Contents

- 1 Introduction
  - 1.1 Overview of caTissue Suite
    - 1.1.1 Document Text Conventions
  - 1.2 Deployment Considerations
  - 1.3 Software Prerequisites
  - 1.4 Hardware Setup
- 2 Deploying the Web Application
  - 2.1 Describing Installation Prerequisites
  - 2.2 Downloading caTissue Suite
  - 2.3 Performing Pre Installation Configuration
  - 2.4 Creating Database Scripts
  - 2.5 Deploying caTissue Suite
  - 2.6 Performing Post Installation Configuration
    - 2.6.1 Configuring JBoss
    - 2.6.2 Configuring JBoss Server to deploy caTissue as HTTPS
  - 2.7 Configuring CSM
  - 2.8 Starting and Shutting down the Application Server
  - 2.9 Accessing the Web Application
  - 2.10 Deployment Errors
- 3 caTIES Integration in Suite
  - 3.1 Downloading MMTx Library and data.zip
  - 3.2 Deploying Gate Library
  - 3.3 Configuring caTIES
  - 3.4 Deploying caTIES
  - 3.5 Running caTIES pipeline
    - 3.5.1 Running Report Loader Server
    - 3.5.2 Running Report De-identification Server
    - 3.5.3 Running Report Concept Code Server
- 4 Testing the System
  - 4.1 Running the Test Case Suite (API)
    - 4.1.1 Running caCORE CSM Enabled Test Case Suite
    - 4.1.2 Getting Test Suite Result and Detailed Report
    - 4.1.3 Source code of Test Cases

- 4.2 Running Test Cases Suite (API) with HTTPS
    - 4.2.1 Configure Client for HTTPS
- 5 Private Public Data Store
    - 5.1 Deploying the Public Database
    - 5.2 Deploying the Web Application on the Public Database
- 6 Appendix
    - 6.1 MySQL Case Sensitivity Issue on Linux

Introduction

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

Overview of caTissue Suite

caTissue Suite is the next-generation tissue banking application which integrates the functionality of the existing caBIG? TBPT (Tissue bank and pathology Tools). 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:

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: Automates the process of coding, storing, and retrieving data from free-text pathology reports. The caTissue Suite is an integrated system that can seamlessly perform tasks across the three applications mentioned above. All the three applications have been independently developed and have their own UML(Unified modeling language) and data models. In addition, while caTissue Core and CAE are web based applications, caTIES is a Java thick client application. Thus, in the integrated solution, caTIES will be 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. Prerequisites	2. Pre-install	3. Install	4. Post Install
JDK 1.5 JBoss 4.0.0 MySQL 4.1 or Oracle 9i/10g Ant 1.6	Configure catissueInstall.properties. Configure the images and text on the home page. Create database and user.	Deploy or upgrade caTissue. Deploy caTIES Services.	Configure catissuecore_properties.xml. Configure JBoss to run on HTTPS. Start JBoss.

Following are some of the important considerations in deploying caTissue:

- Do you want to deploy the web application and database server on the same machine or different machines?
- If you wish to deploy caTIES services, then where do you want to deploy the caTIES services?

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

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

Software Prerequisites

The following table describes the software prerequisites for the caTissue Suite.

Table 1. Software Requirements

	Version	Type

Software Element Name		
Windows	2000 series/XP	Server and client
Mac	10.4.6	Client
Linux	RedHat 9 or RedHat Enterprise ES/AS 2.1 or higher	Server
JBoss	4.0.0	Application server
JDK	1.5	Java
Oracle	9i or 10g	Database ? Server and client **Note:** Oracle client to be installed on the machine which is hosting JBoss server.
		which is hosting JBoss server.
MySQL	4.1.10	Database
Internet Explorer	6.0, 7.0	Web browser
Mozilla Firefox	2.0.0.3	Web browser
Safari (Mac)	2.0.3	Web browser
caCORE	3.2	
CSM	3.2	Common Security Module
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/

Hardware Setup

Recommended server hardware requirements: CPU: 1Gz - 2Gz RAM: 2 GB - 4GB Hard disk: >20GB

Ideal hardware set up for caTissue installation: It is best to have the database and the JBoss application server hosted on different machines for optimal performance. Test and production servers: It is suggested to setup independent test and production environments. The test server can be used for running API test cases, training end users, user acceptance testing, and so forth.

Deploying the Web Application

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

- Describing 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

Describing Installation Prerequisites

The following table describes the installation prerequisites for the caTissue Suite.

Table 2. Software Prerequisites

Software Name	Version	URL
Java (JRE and JDK)	1.5	http://java.sun.com/javase/downloads/index_jdk5.jsp **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	4.0.0	http://labs.jboss.com/jbossas/downloads/
MySQL Database	4.1	http://dev.mysql.com/downloads/mysql/4.1.html
Oracle Database	9i or 10g	http://www.oracle.com
ANT	1.6	http://ant.apache.org/

Downloading caTissue Suite

Once you have downloaded the caTissue_Suite_Installable_v1.0.zip, extract the contents of the file to any directory. This folder will be referred to as CATISSUE_HOME henceforth in this document. 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 present in the installation folder and its description: Table 3. Installation Folder Files

File Name	Description
caTissueInstall.properties	Property file in which all the application configuration parameters are listed.
deploy.xml	ANT script used to deploy the application on a JBoss server. This also creates the database schema for Oracle or MySQL.

Sample-properties-service.xml	This file contains XML tags to be added in the properties-service.xml file. It provides startup parameters for the JBoss server. Note: This is a sample file, which may have to be changed in the JBoss configuration files based on the contents of this file.
Sample-log4j.xml	This file contains XML tags to be added in JBoss log4j.xml to configure the application message logger. Note: This is a sample file, which may have to be changed in the JBoss configuration files based on the contents of this file.
Sample-login-config.xml	This file contains XML tags to be added in JBoss login-config.xml for performing login module configuration. Note: This is a sample file, which may have to be changed in the JBoss configuration files based on the contents of this file.
MySQL_DB_Creation.sql	This file contains SQL statements that can be used to create a user and database for MySQL.
Oracle_DB_Creation.sql	This file contains SQL statements that can be used to create a user and database for Oracle.
caTissueSuite_caCORE_Client folder	This folder contains API files. For example: ClientDemo.java, ApiDemo.java, and build.xml for the client.

Note: Confirm that the environment variables ANT_HOME and JAVA_HOME are set, and that the system PATH includes the path for ANT_HOME/bin and JAVA_HOME/bin.

Performing Pre Installation Configuration

Before installing the application, all the parameters required for the process has to be defined by the user in the following files. File: caTissueInstall.properties Location: CATISSUE_HOME The following table explains the parameters in this file, along with its default and permissible values.

Table 4. Installation configuration parameters

Property Name	Description
jboss.home.dir	Description: This value should be set to JBOSS_HOME. **Note**: The path must be separated by ?/? and not ?\? Default Value:N/A Permissible Values: N/A For example: • In Windows: jboss.home.dir = c:/jboss-4.0.0 • In Linux:

	jboss.home.dir = usr/local/jboss-4.0.0
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: The port number on which JBoss server is running. Default Value: 8080 Permissible Values: N/A
database.type	Description: The database type used in the application. Default Value: N/A Permissible Values: : mysql, oracle
database.host	Description: The hostname or IP address of the machine on which the database server is running. Default Value: localhost Permissible Values: N/A
database.port	Description: The port number to connect with the database server. Default Value: N/A Default port for MySQL: 3306 Default port for Oracle: 1521 Permissible Values: N/A
oracle.tns.name	Description: This entry is required only if oracle database is being used. It is the entry in the tnsnames.ora file that points to the oracle database. Default Value: N/A 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.username	Description: The username used to connect to the database. Default Value: None Permissible Values: N/A
database.password	Description: The password used to authenticate the database user. Default Value: None 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: None Permissible Values: N/A
email.sendEmailFrom.emailAddress	Description: Email address used to send emails from the deployment script. Default Value: None Permissible Values: N/A
email.mailServer	Description: The mail server used to send emails. Default Value: None 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 Note: It is mandatory to configure this parameter to a valid numeric value.
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
caCORE.jBoss.home.dir	Description: The JBoss home directory specified in caCORE/conf/deploy.properties file for generating caCORE API for Dynamic Extensions (DE). Default Value: N/A Permissible Values: N/A
caCORE.project.name	Description: The project name specified in caCORE/conf/deploy.properties file for generating caCORE API for DE. Default Value: N/A Permissible Values: N/A

Note: Other parameters caties.mmtx.home, jboss.server.host and jboss.container.secure are required for deploying caTIES for loading the surgical pathological reports. These have been described in more detail in Section 3

Configuringe institute specific text and images Logo: The CATISSUE_HOME/images/ folder contains a file used to show the logo on the web pages. Replace the file InstitutionLogo.gif with the site specific image. Text: The CATISSUE_HOME/catissuecore_properties folder has a configuration folder, which contains the files used to specify contact information, privacy notice, and disclaimer. The following files should be updated as they relate to each adopting institution.

Table 5. Configuring Site Specific Text

Parameter Name	**Details**
PrivacyNotice.txt	Privacy notice of the application.
Accessibility.txt	Accessibility notice of the application.
Disclaimer.txt	Disclaimer for the application.

ContactUs.txt	Contact information for system help.

Creating Database Scripts

Before deployment of the caTissue Suite application you need to create a database and a user. Note: Skip this step if you are upgrading from caTissue Core v1.2.0.1 to caTissue Suite as there is no need for new database creation for upgrade. In this case there will be separate step to upgrade the database and application which is explained in next section. The sample scripts for creation of database and users for Oracle and MySQL are available in the CATISSUE_HOME folder. You can also 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. 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 6. 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 user through your own script in 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 the flush privileges command 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 7. 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

Deploying caTissue Suite

Once the pre-installation set up is ready, you are ready to deploy the application. The following section provides different approached 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. To deploy caTissue Suite:

1. Go to the command prompt and change the directory to CATISSUE_HOME/ folder.
2. To install fresh instance of caTissue Suite, execute the command : ant -f deploy.xml deploy_all
3. To upgrade from caTissue Core v1.2 or v1.2.0.1, execute the command: ant -f deploy.xml upgrade_all

CAUTION: Take a database backup of caTissue Core database before upgrading. After the successful deployment of the application, a test mail will be sent to the administrative email address specified in the caTissueInstall.properties file. In case of any deployment errors, refer to the section Deployment Errors for details.

Performing Post Installation Configuration

The post installation configuration consists of:

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

Configuring JBoss

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

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

- File: 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.

```
1.<appender name="CATISSUECORE_LOGGER" class="org.apache.log4j.RollingFileAppen
           <param name="File"
value="${jboss.server.home.dir}/log/catissuecore.log" />
            <param name="Append" value="true" />
            <param name="MaxFileSize" value="10000KB"/>
      <param name="MaxBackupIndex" value="10"/>
      <layout class="org.apache.log4j.PatternLayout">
            <param name="ConversionPattern" value="%d{dd MMM yyy HH:mm:ss}, %-6
      </layout>
   </appender>
<category name="catissuecore.logger">
            <priority value="ERROR" />
            <appender-ref ref="CATISSUECORE_LOGGER" />
   </category>

2.<appender name="FILE"  class="org.jboss.logging.appender.DailyRollingFileAppe
      <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"/>
```

Note: Log4j is an open source application for logging application messages. Please refer to [1] for more detail about loggers. 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"  class="org.jboss.logging.appender.DailyRollingFileAppen
```

```
<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"/>
```

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"
```

2. 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.

- File: caTissueCore_Properties.xml

Location: JBOSS_HOME/server/<jboss.server.name>/catissuecore-properties/ This file contai ns the configuration properties to run caTissue. The following table describes all the properties, their default values, and their permissible values.

Table 8. 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
Institution.url	Description: The URL of the institution?s website 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
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 drop down 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: The default 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

Configuring JBoss Server to deploy caTissue as HTTPS

IMPORTANT: Since caTissue contains patient identified information, it is important to deploy it in a secure environment. For more information read http://en.wikipedia.org/wiki/Https. Even if the application is installed within secure firewall, it is highly recommended to deploy the application as HTTPS to avoid any unauthorized access. To deploy caTissue as a HTTPS based secure web application, you need to perform the following steps:

1. Create a self-signed certificate.
2. Move the self-signed 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: Creating a Self-Signed Certificate The Java Developer's Kit includes a utility to create certificates. To use it, go to the command prompt and type the following:

```
keytool -genkey -alias tomcat -keyalg RSA
```

You will be prompted to enter all the information required for the certificate as shown next:

Step 2: Moving the Keystore File

1. In the example described in Step 1, you will notice that the keystore file is created in the directory from where the command was executed, such as C:\Documents and Settings\mcgradyt\.keystore.
2. Rename this file to chap8.keystore.
3. Copy this keystore file to the conf/ directory of your JBoss installation. For example:

JBOSS_HOME/server/<jboss.server.name>/conf.

Step 3: Editing the JBoss Configuration File

1. Open file JBOSS_HOME/server/<jboss.server.name>\deploy\jbossweb-tomcat55.sar\server.xml.
2. Inside this file, there should be a tag as in the following:

```
<!-- SSL/TLS Connector configuration using the admin devl guide keystore
 <Connector port="8443" address="${jboss.bind.address}"
   maxThreads="100" strategy="ms" maxHttpHeaderSize="8192"
   emptySessionPath="true"1
   scheme="https" secure="true" clientAuth="false"
   keystoreFile="${jboss.server.home.dir}/conf/chap8.keystore"
   keystorePass="rmi+ssl" sslProtocol = "TLS" />
 -->
```

3. In step 2 block above, perform the following changes:

1. Uncomment the block if it is commented.
2. Change the port to 443 or to any desired port on which you want to run the application on.
3. Change the keystore password in parameter keystorePass to the password specified while creating the certificate.</nowiki>

4. After the changes, the block would look like this: <!-- SSL/TLS Connector configuration using the admin devl guide keystore --> <Connector port="443" address="${jboss.bind.address}" maxThreads="100" strategy="ms" maxHttpHeaderSize="8192" emptySessionPath="true" scheme="https" secure="true" clientAuth="false" keystoreFile="${jboss.server.home.dir}/conf/chap8.keystore" keystorePass="changeit" sslProtocol = "TLS" /> 5. Comment the section where the HTTP connector tag is defined so that the application is not accessible through HTTP. 6. 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 deploy_all or the deploy_app 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">
   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>
```

Table 9. 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/Appli
APP_PROPERTIES_PATH	Description: Name of the property specifying the application configuration. Value:JBOSS_HOME/server/<jboss.server.name>/catissuecore-properties/caTis

Note:

- gov.nih.nci.security.configFile is the name of the property file, which points to the fully qualified path where ApplicationSecurityConfig.xml has 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.RDBMS
       <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=? an
 <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 10. 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 Shutting down the Application Server

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

For Windows: Open the file JBOSS_HOME/bin/run.bat and search for the line set JAVA_OPTS and modify it to appear like: set JAVA_OPTS=%JAVA_OPTS% -Xms512m -Xmx1024m -XX:PermSize=64m -XX:MaxPermSize=256m

For Linux and Unix: Open file /usr/local/jboss/bin/run.conf and search for the line starting with JAVA_OPTS and modify it to appear like:

set JAVA_OPTS="-server -Xms128m -Xmx1024m -XX:PermSize=64m -XX:MaxPermSize=256m

Note: If you face ?java.lang.OutOfMemoryError: Java heap space? error during the use of the application, increase the ?Xmx parameter in the above command by 128m.

Now, start the JBoss server as per the instructions below

Table 11. Server Startup and Shutdown

Operation	Windows	Linux and Unix
Start	JBOSS_HOME/bin/run.bat	nohup $JBOSS_HOME/bin/run.sh
Stop	JBOSS_HOME/bin/shutdown.bat	JBOSS_HOME/bin/shutdown.sh

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 and first.admin.password of 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: The ANT_HOME is not set or set incorrectly.

Solution: Set the ANT_HOME environment variable and retry the deployment.

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

Cause: The 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 access http://machine_name:port_number/catissuecore

Cause: Check if the jboss.home parameter in caTissueInstall.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 caTissueInstall.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 application 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: System 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: You need to ensure that the mysql.user table has File_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.

caTIES 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 12. 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, user has the option to select one of the three de-identifier options described in the Table 13.
Concept coder	This service concept codes the de-identified SPRs with the concepts from the NCI Metathesaurus.

Table 13. De-identifier options

DeID de-identifier	DeID de-identifier takes help of the DeID de-identification tool to de-identfy SPRs. The DeID tool is available only for windows platform, therefore this

	de-identifier can run only on windows platform. DeID tool must be installed on the machine prior to deploying DeID de-identifier.
Harvard Scrubber de-identifier	User may choose this option to use Harvard Scrubber as the de-identification tool. Harvard Scrubber de-identification tool is an open source tool and is bundled with the de-identifier hence not required to install on machine prior to deploying de-identifier service.
Do-Nothing de-identifier	By using this option, user might choose not to de-identify the reports.
Custom de-identifier	This option is available for the 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 14. 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 the 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 the 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 concept coder services (that is, 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 as MMTX_ROOT henceforth.
3. Unzip ncimeta-0601.zip and extract to the MMTX_ROOT\nls\mmtx\data\ folder.
4. Update the mmtxRegistry.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 two parameters in CATISSUE_HOME/caTissueInstall.properties.

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. Default Value: None Permissible Values: None
jboss.server.host	Description: The hostname or IP address of the machine on which caTissue server is deployed. Note: This parameter should not be set to localhost. Set the actual hostname or IP address. Default Value: None Permissible Values: None
jboss.container.secure	Description: Should be set to yes if JBoss server is deployed as HTTPS. Default Value: None Permissible Values: yes/no

1. Run ant -f deploy.xml deploy_gate.

Configuring caTIES

Configure the following parameters in the CATISSUE_HOME/deploycaties.properties file.

Table 15. 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 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 report loader server. Default Value: None Permissible Values: yes, no
report.loader.installation.dir	Description: If you opt to install 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 report 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
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 report concept code server then this should be used to specify the location where 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 report de-identification server. Default Value: None Permissible Values: yes, no
deididentifier.installation.dir	Description: If you opt to install report de-identification server, then this property should be used to specify the location where 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 d-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 Harvard Scrubber 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/caTIES_conf/sites_configuration.xml

Note: The properties with characters beginning and ending with @@ will be automatically replaced with the corresponding values from the CATISSUE_HOME/caTissueInstall.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/caTissueInstall.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_ABBRIVI
        </site>
 </sites>
```

Deploying caTIES

To deploy caTIES:

1. From the command prompt, go to the CATISSUE_HOME/ folder.
2. Run the ANT task as follows: ant -f deploycaties.xml 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 caT replaced with the value specified in caTissueInstall.properties for the proper Default Value: @@ADMIN_EMAIL@@
password	Description: Password of the user whose user name is specified in property u replaced with the value specified in caTissueInstall.properties file for the pro Default Value: @@ADMIN_PASSWORD@@ Permissible Values: None Note: If you are a new user of caTissue web application, you must change th password here
Report Loader Server Settings	
inputDir	Description: Path of the directory from which the report loader will pick up providing input to the caTIES pipeline. The token will be automatically repl 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 for automatically replaced with the value specified in deploycaties.properties fo Default Value: @@BAD_FILES_DIR@@ Permissible Values: N/A

processFileDir	Description: Path of the directory where successfully processed HL7 format automatically replaced with the value specified in deploycaties.properties fo Default Value: @@PROCESS_FILES_DIR@@ Permissible Values: N/A
filePollerSleepTime	Description: Sleep time in millisecond of report loader thread and report que 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 inform 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 re will be automatically replaced with the value specified in deploycaties.prope 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 automaticall Default Value: @@GATE_HOME@@ Permissible Values: N/A
caties.creole.url.name	Description: Path to the creole.xml, which specifies values of properties requ Default Value: http://@@HOST@@:@@PORT@@/gate/gate_3_1/applica

	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/caT 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/caT Permissible Values: N/A
caties.section.chunker.url.name	Description: Path to the section chunker. Default Value: http://@@HOST@@:@@PORT@@/gate/gate_3_1/applica 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/caT 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/applica 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/caT Permissible Values: N/A
conceptCoderSleepTime	Description: Sleep time in millisecond of report concept code pipe line mana 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, the report text it creates Binar The Binary output contains the information related to concepts appearing in output is very large, you can discard them 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 XML XML output contains the information related to concepts appearing in the re is very large, you can discard them using this property.
Report De-identification Server Settings	
deidentfierClassName	Description: Since there are different options for the de-identification tool pr de-identification server by this property. It is the job of given class which sh selected tool. The token is automatically replaced by respective class name a be automatically replaced with the value specified in deploycaties.properties 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 i 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 m Default Value: 86400000 Permissible Values: any positive value <= Long.MAX
deidHome	Description: Path of the home directory where DeID software is installed. T 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 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 na 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 method call. This property is available if and only if the user has opted to ins Default Value: Dataset.dtd Permissible Values: N/A
harvardScrubberDTDFileName	Description: Data type definition file name required to extract text from the Scrubber de-identification tool. This property is available if and only if the u de-identification server. Default Value: <DEIDENTFIER_SERVER_INSTALLATION_DIR>/Scrub Permissible Values: N/A
harvardScrubberConfigFileName	Description: Configuration file name required for report de-identification. T user has opted to install Harvard Scrubber de-identification server. Default Value: <DEIDENTFIER_SERVER_INSTALLATION_DIR>/caTI Permissible Values: N/A

Running Report Loader Server

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 Report (SPR), report loader server also handles cases like participant conflict, SCG conflict, and so on.

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 report loader server, run the following ANT task at the command prompt:

```
ant stop_report_loader_server
```

Running Report De-identification Server

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 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. Pre-requisites to use deid deidentifier are:

A. De-identification software must be pre-installed. The De-ID software is downloadable from http://cliniscience.grouphub.com/ site. Currently the software is supported only on Windows OS, therefore the de-identification server supports Windows OS only.
B. Installer 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.

2. The Harvard Scrubber deidentifier: In this the de-identification server takes help of a Harvard Scrubber, an open source tool for report text de-identification.

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

4. Custom deidentifier: If user has written his own program to de-identify report text using then he can use it with the de-

A.The class implementing custom de-identification must extend the class caTissue Suite class edu.wust.catissuecore.deidentifier.AbstractDeidentifier and should implement all the abstract methods specified in the above class.
B. Create a jar of source code and copy it to the <DEIDENTIFIER_INSTALLATION_DIR>/lib directory.
C. Verify the deidentifierClassName property in the <DEIDENTIFIER_INSTALLATION_DIR>/caTIES_conf/caTIES.properties file.
D. Copy required resources to <DEIDENTIFIER_INSTALLATION_DIR> directory.

Where, <DEIDENTIFIER_INSTALLATION_DIR> is the path specified by the deid.installation.dir in CATISSUE_HOME/caTIES_conf/caTIES.properties file. To run report de-identification server perform following steps:

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

```
ant run_deidentifier_server
```

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

ant stop_deidentifier_server

Running Report Concept Code Server

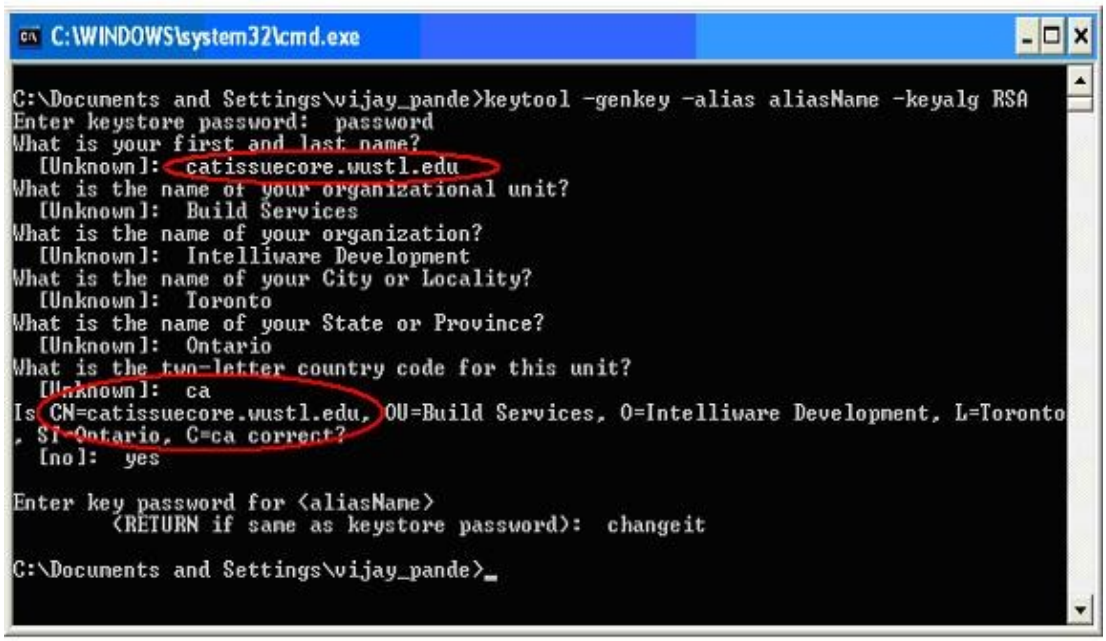
Concept coder server is responsible for concept coding report text. Concept coding is the process of medically related lexical classification of words that appear in the report text.

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/creole.xml where JBOSS_HOST and JBOSS_PORT are the hostname and port number respectively on the JBoss server.

2. If JBoss is running on HTTPS then please make sure that keystore file created is having Common Name (CN) should be same as the machine name on which JBoss is running. For more information please refer topic Creating self signed certificate explained at Configuring JBoss Server to deploy caTissue as HTTPS.

For e.g. keystore file for concept coder pointing to demo site can be created as shown below:



3. Switch to the folder where report concept coder server is deployed (specified by concept.code.installation.dir in caTIES_conf/caTIES.properties file).
4. Run the following ANT task at the command prompt:

ant run_concept_code_server

5. 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.emailAddress and first.admin.password.

Running the Test Case Suite (API)

This section describes the process to configure and run the test suite of CSM enabled caCORE client to use caTissue Suite application using caCORE API

The CATISSUE_HOME/caTissueSuite_Client/ directory contains the files and libraries required to compile and run the caCORE CSM enabled client program.

Before using caCORE API, configure the host property in remoteService.xml file present in CATISSUE_HOME/caTissueSuite_Client/conf

File: remoteService.xml

Table 16. 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:8080 to 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 caTissueSuite_Client.
2. Edit the demo client program ClientDemo.java and set the appropriate username and password in the startSession(username,password) method.
3. Run ANT task at the command prompt to compile and run the client program. The syntax of ANT task is: ant <target_name>.

Note: The API test cases have to be executed on a fresh deployment with empty database. DONOT run these test cases on a production database as it inserts test data in to the database. Execute these test cases on a different test server and database.

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

Table 17. 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: In caTissue API while adding test cases user needs to set Logger inside 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_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_HOME/caTissueSuite_Client/src/edu/wustl/catissuecore/bizlogic/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 Configuring JBoss Server to deploy caTissue as HTTPS.

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 as HTTPS.
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");
```

3. In the remoteService.xml file, set the value for the property serviceURL to

https://@@HOST@@:@@PORT/catissuecore/http/remoteService where @@HOST:@@PORT should be replaced with host URL as shown
in Configure remoteService.xml.

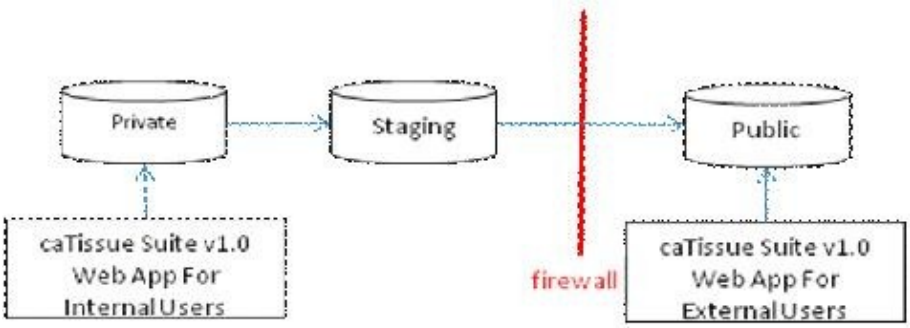
To run the APIs for Suite on HTTPS, refer to "Running the Test Case Suite (API)".

Private Public Data Store

caTissue Suite contains patient/participant identifying data. Since the database containing identified data always resides in the institutional IT firewall, it poses challenges 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 where caTissue database is de-identified and copied into a database, which is publicly accessible. Now this public database can be made freely accessible to authorized researchers worldwide.

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 v1.0 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 or using the caTissue Suite API.
3. The public database should not contain any data. It will be overwritten the next time when the public database is coordinated with the private database. Synchronized
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. Generate the commands.properties file.
3. 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/public_private_migrator/privatePublic.properties

Table 18. Private Public Datastore Configuration Parameter

Property Name	Description
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
privateDBType	Description: Database type of private caTissue Suite instance. Default Value: N/A Permissible Values: N/A
privateDBAUserName	Description: Administrator user name of database of private caTissue Suite instance. This is required only for Oracle. Default Value: N/A Permissible Values: N/A
privateDBAPassword	Description: Administrator password of database 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
stagingDBHost	Description: Database host name of staging 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
stagingDBType	Description: Database type of staging database. Default Value: N/A Permissible Values: N/A
stagingDBPort	Description: Database port of staging database. Default Value: N/A Permissible Values: N/A
publicDBName	Description: Database name of public caTissue Suite instance. 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
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
publicDBType	Description: Database type of public caTissue Suite instance. Default Value: N/A Permissible Values: N/A
publicDBAUserName	Description: Administrator user name of database of public caTissue Suite instance. This is required only for Oracle.

	Default Value: N/A Permissible Values: N/A
publiceDBAPassword	Description: Administrator password of database of public caTissue Suite instance (required only for Oracle). Default Value: N/A Permissible Values: N/A
stagingDBPassword	Description: Database password of staging database. Default Value: N/A Permissible Values: N/A
stagingDBType	Description: Database type of staging database. Default Value: N/A Permissible Values: N/A
stagingDBPort	Description: Database port of staging database. Default Value: N/A Permissible Values: N/A
publicDBName	Description: Database name of public caTissue Suite instance. 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
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
publicDBType	Description: Database type of public caTissue Suite instance.

	Default Value: N/A Permissible Values: N/A
publicDBAUserName	Description: Administrator user name of database of public caTissue Suite instance. This is required only for Oracle. Default Value: N/A Permissible Values: N/A
publiceDBAPassword	Description: Administrator password of database of public caTissue Suite instance (required only for Oracle). Default Value: N/A Permissible Values: N/A

Generating the command.properties File To generate the command.properties file:

- Switch to the CATISSUE_HOME/public_private_migrator directory and run the following ANT task:

ant -f privatePublic.xml generate_command_file

Running the migration tool To run the migration tool, run the following command: ant -f privatePublic.xml migrate

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 job or Windows Scheduler to execute the migrate command.

CAUTION: Every time migrate command is executed, the public database is recreated, and any updates to the public database are lost. The synchronization tool does not take a backup of the public database. Optionally, you can also add a database backup creation in the scheduled task.

Deploying the Web Application on the Public Database

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

Appendix

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 for more details on this issue.

To set the system variable on Linux:

If the file my.cnf is available in installed directory of MySQL, 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</nowiki>
NOTE: After setting lower_case_table_names to 1 on Linux, you need to recreate
```