webapp.dir" value="${jboss.dir}/server/default/deploy/${webapp.name}.war" />

Note: If the jboss instances are set up under one jboss container, then also perform the following change:

* + Update the webapp.dir to delete the string /server/default from the value
  + After editing, the section should look as follows:

<property name="defaultPort" value="SERVICE\_PORT" />

<property name="webapp.dir" value="${jboss.dir}/deploy/${webapp.name}.war"/>

1. To deploy the Globus on JBoss run the following commands:
   * cd CAGRID\_LOCATION/antfiles/jboss
   * ant -f jboss.xml deploySecureJBoss -Djboss.dir=JBOSS\_HOME

Please replace the JBOSS\_HOME string with the home directory of JBoss where you wish to deploy grid service.

**Note:**If you have not set the ANT\_OPTS parameter and are dbehind a proxy server, you will encounter a connection timeout error after the above ant command.

1. Update JBOSS\_HOME\server\default\deploy\jbossweb-tomcat55.sar\server.xml
   1. Replace the <server> element to (usually he first line of the file)   
      <Server debug="0" port="8005" shutdown="SHUTDOWN">
   2. Commentout the complete element

<Connector port="8080" disableUploadTimeout="true"/>

* 1. Add a new<connector>element for the jboss port set for caTissue as –

<Connector

className="org.globus.tomcat.coyote.net.HTTPSConnector"

port="SERVICE\_PORT" maxThreads="150" minSpareThreads="25"

maxSpareThreads="75" autoFlush="true"

disableUploadTimeout="true" scheme="https"

enableLookups="true" acceptCount="10" debug="0"

protocolHandlerClassName="org.apache.coyote.http11.Http11Protocol"

socketFactory="org.globus.tomcat.catalina.net.BaseHTTPSServerSocketFactory"

cert="complete\_path/THE\_HOSTNAME\_YOU\_ENTERED-cert.pem"

key="complete\_path/THE\_HOSTNAME\_YOU\_ENTERED-key.pem"

/>

**NOTE:**

1. For cert and key file names, please replace the text in blue color with the fully qualified path along with the file names generated in step 1.e.
2. Please replace the port number (SERVICE\_PORT) with the JBoss port set for caTissue caGrid service environment.
   1. Add following element in the<Engine name="jboss.web" ... > section:  
      <Valve className="org.globus.tomcat.coyote.valves.HTTPSValve55"/>
3. Update file JBOSS\_HOME\server\default\deploy\wsrf.war\WEB-INF\web.xml
   1. Search for lines:

<param-name>defaultPort</param-name><param-value>8443</param-value>

* 1. Change port 8443 to the JBoss port set for caTissue grid service.

1. Secure Globus by adding the host credentials entries into global\_security\_descriptor.xml file located at JBOSS\_HOME\server\default\deploy\wsrf.war\WEB-INF\etc\globus\_wsrf\_core.

The content of file must look as shown below.

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

<securityConfig xmlns="http://www.globus.org">

<credential>

<key-file value="complete\_path/THE\_HOSTNAME\_YOU\_ENTERED-key.pem"/>

<cert-file value="complete\_path/THE\_HOSTNAME\_YOU\_ENTERED-cert.pem"/>

</credential>

</securityConfig>

**NOTE:** For cert and key file names, please replace the text in blue color with the fully qualified path along with the file names generated in step 1.e.

1. Add a new <parameter> element under <globalConfiguration> element in server-config.wsdd file located at JBOSS\_HOME\server\default\deploy\wsrf.war\WEB-INF\etc\globus\_wsrf\_core as shown below:

<globalConfiguration>

...

<parameter name="containerSecDesc"

value="jboss.server.home.dir\deploy\wsrf.war\WEB-INF\etc\globus\_wsrf\_core\global\_security\_descriptor.xml" />

...

<globalConfiguration>

**NOTE:** In the above text, replace jboss.server.home.dir with the location to the JBoss server’s home directory.

1. **File:** log4j.xml

**Location:**JBOSS\_HOME/server/default/conf

Add the following to the log4j file:

<category name="org.globus.gsi.gssapi">

<priority value="WARN" />

<appender-ref ref="FILE"/>

</category>

<category name="STDOUT">

<priority value="WARN" />

<appender-ref ref="FILE"/>

</category>

**Note:** Log4j is an open source application for logging application messages. Please refer to <http://logging.apache.org/log4j/docs/> for more details.

In log4j.xml file Jboss defines two main appenders called ‘FILE’ appender and ‘CONSOLE’ appender. FILE appender appends log information into server.log file and CONSOLE appender shows logs on console. By default logger level of FILE appender is ‘DEBUG’ and for CONSOLE appender is ‘INFO’. You may change the logger level as per your requirement by modifying this parameter. Parameter name is ‘Threshold’ and allowed values are ‘DEBUG’,’INFO’,’WARN’,’ERROR’,’FATAL’ and ‘OFF’.

Example:

<appender name="FILE" classorg.jboss.logging.appender.DailyRollingFileAppender">

<errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>

<param name="File" value="${jboss.server.home.dir}/log/server.log"/>

<param name="Append" value="false"/>

<param name="Threshold" value="INFO"/>

</appender>

In the above example logger level for FILE appender is set to ‘INFO’.

1. Find following in JBOSS\_HOME\conf\log4j.xml

<appender name="FILE"

1. Append following before </appender>

<filter class="org.jboss.logging.filter.TCLFilter">

<param name="AcceptOnMatch" value="true"/>

<param name="DeployURL" value="wsrf.war"/>

</filter>

The filter steps will skip writing of the unwanted logs into server.log file. The same can be repeated with tag <appender name="CONSOLE"...>, for not displaying logs on the console.

## Installing a caTissue service:

1. Open a command prompt and change your current directory to CATISSUE\_HOME/modules/caGrid\_service
2. Update the filecredentials.properties

| Parameter Name | Details |
| --- | --- |
| user.name | caTissue user name to which the caGrid users should be mapped to  **CAUTION:** The Administrator should make sure that the user configured here should be of Scientist role in the application. That user should not be principal investigator or coordinator of any collection protocol.  **NOTE:**  This caTissue account will expire as per the property settings in password.expire\_after\_n\_days.  **It is highly recommended that you set a calendar alert to notify administrators that this account password will expire in n days. The grid service will not be accessible if the account expires.** |
| password | Password of the common user for which all the caGrid users will be mapped to. |
| keystoreFilePath | Description: Path to the keystore file if the caTissue is deployed as HTTPS. This key should match to the key of caTissue web application instance. The path must be separated by ‘/’ character. To generate the keystore file, perform the first step mentioned in the section[Configuring JBoss Server to deploy caTissue using HTTPS](#_Configuring_JBoss_Server)  Default Value: N/A  Permissible Value: N/A |

After editing the file, copy it to {user.home}/catissueservicewhere{user.home}is the Windows default user directory (for example, C:\Documents and Settings\srikanth\catissueservice), and in Unix or Linux the user home directory (e.g. /home/srikanth/catissueservice)

**NOTE:** This file will have to be copied to the user home of the user under which the JBoss server is going to be run.

1. Update file:serviceMetadata.xml

**Location:** CATISSUE\_HOME/modules/caGrid\_service**/**etc

Locate ns3:PointOfContactaffiliation and add in the details for email, firstName, lastName and role. PhoneNumber is optional.

Locate ns9:ResearchCenter displayName and update details such as your research center displayName, shortName and address information. These details will be used to place the service on the Google map on the Grid Portal page <http://cagrid-portal.nci.nih.gov/web/guest> .

Details about the person maintaining the service should also be added under PointOfContact.

1. Update file: service.properties

**Location:**CATISSUE\_HOME/modules/caGrid\_service

The property cqlQueryProcessorConfig\_appserviceUrl should point to the host and port of the caTissue web application instance.

The current value of the parameter is

http\://host\:port/catissuecore/http/remoteService

where the host and port with the hostname and port number of the caTissue web application.

For example, http\://testserver3.wustl.edu\:8080/catissuecore/http/remoteService

1. Update fileCATISSUE\_HOME/modules/caGrid\_service/build-deploy.xml
   1. Search for line<property name="jboss.dir" value="${env.JBOSS\_HOME}"/>
   2. Replace ${env.JBOSS\_HOME} with the folder name of JBOSS\_HOME. For example,

"<property name="jboss.dir" value="/usr/local/jboss-4.0.4.GA/”From this folder, run the following commands:

>ant clean

>ant all

>ant deployJBoss

1. Start the JBoss server
   1. Open a command prompt
   2. Change directory: cd %JBOSS\_HOME%\bin
   3. Run command:run.bat -c default

**NOTE:** You can verify that the grid service has been successfully deployed by typing the following URL in the browser: https://<hostname>:<port>/wsrf/services/cagrid/CaTissueSuite

1. To stop the JBoss server
   1. Open a command prompt
   2. Change directory: cd %JBOSS\_HOME%\bin
   3. Run command:shutdown.bat -S

## Running the caGrid test queries:

1. Change directory toCATISSUE\_HOME/modules/caGrid\_service
2. Update the file run-tools.xmlto configure the URL for caTissue web application

<property name="service.url" value="https://host:port//wsrf/services/cagrid/CaTissueSuite"/>

In the above string, replace the host and port strings with the corresponding values of the caTissue caGrid service.

1. Following example the CQL queries are located at CATISSUE\_HOME/modules/caGrid\_service/cqls folder

* Query1: Return Molecular specimens available in a collection protocol = 'CP Prostate Cancer'
* Query2: Return Tissue specimens which have been thawed
* Query3: Return Tissue specimens that were fixed in formalin 30 minutes or less and were embedded in low melting point paraffin
* Query4: Return RNA extracts derived from specimens where the tissue fixative was not formalin
* Query5: Return particiapnts whose is having medical record from site ='Washu'
* Query6: Return Molecular specimen where Parent spcimen is of type 'Fluid' where CP title like '%Prostate%'
* Query7: Return Cell specimen of Male participant and PPI ='12345'
* Query8: Return number of Tissue specimen collected in collection protocol = 'CP Prostate Cancer'
* Query9: Return malignant prostate DNA specimens from African American participants who are on a CP with the word "Prostate" in the title
* Query10: Return breast frozen malignant tissue specimens where tissue review event shows greater than 90% neoplastic cellularity
* Query11: Return Tissue specimens which have been collected by the 'Lavage' procedure && that have been collected in an unacceptable quality
* Query12: Return Patient for Tumor identified by needle Biopsywhere CP title like '%Prostate%'

1. To execute the queries, run the following commands from CATISSUE\_HOME/modules/caGrid\_service:
   1. All queries: ant runAll
   2. Individual test cases: ant runQuery1, ant runQuery2, ant runQuery3 and so forth
2. For more information on writing CQL queries, please refer to - <http://www.cagrid.org/mwiki/index.php?title=Data_Services:CQL>

**Known issues:**

Bug 9903: Cannot query on base classes via CQL due to a known issue on caGrid.

Refer to <https://cabig-kc.nci.nih.gov/Biospecimen/KC/index.php/CaGridKnownIssues> for details.

Bug 13057: With maxRecordsPercaCOREQueryAllowed = 50000 throws java.lang.OutOfMemoryError while executing cql query

# AppendixA – MySQL Case Sensitivity on Linux

This Appendix describes the MySQL case sensitivity issue on Linux and the various deployment targets.

## MySQL Case Sensitivity Issue on Linux

A MySQL server running on Linux is case-sensitive with regards to database and table names. This property is defined by the lower\_case\_table\_names system variable.This table can be configured when starting the MySQL server. For example: using the command mysqld.

Please refer to Bug #[447](http://nagarajanlab.wustl.edu/bugs/show_bug.cgi?id=447) for more details on this issue.

To set the system variable on Linux:

If the file my.cnf is available in the directory where MySQL is installed, then add the following line in the file.

lower\_case\_table\_names=1

If the file is not available, then create the file my.cnf in the folder /etc using the following commands:

cat > /etc/my.cnf << EOF

[mysqld]

datadir=/var/lib/mysql

socket=/var/lib/mysql/mysql.sock

lower\_case\_table\_names=1

[mysql.server]

user=mysql

basedir=/var/lib

[safe\_mysqld]

err-log=/var/log/mysqld.log

pid-file=/var/run/mysqld/mysqld.pid

EOF

Restart MySQL service using the following command:

/sbin/service mysql restart

**NOTE:** After setting lower\_case\_table\_names to 1 on Linux, you need to recreate the caTissue Suite database.

# Appendix B – Clinical Annotation Forms List

Clinical and pathology forms are modeled based on the CAP checklists and loaded into caTissue using the Dynamic Extensions (DE) feature.

Using the Show\_Hide\_Forms.xml file, administrators can select which forms are displayed in the User Interface. The table below contains form name that can be added to the Show\_Hide\_Forms.xml file.

## Form Names

| **Form Name** | **Group Name** |
| --- | --- |
| LabAnnotation | clinical\_annotation |
| FamilyHistoryAnnotation | clinical\_annotation |
| TreatmentRegimen | clinical\_annotation |
| TreatmentAnnotation | clinical\_annotation |
| RadRXAnnotation | clinical\_annotation |
| ChemoRXAnnotation | clinical\_annotation |
| EnvironmentalExposuresHealthAnnotation | clinical\_annotation |
| SmokingHealthAnnotation | clinical\_annotation |
| AlcoholHealthAnnotation | clinical\_annotation |
| HealthExaminationAnnotation | clinical\_annotation |
| RecurrenceHealthExaminationAnnotation | clinical\_annotation |
| DistantRecurrenceHealthExaminationAnnotation | clinical\_annotation |
| LocalRecurrenceHealthExaminationAnnotation | clinical\_annotation |
| NoEvidentDiseaseHealthAnnotation | clinical\_annotation |
| NewDiagnosisHealthAnnotation | clinical\_annotation |
| SpecimenBaseSolidTissuePathologyAnnotation | pathology\_specimen |
| ColorectalSpecimenPathologyAnnotation | pathology\_specimen |
| PancreasSpecimenPathologyAnnotation | pathology\_specimen |
| MelanomaSpecimenPathologyAnnotation | pathology\_specimen |
| CNSSpecimenPathologyAnnotation | pathology\_specimen |
| ProstateSpecimenPathologyAnnotation | pathology\_specimen |
| KidneySpecimenPathologyAnnotation | pathology\_specimen |
| LungSpecimenPathologyAnnotation | pathology\_specimen |
| BreastSpecimenPathologyAnnotation | pathology\_specimen |
| BasePathologyAnnotation | pathology\_scg |
| BaseHaematologyPathologyAnnotation | pathology\_scg |
| BaseSolidTissuePathologyAnnotation | pathology\_scg |
| ProstatePathologyAnnotation | pathology\_scg |
| RadicalProstatectomyPathologyAnnotation | pathology\_scg |
| NeedleBiopsyProstatePathologyAnnotation | pathology\_scg |
| RetropubicEnucleationPathologyAnnotation | pathology\_scg |
| TransurethralProstaticResectionPathologyAnnotation | pathology\_scg |
| PancreasPathologyAnnotation | pathology\_scg |
| LungPathologyAnnotation | pathology\_scg |
| LungBiopsyPathologyAnnotation | pathology\_scg |
| LungResectionBasedPathologyAnnotation | pathology\_scg |
| KidneyPathologyAnnotation | pathology\_scg |
| KidneyBiopsyBasedPathologyAnnotation | pathology\_scg |
| KidneyNephrectomyBasedPathologyAnnotation | pathology\_scg |
| MelanomaPathologyAnnotation | pathology\_scg |
| CNSPathologyAnnotation | pathology\_scg |
| BreastPathologyAnnotation | pathology\_scg |
| ColorectalPathologyAnnotation | pathology\_scg |
| LocalExcisionBasedColorectalPathologyAnnotation | pathology\_scg |
| ExcisionalBiopsyBasedColorectalPathologyAnnotation | pathology\_scg |
| ResectionBasedColorectalPathologyAnnotation | pathology\_scg |

# Appendix C –Steps for deploying caTissue with IDP

Below are the steps to deploy caTissue v1.2 with IDP. By default always caTissue will be deployed with IDP. Following are the steps to configure it.

## Idp Configuration

Before deploying the application, please ensure that all required parameters are defined in the following file.

**File:**install.properties or upgrade.properties

**Location:**CATISSUE\_HOME

For information on how to update this file please refer to sectionPerforming Pre-Installation Configuration(IDP Properties sub-section)

**File:**IDPAuthentication.xml

**Location:**JBOSS\_HOME/server/default/catissuecore-properties/

This is a XML file which contains all the parameters which are required to configure properly before deploying application. Following is the example auth file.

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

<IDPS>

<IDP name="IDP\_NAME">

<class name="AUTH\_MANAGER\_CLASS\_NAME" />

<idpdetails>

<defaultidp>DEFAULT\_IDP\_VALUE</defaultidp>

<displayname>DISPLAY\_NAME</displayname>

</idpdetails>

<idp-configuration>

If IDP points to LDAP then specify all the LDAP details in this section else

keep it empty.

</idp-configuration>

<user-attributes>

If required any user attributes such as first name last name could be specified

here.

</user-attributes>

<migration-details>

<migrator>

<rule-class>RULE\_CLASS\_NAME</rule-class>

<migrator-class>MIGRATOR\_CLASS\_NAME</migrator-class>

<target-idp-name>TARGET\_IDP\_NAME</target-idp-name>

</migrator>

<migrator>

<rule-class>RULE\_CLASS\_NAME</rule-class>

<migrator-class>MIGRATOR\_CLASS\_NAME</migrator-class>

<target-idp-name>TARGET\_IDP\_NAME</target-idp-name>

</migrator>

</migration-details>

</IDP>

</IDPS>

| **Tag Name** | **Description** |
| --- | --- |
| IDP\_NAME | Description: Name of IDP.  Default Value: N/A  Permissible Values: NA |
| AUTH\_MANAGER\_CLASS\_NAME | Description: Fully qualified name of authentication manger for any particular IDP.  Default Value: NA  Permissible Values: NA |
| DEFAULT\_IDP\_VALUE | Description: ‘true’ if any particular IDP is default one.  Default Value: false  Permissible Values:true/false |
| DISPLAY\_NAME | Description: Display name of IDP.  Default Value: None  Permissible Values: N/A |
| RULE\_CLASS\_NAME | Description: Fully qualified class name of rule class defined for any IDP. This is mainly for migration. Rule for migration will be specified in this particular class. The users who will satisfy this rule will be eligible for migration.  Default Value: None  Permissible Values: N/A |
| TARGET\_IDP\_NAME | Description: Actual IDP\_NAME of a Target idp for which migration rule and migrator classes are provided.  Default Value: It can take only IDP\_NAME attribute of any target IDP.  Permissible Values:It can take only IDP\_NAME attribute of any target IDP. |
| MIGRATOR\_CLASS\_NAME | Description: Fully qualified class name of migratory class. This is mainly for migration. Whenever user will satisfy rules specified in Rule class those users are eligible for migration. For migration of those users migratory class will be used.  Default Value: None  Permissible Values: N/A |

Example <idp-configuration> for LDAP and <user-attributes> if required is as follows: By default for CSM idp both should be empty.

<idp-configuration> <initContextFactory>com.sun.jndi.ldap.LdapCtxFactory</initContextFactory>

<securityAuthetication>simple</securityAuthetication>

<securityProtocol></securityProtocol>

<directory>ldap://10.88.26.32:10389/</directory>

<searchbase>ou=users,ou=system</searchbase>

<searchfilter>CN=Users</searchfilter>

<stringToReplace>Users</stringToReplace>

<bindUser>uid=admin,ou=system</bindUser>

<bindPassword>secret</bindPassword>

</idp-configuration>

<user-attributes>

<firstName>givenname</firstName>

<lastName>sn</lastName>

</user-attributes>

## caGrid Idp Configuration

1. Download the sync-description.xml file from the caGrid ivy to the deployment machine.
2. Download the certifiactes from the caGrid ivy to the deployment machine.
3. Refer the below links for downloading sync-description.xml file and certificates:
4. **For Training grid**: [http://software.cagrid.org/repository-1.3.0.2/caGrid/target\_grid/training-1.3/](http://software.cagrid.org/repository-1.3.0.2/caGrid/target_grid/training-1.3/" \t "_top)
5. **For Production grid**: <http://software.cagrid.org/repository-1.3.0.2/caGrid/target_grid/nci_prod-1.3/>
6. Follow the steps given in section [IDP Configuration](#_Idp_Configuration).
7. Modify the IDPAuthentication.xml located at caTISSUE\_HOME/catissuecore-properties folder. Update the tag

<SyncDescFilePath>File path forsyncdescription.xml﻿</SyncDescFilePath> with the location where sync-description.xml file has been copied.

1. ﻿﻿Create new folders .globus and certificates under /user/home. Copy the certificates into the /user/home/.globus/certificates folder.

Please refer the following example to update the IDPAuthentication.xml for Grid IDP:

1. The migrator class and migration rules class needs to be updated in CSM IDP tag.

<migration-details>

<migrator>

<rule-class>edu.wustl.migration.rules.CAGridMigrationRule</rule-class>

<migrator-class>edu.wustl.migrator.CAGridMigrator</migrator-class>

<target-idp-name>caGridTrainingIDP</target-idp-name>

</migrator>

<migrator>

<rule-class>edu.wustl.migration.rules.CAGridMigrationRule</rule-class>

<migrator-class>edu.wustl.migrator.CAGridMigrator</migrator-class>

<target-idp-name>caGridDorianIDP</target-idp-name>

</migrator>

</migration-details>

1. The IDP tag for Training Grid should be updated as follows:

<IDP name="caGridTrainingIDP">

<class name="edu.wustl.authmanager.CAGridAuthManager" />

<idpdetails>

<defaultidp>false</defaultidp>

<displayname>caGrid\_Training</displayname>

</idpdetails>

<idp-configuration>

<DorianServiceURL>https://dorian.training.cagrid.org:8443/wsrf/services/cagrid/Dorian</DorianServiceURL>

<securityAuthetication>simple</securityAuthetication>

<securityProtocol>ssl</securityProtocol>

<AuthenticationServiceURL>https://dorian.training.cagrid.org:8443/wsrf/services/cagrid/Dorian</AuthenticationServiceURL>

<SyncDescFilePath>C:/grid/caGrid/caGrid/repository/caGrid/target\_grid/training-1.3/sync-description.xml</SyncDescFilePath>

</idp-configuration>

<user-attributes>

</user-attributes>

<migration-details>

</migration-details>

</IDP>

1. IDP for Production Grid should be updated as follows:

<IDP name="caGridDorianIDP">

<class name="edu.wustl.authmanager.CAGridAuthManager" />

<idpdetails>

<defaultidp>false</defaultidp>

<displayname>caGrid\_Production</displayname>

</idpdetails>

<idp-configuration>

<DorianServiceURL>https://cagrid-dorian.nci.nih.gov:8443/wsrf/services/cagrid/Dorian</DorianServiceURL>

<securityAuthetication>simple</securityAuthetication>

<securityProtocol>ssl</securityProtocol>

<AuthenticationServiceURL>https://cagrid-dorian.nci.nih.gov:8443/wsrf/services/cagrid/Dorian</AuthenticationServiceURL>

<SyncDescFilePath>C:/grid/caGrid/caGrid/repository/caGrid/target\_grid/nci\_prod-1.3/sync-description.xml</SyncDescFilePath>

</idp-configuration>

<user-attributes>

</user-attributes>

<migration-details>

</migration-details>

</IDP>

## Idp Deployment

1. Go to the command prompt and change the directory to CATISSUE\_HOME/ folder.
2. Execute the command:
3. New Installation

ant install:jboss

**Note:**

This target will install a new JBoss server at the location defined by the property application.base.path.windows/application.base.path.linux specified in install.properties. If a server is already present at that location, a user promt will be displayed confirming to proceed and overrite the JBoss server with the new one. To proceed, please enter ‘Y’ and press Enter. In case user does not want to overrite the JBoss server, please enter ‘N’ and press Enter. In this case the deployment will be terminated

1. Upgrade

ant upgrade:jboss

# Appendix D - Redeploying the application after creating a Local Extension

When working with Local Extensions in caTissue, there are two uses cases:

1. A new Form might need to be created
2. An existing Form might need to be updated with additional fields.

In both the cases, the new/updated form will not be available for data entry until the application is redeployed. Below sections describe the steps to redeploy the application for both these cases.

## Creating a New Form

Following are the steps to redeploy caTissue application:

* 1. First shutdown the JBoss
  2. Execute following command
     1. New Installation

ant install:jboss

* + 1. Upgrade

ant upgrade:jboss

* 1. Start JBoss and access the application. The new form should be available for data entry.

## Updating an Existing Form

In case when you edit an exisiting form, there are two steps to follow.

1. Remove the caCORE JAR generated for the form: The old caCORE JAR for the particular form will have to be deleted.

* In case the name of the form is “**Participant\_Details**”. This belongs to a group called “**participant\_group**”.
* The JAR for the form will be created in the location specified by the property *cacore.deployable.location* in the caTissueInstall.properties file. The name would be “participant\_group.jar”.
* Delete this JAR file.

1. Redeploy the caTissue application as follows:
   * First shutdown the JBoss from
   * The run the following command
     + New Installation

ant install:jboss

* + - Upgrade

ant upgrade:jboss

* + Start the JBoss and access the application. The new form should be available for data entry.

# Appendix E - Verification of upgrade data

This section details the specific steps to be executed in order to compare the pre-upgrade and the upgraded (2.0 version) databases. This is provided as supplemental verification to help meet institutional policies. Please check with your institutions internal policies for documentation required for data verification of production systems.

<https://ncisvn.nci.nih.gov/svn/catissue_persistent/caTissue_Database_Comparison_Scripts/>

## Verification of MySQL Data

* 1. Checkout the MySQL files from the NCI SVN location

<https://ncisvn.nci.nih.gov/svn/catissue_persistent/caTissue_Database_Comparison_Scripts/>

This location will be referred as DBScriptHome.

* 1. Backup of the database which is to be upgraded and import it in a new database, say ‘db\_before\_upgrade’.
  2. Open MySQL command line client and login by providing the username and password for MySQL server.
  3. Select ‘mysql’ database by running the following command:
     1. use mysql;
  4. Execute the following commands in the given sequence:
     1. source DBScriptHome \MySQL\mysql\_datacomp.sql
     2. source DBScriptHome \MySQL\sf\_compDBTab\_W\_Diffcols.sql
     3. source DBScriptHome \MySQL\sf\_compTablesDataRows.sql
     4. source DBScriptHome \MySQL\sf\_compTablesInschemas.sql
     5. source DBScriptHome \MySQL\sf\_getcolsdiffstruct.sql
     6. source DBScriptHome \MySQL\sf\_getColumns.sql
     7. source DBScriptHome \MySQL\sp\_compareDataWithDiffStruc.sql
     8. source DBScriptHome \MySQL\sp\_CompTabData\_diffStruc.sql
     9. source DBScriptHome \MySQL\sp\_compTables.sql
  5. Upgrade the database as specified in the deployment guide. Let the database name be: ‘db\_after\_upgrade’.
  6. Run the following command for comparing the databases:  
          call mysql.sp\_compTables(db\_before\_upgrade,db\_after\_upgrade);
  7. The table ‘comptablesdatarows’ in ‘mysql’ database contains the comparison between the two schemas.

## Verification of Oracle Data

* 1. Checkout the Oralce file form the NCI SVN at <https://ncisvn.nci.nih.gov/svn/catissue_persistent/caTissue_Database_Comparison_Scripts/>.

This location will be referred as DBScriptHome.

* 1. Take the backup of the database which is to be upgraded and import it in a new database, say ‘db\_before\_upgrade’.
  2. Upgrade the database as specified in the deployment guide. Let the database name be: ‘db\_after\_upgrade’.
  3. Open SQL\*Plus and connect as sysdba.
  4. Execute following sql script –

@ DBScriptHome\Oracle\oraSPS.sql

(Note – If this sql script generates error due to cyclic dependency then please re-run the script.)

* 1. Edit oracComp.sql file to replace db\_before\_upgrade& db\_after\_upgrade with actual schema names.
  2. Execute the following commands:

@ DBScriptHome\Oracle\oracComp.sql

**Note:**In order to generate DB comparison report in excel/csv, please run the contents of oracComp.sql through SQL-Developer/TOAD. This will not display properly in a command line.