CAGWAS

Configuration Guide

National Cancer Institute

Center for Bioinformatics

December 11, 2007

TABLE OF CONTENTS

ABOUT THIS GUIDE

This section introduces you to the *caGWAS Configuration Guide.* It includes the following topics:

- *Purpose* on this page
- *Audience* on this page
- *Topics Covered* on page 2
- *Additional References* on page 2
- *Text Conventions Used* on page 2

Purpose

This guide explains how to customize caGWAS for your study. It explains the database, the caIntegrator-spec module, and the GWAS application. The guide includes procedures for obtaining, installing, configuring, and managing each component of caGWAS.

Audience

This configuration guide is best suited for an experienced Java developer who is familiar with the following J2EE technologies:

- Unix/Linux environment (configuring environment variables; Installing Ant, JDK, and JBOSS server)
- Ant build scripts
- J2EE web application development using the Struts framework, Servlet/JSP's, javascript, AJAX, and XML/XSLT.
- J2EE middle-ware technologies such as n-tier service oriented architecture and software design patterns.

In addition, to properly configure the database (Oracle 9i or MySQL), you will need assistance/access from a database administrator.

Topics Covered

This brief overview explains what you will find in each chapter and appendix.

- *Chapter 1, Understanding caGWAS Architecture,* on page 5 introduces caGWAS and what you must do before you begin using it.

- *Chapter 2, Working with caGWAS,* on page 13 provides procedures for installing caGWAS.

- *Appendix A, Third-Party Tools,* on page 23 provides a list of the third-party `jar` files that caGWAS uses.

Additional References

For more information about caGWAS, see the following references:

- [caGWAS Technical Guide](#)

Text Conventions Used

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

Convention	Description	Example
Bold & Capitalized Command **Capitalized command > Capitalized command**	Indicates a Menu command Indicates Sequential Menu commands	**Admin > Refresh**
TEXT IN SMALL CAPS	Keyboard key that you press	Press ENTER
TEXT IN SMALL CAPS + TEXT IN SMALL CAPS	Keyboard keys that you press simultaneously	Press SHIFT + CTRL and then release both.
`Monospace type`	Used for filenames, directory names, commands, file listings, and anything that would appear in a Java program, such as methods, variables, and classes.	`URL_definition ::= url_string`
Icon	A toolbar button that you click	Click the **Paste** button () to paste the copied text.
Boldface type	Options that you select in dialog boxes or drop-down menus. Buttons or icons that you click.	In the Open dialog box, select the file and click the **Open** button.
Italics	Used to reference other documents, sections, figures, and tables.	*caCORE Software Development Kit 1.0 Programmer's Guide*

Convention	Description	Example
Italic boldface monospace type	Text that you type	In the New Subset text box, enter `Proprietary Proteins.`
Note:	Highlights a concept of particular interest	**Note:** This concept is used throughout the installation manual.
Warning!	Highlights information of which you should be particularly aware.	**Warning!** Deleting an object will permanently delete it from the database.
{}	Curly brackets are used for replaceable items.	Replace `{root directory}` with its proper value, such as `c:\cagwas`

Credits and Resources

The following people contributed to the development of this document.

caGWAS Development and Management Teams		
Development	*Project and Product Management*	*Documentation and Quality Assurance*
Himanso Sahni [3]	Subha Madhavan [1]	*Documentation*
Michael Holck [4]	Ryan Landy [3]	Carolyn Kelley Klinger [2]
Dana Zhang [3]		*Quality Assurance*
Alex Jiang [3]		Leonie Misquitta[5]
		Nonna Rabinovich[5]
1 National Cancer Institute Center for Biomedical Informatics and Information Technology (NCI CBIIT)	2 Lockheed Martin	3 Science Application International Corporation (SAIC)
4 ScenPro	5 NARTech	

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

UNDERSTANDING CAGWAS ARCHITECTURE

This chapter includes the following sections:

Introduction to caIntegrator

"Informatics tools to integrate clinical trials and high throughput molecular analysis"

The caIntegrator knowledge framework provides researchers with the ability to perform ad hoc querying and reporting across multiple domains. This application framework comprises an n-tier service oriented architecture that allows pluggable web-based graphical user interfaces, a business object layer, server components that process the queries and result sets, a data access layer and a robust data warehouse.

caIntegrator is envisioned to be the foundation for a number of translational applications. One such reference implementation at NCICB is called CGEMS (Cancer Genetics Markers of Susceptibility) – http://caintegrator.nci.nih.gov/cgems. This knowledge framework offers a paradigm for rapid sharing of information and accelerates the process of analyzing results from various biomedical studies with the ultimate goal to rapidly change routine patient care.

More information is available at the caIntegrator website: http://caintegrator-info.nci.nih.gov.

Introduction to caGWAS

Cancer Genome-Wide Association Studies (caGWAS) allows researchers to integrate, query, report, and analyze significant associations between genetic variations and disease, drug response or other clinical outcomes. New breakthroughs in SNP array technologies make it possible to genotype hundreds of thousands of single nucleotide polymorphisms (SNPs) simultaneously, enabling whole genome association studies. Within the Clinical Genomic Object Model (CGOM), the caIntegrator team created a domain model for Whole Genome Association Study Analysis. CGOM-caGWAS is a semantically annotated domain model that captures associations between Study, Study Participant, Disease, SNP Association Analysis, SNP Population Frequency and SNP annotations.

Following the principals of caBIG, caGWAS APIs and web portal provide:

- A semantically annotated domain model, database schema with sample data, seasoned middleware, APIs, and web portal for GWAS data;

- platform and disease agnostic CGOM-caGWAS model and associated APIs;

- the opportunity for developers to customize the look and feel of their GWAS portal;

- a foundation of open source technologies;

- a well-tested and performance-enhanced platform, as the same software is being used to house the CGEMS data portal at https://caintegator.nci.nih.gov/cgems;

- accelerated analysis of results from various biomedical studies; and

- a single application through which researchers and bioinformaticians can access and analyze clinical and experimental data from a variety of data types, as caGWAS objects are part of the CGOM, which includes microarray, genomic, immunohistochemistry, imaging, and clinical data.

Prerequisites

The general requirements common to multiple components of caIntegrator are a prerequisite, and must be installed on your machine prior to attempting to work with the caIntegrator framework. The following are open source technologies which power caIntegrator:

- Java Software Development Kit (JDK) version 1.5.0_06 (http://java.sun.com/j2se/1.5.0/download.jsp)

- JBoss Container (JBoss version 4.0.4 recommended) (http://labs.jboss.com/jbossas/downloads/)

- Jakarta Ant version 1.7.0 (http://archive.apache.org/dist/ant/binaries/)

- Oracle 9i (http://www.oracle.com)

- MySQL Enterprise Edition (http://dev.mysql.com/downloads/)

Please acquire each of these and follow the installation instructions provided with each respective product for your environment.

Setting Up the Environment

Before you begin using caGWAS, you must set the appropriate path and variables for your environment.

In a Windows environment, refer to the following illustration for the path and environment variables you should use:

In a Linux environment, refer to the following illustration for the path and environment variables you should use:

General Hardware Requirements

caIntegrator WGS has been tested with the following configuration, on two separate machines.
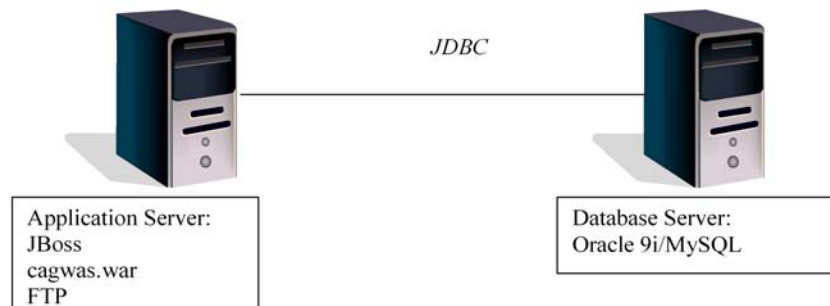
Application Server

Red Hat EL 3.0 on HP Proliant DL360 G4, with dual 3.40 GHz Xeon 64T CPUs, 8 GB RAM, mirrored 144 GB SCSI hard drives.

Database Server

Sun Solaris 8.0 on Sun v1280, with four 900 MHz US-III CPUs, 8 GB RAM, mirrored local and SAN disks.

Server Configuration

The following image depicts the recommended server configuration for caGWAS.

Obtaining the caIntegrator Source

The caIntegrator source bundle will be distributed through the caIntegrator gForge website at http://gforge.nci.nih.gov/projects/caintegrator/. All distributions will be in the form of zip archives and will require a zip utility such as WinZip, the Java Jar utility or 7-Zip to open. Begin by selecting the bundled download named `caIntegrator_wgs1.2_src_bundle.zip`.

Source Distribution

The source code distribution below contains the caIntegrator source and its reference implementation caGWAS.

Save this archive, `caIntegrator_wgs1.2_src_bundle.zip`, to your machine and extract it into a local directory.

After this extraction, you will see four main components:

- `caintegrator-analysis-commons.zip`
- `caintegrator-application-commons.zip`
- `caintegrator-spec.zip`
- `cagwas.zip`

Each of these archives will be referenced in the subsequent portions of this guide. Continue by downloading the database initialization files included in `wgs_db.zip`. This archive will contain the required files for properly setting up the database for the caGWAS reference implementation of caIntegrator.

Database Download

The file below contains all necessary instructions and information for importing the database file for the WGS reference implementation.

- `wgs_db.zip`

Database Technology

caIntegrator employs a basic star schema with modification for the study data warehouse design that supports the integration of clinical and genomic data. It is a generic, query-optimized schema that contains fact tables such as `GENOTYPE_FACT` and `SNP_FREQUENCY_FACT`, etc. Look up entities such as `GENE_DIM`, `SPECIMEN`, and `STUDY_POPULATION` tables make up the dimensions in the schema. This schema provides a highly de-normalized view of the data and a data neutral framework from which queries can be executed with quick retrieval time.

You can configure an Oracle or a MySQL database for caGWAS. Note that unlike MySQL, which is an open-source database management system, Oracle is a licensed application that creates proprietary data. For more information, see:

- *Using an Oracle Database* on page 9
- *Using a MySQL Database* on page 10

Using an Oracle Database

Assumptions and Requirements

- Oracle 9i has been set up and you have access to the administrator account.
- You have proper privileges to create a named tablespace and you are familiar with the Oracle DB technology.

In addition to the above assumptions, it is also required that you have properly downloaded `wgs_db.zip` from the caIntegrator GForge site as explained in *Source Distribution* on page 8.

Initializing the Oracle Database

To initialize the database, perform the following steps:

1. Unzip `wgs_db.zip` to a directory on your local machine such as `c:\caintegrator\db`.
2. Login into Oracle as database the administrator.
3. Create a named tablespace with at least 8 GB.
4. Create a user with this tablespace as default tablespace.

At this point, you have two options: import a dump to set up the database with the distributed data, or run the `DDL` script, which will set up the database structure, but not import any data.

Initialization Option 1: Import a dump file

1. Create database by importing the `DMP` file from the download site.
2. Locate the file `wgs.dmp` from the extracted location (labeled below as `<location>`).
3. Issue the following command substituting the <variables> with your settings:

```
imp <username>/<password>@<database>

file=<location>/wgs.dmp log= <name.log> full=y
```

Option 1 Example

1. Extract `wgs_db.zip` file to your local C drive. Verify that the extracted file `c:\wgs.dmp` exists.
2. Go to the command line to login to the database named `wgsdb` as the user named `wgs` with password `wgspass`, which the database administrator created for this import. Then issue the following command:

```
Prompt>imp wgs/wgspass@wgsdb file=c:\wgs.dmp log=wgs.log
full=y
```

3. Press ENTER.

Note: You or the DBA can verify this database import by checking the log file for errors.

Initialization Option 2: Run a `DDL` script

1. Create a database by running the DDL script from the download site.

2. Locate the file `wgs.sql` from the extracted location (labeled below as <location>).

3. Log into the database as created named user.

4. Issue the following command:

    `@<location>/wgs.sql`

Option 2 Example

1. Extract `wgs_db.zip` to your local C drive. Verify that the extracted file `c:\wgs.sql` exists.

2. Log into the database named `wgsdb` as the user named `wgs` with password `wgspass`, through SQL Plus, then issue the following SQL statement:

    `@c:\wgs.sql`

3. Press ENTER.

Verifying the Oracle Database

Once the database has been created through importing (option1) or the database has been created through running the DDL script (option 2) without error, it can be verified by comparing the number of the objects in current database with the number of the objects from the original database. This verification process can be performed through a SQL query in the SQL Plus or TOAD environment. For both options, the number of objects should match the list below, which was taken from the original database.

Tables	25
Views	4
Indexes	61

Table 1.1 Number of objects of each type in the caGWAS database

In addition, most of the tables in the option 1 database should contain seed data for the public to view; no data is available in the option 2 database.

Using a MySQL Database

Assumptions and Requirements

Before using a MySQL database with caGWAS, it is assumed that:

*   MySQL has been set up and you are familiar with MySQL database technology.

*   You have access to the administrator account or to a user that has been granted proper privileges to create a database (schema).

It is also required that you have downloaded `mysql_cgemstemp.dmp` from the caIntegrator GForge site as explained in *Source Distribution* on page 8.

Initializing the MySQL Database

You can import `mysql_cgemstemp.dmp` using either of the methods described below.

- Create the database by importing from the database server, or
- Create the database by importing the file while logged into the database from the mysql client.

Option 1: Import from the database server

- Run the following command on the database server:

  `MYSQL_DIRECTORY` - references where MySQL was installed

  `CONF_DIRECTORY` - references the location of where the `my.cnf` file is located

  ```
  <MYSQL_DIRECTORY>/bin/mysql --defaults-
  file=<CONF_DIRECTORY>/my.cnf < mysql_cgemstemp.dmp
  ```

Option 2: Import the file through the mysql client

1. Log into the MySQL instance as a user with the proper privileges:

   ```
   mysql -h <hostname> -u <user> -P<port> -p
   ```

   For example:

   ```
   mysql -h myserver.domain -u cgemsuser -P3630 -p
   ```

2. At the `mysql` prompt, run the following:

   ```
   mysql> source mysql_cgemstemp.dmp
   ```

Verifying the MySQL Database

Once the database has been created by importing from the database server (option1 above) or from your client (option 2 above) without error, you can verify it by comparing the number of the objects in the current database with the number of the objects from the original database. This verification process can be performed through a SQL query. For both options, the number of objects should match the list below, which was taken from the original database.

```
select count(*) from information_schema.tables

where table_schema='cgemstemp';
```

Tables: 31

```
select count(*) from information_schema.views

where table_schema='cgemstemp';
```

Views: 0

```
select count(distinct table_name, index_name) from
information_schema.statistics

where index_schema='cgemstemp';
```

Indexes: 89

WORKING WITH CAGWAS

This chapter includes the following sections:

- *Assumptions and Requirements* on this page
- *Installing caGWAS* on page 14
- *Configuring caGWAS* on page 15
- *Deploying caGWAS* on page 18
- *Verifying Installation* on page 21

Assumptions and Requirements

The database component listed in this guide must be installed, configured, and deployed prior to installing this component.

There is currently a 2GB RAM minimum requirement for any server that this application is deployed on. This is due to the extremely large data sets that are being processed and returned to the user.

The following components are required and come as part of the bundled download.

Software	*Version*	*Source included in the bundle*
`caintegrator-application-commons`	1.2	Yes
`caintegrator-analysis-commons`	1.2	Yes
`caintegrator-spec`	1.2	Yes
`cagwas`	1.4	Yes
`weka.jar`	3-4-*	No

Table 2.1 Application software distributed as part of caGWAS 1.0 bundled download

To download `weka.jar`

1. Go to http://www.cs.waikato.ac.nz/~ml/weka/index.html

2. Read their licensing terms and conditions.

3. Download a `weka zip` archive containing the weka source files and instructions to build the `weka.jar`.

4. The latest file at the time this guide was written was `weka-3-4-10.zip` located on the Download page under the "Other platforms (Linux, etc.)" heading.

Unzip the archive under `{work-area}`. This will create a new directory called `{work-area}/weka-3-4-10`.

In step 2 of installation, you will extract this file into the `caintegrator-spec` package.

Installing caGWAS

caGWAS is dependent on the following jars, as mentioned above: `caintegrator-analysis-commons.jar`, `caintegrator-application-commons.jar` and `caintegrator-spec.jar`.

The application requires these jars to be built prior to the build/deployment of `cagwas.war`. In this section, Steps 1 through 3 provide instructions on building these dependent jars. Step 4 and 5 provide instructions on building, configuring and deploying caGWAS.

Note: The build script creates a directory called "artifacts" in the same level as the directory where each of the components was extracted. For example, if you unzip `caintegrator-analysis-commons.zip` under the folder `c:\dev`, the build script will create a folder called `c:\dev\artifacts` to contain the binaries from the builds.

In the subsequent sections, `{work-area}` refers to the work area where the component zip archives will be extracted (such as `c:\dev`).

For detailed instructions for installing caGWAS, see:

- *Step 1: Build caintegrator-analysis-commons* on page 14
- *Step 2: Build caintegrator-spec* on page 15
- *Step 3: Build caintegrator-application-commons* on page 15

To ensure a correct installation of caGWAS, you must follow these steps in exactly this order.

Step 1: Build `caintegrator-analysis-commons`

1. Extract `caintegrator-analysis-commons.zip` in `{work-area}`. Verify that the `src` and `lib` folders are available under `{work-area}/caintegrator-analysis-commons`.

2. From the command shell (like `cmd`, `bash`, `sh` etc.), change the working directory to `{work-area}/caintegrator-analysis-commons`.

3. To build `caintegrator-analysis-commons`, run `ant -f build_dependency`.

4. Verify that there were no errors and that `caintegrator-analysis-commons.jar` has been created in {work-area}/artifacts.

Step 2: Build **`caintegrator-spec`**

Note: Recall that `weka.jar` was been extracted earlier. This file and `caintegrator-analysis-commons.jar` are required to build `caintegrator-spec.jar`.

1. Extract `caintegrator-spec.zip` under {work-area}. Verify that the `src` and `deployed_jars` folders are available under {work-area}/caintegrator-spec.

2. Copy the {work-area}/weka-3-4-10/weka.jar file to the {work-area}/ caintegrator-spec/deployed_jars folder.

3. From the shell prompt, change the working directory to {work-area}/caintegrator-spec.

4. Build `caintegrator-spec` by running `ant -f build_dependency`.

5. Verify that there were no errors and that `caintegrator-spec.jar` has been created under {work-area}/artifacts folder.

Step 3: Build **`caintegrator-application-commons`**

Note: `caintegrator-spec.jar` and `caintegrator-analysis-commons.jar` are required to build `caintegrator-application-commons`.

1. Extract `caintegrator-application-commons.zip` file under {work-area}. Verify that the `src` and `lib` folders are available under {work-area}/caintegrator-application-commons.

2. From the shell prompt, change working directory to {work-area}/caintegrator-application-commons.

3. Build `caintegrator-application-commons` by running `ant -f build_dependency`.

4. Verify that there were no errors and that `caintegrator-application-commons.jar` has been created in {work-area}/artifacts.

Configuring caGWAS

You can configure caGWAS to reflect custom properties. The properties listed in this section are not all of the properties in caGWAS but rather only those that you can customize.

To configure caGWAS

1. Set the following properties in the `build.properties` file.

```
****************************
MySQL Setup
****************************
```

```
# Database Properties
# select DB type: mysql or oracle
db.type=mysql

# REPLACE THE FOLLOWING:
YOUR_DB_SERVER:YOUR_DB_PORT,YOUR_DB_SCHEMA_NAME,
YOUR_DB_USERNAME, YOUR_DB_PASSWORD
cagwas.jndi.connection=cagwas
cagwas.databaseUser=YOUR_DB_USERNAME
cagwas.databasePassword=YOUR_DB_PASSWORD
cagwas.databaseServerURL=YOUR_DB_URL
cagwas.databaseServerPort=YOUR_DB_PORT
cagwas.databaseSchemaName=YOUR_DB_SCHEMA_NAME
cagwas.urlDbalias=jdbc:${db.type}:
${cagwas.databaseServerURL}:${cagwas.databaseServerPort}
${cagwas.databaseSchemaName}

# Use for MySQL
db.jdbcRuntimeDriver.mysql=com.mysql.jdbc.Driver
db.dialect.mysql=org.hibernate.dialect.MySQLDialect
db.exception.sorter.class.mysql=com.mysql.jdbc.integration
.jboss.ExtendedMysqlExceptionSorter

# External Properties Path
#UNIX EXAMPLE /local/content/${module_name}/config
#WINDOWS EXAMPLE C:/local/content/${module_name}/config
EXTERNAL.CONFIG.DIR=YOUR_EXTERNAL_PROPERTIES_DIRECTORY

***************************
Oracle Setup
***************************
# Database Properties
# select DB type: mysql or oracle
db.type=oracle

# REPLACE THE FOLLOWING:
YOUR_DB_SERVER:YOUR_DB_PORT,YOUR_DB_SCHEMA_NAME,
YOUR_DB_USERNAME, YOUR_DB_PASSWORD
cagwas.jndi.connection=cagwas
cagwas.databaseUser=YOUR_DB_USERNAME
cagwas.databasePassword=YOUR_DB_PASSWORD
cagwas.databaseServerURL=YOUR_DB_URL
cagwas.databaseServerPort=YOUR_DB_PORT
cagwas.databaseSchemaName=YOUR_DB_SCHEMA_NAME
cagwas.urlDbalias=jdbc:${db.type}:
${cagwas.databaseServerURL}:${cagwas.databaseServerPort}
${cagwas.databaseSchemaName}

# Use for Oracle DB
```

```
db.jdbcRuntimeDriver.oracle=oracle.jdbc.driver.
OracleDriver
db.dialect.oracle=org.hibernate.dialect.Oracle9Dialect
db.exception.sorter.class.oracle=org.jboss.resource.
adapter.jdbc.vendor.OracleExceptionSorter

# External Properties Path
#UNIX EXAMPLE /local/content/${module_name}/config
#WINDOWS EXAMPLE C:/local/content/${module_name}/config
EXTERNAL.CONFIG.DIR=YOUR_EXTERNAL_PROPERTIES_DIRECTORY
```

2. Set the following properties in the `mailzip.properties` file for your configuration.

```
# Project Name (Full)
project=YOUR_FULL_PROJECT_NAME

# Project Acronym
acronym=YOUR_PROJECT_ACRONYM

# Mail server hostname
host=YOUR_MAIL_SERVER

# Hostname and port of FTP server
ftpHostnameAndPort=YOUR_LOCATION_OF_FTP_FILES_TO_REFERENCE
_IN_EMAIL

# Tech support information
techSupportNumber=YOUR_TECH_SUPPORT_NUMBER
techSupportURL=YOUR_TECH_SUPPORT_URL
techSupportStartTime=YOUR_TECH_SUPPORT_START_TIME
techSupportEndTime=YOUR_TECH_SUPPORT_STOP_TIME
techSupportMail=YOUR_TECH_SUPPORT_EMAIL

# Request Email
userRequestMail=YOUR_USER_REQUEST_EMAIL
userRequestCC=YOUR_USER_REQUEST_CC

# Feedback Email
feedback.mailTo=YOUR_FEEDBACK_EMAIL
feedback.mailSubject=YOUR_FEEDBACK_SUBJECT

# Remote plone site or your server url such as localhost
remote.url=http://localhost:8080/cagwas
remote.about=/about_content.html
remote.update=/update_content.html
remote.registration=/controlled_data_access_content.html
remote.study1=/study1_content.html
remote.study2=/study2_content.html
remote.glossary=/glossary_content.html
remote.contacts=/contacts_content.html
```

```
remote.cite=/cite_content.html
remote.citeExample=/cite_example_content.html
remote.access=/data_access_content.html
remote.openAccess=/open_data_access_content.html
remote.controlledAccess=/
controlled_data_access_content.html

ftp_anonBrowse_URL=YOUR_ANONYMOUS_FTP_URL
ftp_secure_URL=YOUR_SECURE_FTP_URL
```

3. Make sure your JDBC driver is present in `${JBOSS_HOME}/server/default/lib` for either MySQL or Oracle.

4. Build using the supplied `build.xml`.

5. Copy the `cagwas-ds.xml` to the JBOSS `deploy` directory.

6. Configure JBOSS config files using the templates built for you.

```
login-config.xml
properties-services.xml
log4j.xml
```

7. Copy the following generated files to your specified `EXTERNAL_PROPERTIES_DIRECTORY`

```
mail.properties
zip.properties
cagwas.hibernate.cfg.xml
```

8. Modify your `${JBOSS_HOME_DIR}/bin/run.conf` to specify the `JAVA_OPTS` below

```
JAVA_OPTS="-server -Xms2048m -Xmx2048m -
XX:ThreadStackSize=128 -XX:SurvivorRatio=10 -
XX:PermSize=128m -XX:MaxPermSize=128m -
Dsun.rmi.dgc.client.gcInterval=3600000 -
Dsun.rmi.dgc.server.gcInterval=3600000 -
Djava.awt.headless=true"
```

9. Start JBOSS.

Deploying caGWAS

To deploy caGWAS, see the following sections:

- *Step 1: Build cagwas.war* below

- *Step 2: Configure JBoss for WGS application* on page 19

- *Step 3: Deploy WGS application under JBoss* on page 20

Step 1: Build **cagwas.war**

Note: `caintegrator-spec.jar`, `caintegrator-analysis-commons.jar` and `caintegrator-application-commons.jar` are required to build `cagwas.war`.

1. Extract `cagwas.zip` under {`work-area`}. Verify that the `src`, `not_deployed_jars`, `test`, `conf` and `WebRoot` folders are available under {`work-area`}/`cagwas`.

2. From the shell prompt, change the working directory to {`work-area`}/ `cagwas`.

3. In the {`work-area`}/`cagwas` folder, create the following two folders: `bulk_download_cagwas` `pub_download_cagwas`

4. Add the following two lines to the `mailzip.properties` file:

5. Build `cagwas.war` by running `ant build_war_anthill`.

6. Verify that there were no errors and that `cagwas.war` has been created in {`work-area`}/`artifacts`.

7. Go to the `artifacts` folder (which was created by running the `ant` build in step 1 and is located at /{`work-area`}/`artifacts`/). You should see the following files:

    º `ApplicationSecurityConfig.xml`

    º `oracle-ds.xml`

    º `properties-service.xml`

    º `login-config.xml`

    º `mail.properties`

    º `zip.properties`

Step 2: Configure JBoss for WGS application

1. Modify the {`work-area`}/`artifacts`/`oracle-ds.xml` with your database information for the `your_db_ip_address`, `your_db_instance`, `user-name`, `password` tags

```
<local-tx-datasource>
        <jndi-name>cagwas</jndi-name>
    <connection-url>
jdbc:oracle:thin:@your_db_ip_address:1521:your_db_instance
    </connection-url>
        <user-name>user name</user-name>
        <password>password</password>
    <driver-class>oracle.jdbc.driver.OracleDriver
    </driver-class>
        <exception-sorter-class-name>
org.jboss.resource.adapter.jdbc.vendor.OracleExceptionSorter
        </exception-sorter-class-name>
</local-tx-datasource>
```

2. Copy `oracle-ds.xml` from the {`work-area`}/`artifacts`/ folder and place it in: /`<JBOSS_HOME>`/`server/default/deploy/`. Alternatively, if the `oracle-ds.xml` already exists, copy the contents of `oracle-ds.xml` from the {`work-area`}/`artifacts`/ folder and add them to the existing `oracle-ds.xml` file.

> **Note:** The data source should be named `cagwas`.

3. Create a externalized properties folder such as
   `c:\caintegrator\externalized_properties_folder` on your local file system. This folder will be referred to as
   `<externalized_properties_folder>`

4. Copy the following files from the `{work-area}/artifacts/` folder to
   `<externalized_properties_folder>` that was created in the previously step.

   ```
   mail.properties
   zip.properties
   ```

   > **Note:** You may wish to change the default values under mail and zip property files. Those values can be changed at any time and will not require rebuilding the application, but you will need to restart the server.

5. Edit your `/<JBOSS_HOME>/server/default/deploy/properties-services.xml` file.

   a. Scroll down to

   ```
   <mbean
   code="org.jboss.varia.property.SystemPropertiesService"
   name="jboss:type=Service,name=SystemProperties">
   ```

   b. Copy the text that is inside the file `{work-area}/artifacts/` `properties-service.xml`, which should look like the following, where `c:\caintegrator\externalized_properties_folder is your` `<externalized_properties_folder>`

   ```
   <attribute name="Properties">

   gov.nih.nci.cagwas.zip.properties=C:\caintegrator\externalized_p
   roperties_folder\zip.properties

   gov.nih.nci.cagwas.mail.properties=C:\caintegrator\externalized_
   properties_folder\mail.properties

   <attribute/>
   ```

   and add it inside the `mbean` node.

Step 3: Deploy WGS application under JBoss

The `cagwas.war` is ready to be deployed to JBoss.

1. Copy the `cagwas.war` file, located at `/artifacts/` to `/<JBOSS_HOME>/server/default/deploy/`.

2. Using the command located in the `/<JBOSS_HOME>/server/default/` `deploy/bin` folder, restart the JBoss container that owns the directory where this file was placed.

Verifying Installation

It is very important to start JBoss as noted in the first step below. Start JBoss using the command located in the `/<JBOSS_HOME>/server/default/deploy/bin` folder.

To verify correct installation of caGWAS, do the following.

1. Start JBoss and navigate your web browser to: `http://<JBOSS_SERVER:PORT>/cagwas/` (for example `http://localhost:8080/cagwas`).

2. Select the **Browse data** tab.

3. Select GWAS Study 1, the default version, 2.0, and the default dataset option, Association Finding.

4. Click **Submit**. The Search Association Findings page appears.

5. Accept all defaults on the Search Association Findings page and then click **Submit**. The system retrieves all association findings records for the study and displays the results in a table.

caGWAS uses the NCICB common security module to control access to genotype and protected data under Study_Particpant objects. All the other relevant information is classified as *Limited Data Set* and does not require special access restrictions. If you would like to leverage CSM and LDAP for authorization/authentication and to establish security policies at your site, then please refer to the NCICB download site at http://ncicb.nci.nih.gov/download/ for CSM 3.1.

THIRD-PARTY TOOLS

caGWAS uses the following third-party `jar` files.

Third-Party Tool	*Version*	*URL*	*License Info/URL*
ehCache	1.1	http://ehcache.sourceforge.net/	Available under the Apache 1.1license. Ehcache's copyright and licensing has been reviewed and approved by the Apache Software Foundation, making ehcache suitable for use in Apache projects.
JFreechart	1.0.0 rc1	http://www.jfree.org/jfreechart/index.php	http://www.jfree.org/lgpl.php complete source code is included, under the terms of the GNU Lesser General Public Licence; http://www.jfree.org/jfreechart/index.php
Apache Struts	1.1	http://struts.apache.org/	
Rserve		http://stats.math.uni-augsburg.de/Rserve/	GPL software license
DWR	1.0	http://getahead.ltd.uk/home	http://www.apache.org/licenses/LICENSE-2.0.html
DOM4J	1.5.2	http://www.dom4j.org/	http://www.dom4j.org/license.html BSD License

Table A.1 Third-Party Tools

Third-Party Tool	Version	URL	License Info/URL
Hibernate	2.1.7c	http://www.hibernate.org	http://www.hibernate.org/356.html BSD License
Krysalis-jCharts	1.0.0-alpha-1	http://jcharts.sourceforge.net/index.html	http://jcharts.sourceforge.net/license.html BSD License
log4j	1.2.8	http://logging.apache.org/log4j/docs/	http://vmgump.apache.org/gump/public-jars/nlog4j/jars/LICENSE.txt
p6spy		http://www.p6spy.com/	http://www.ibiblio.org/maven/p6spy/licenses/license.htm
velocity	1.3.1	http://jakarta.apache.org/velocity/	http://jakarta.apache.org/velocity/docs/license.html
xdoclet	1.2b3-dev	http://xdoclet.sourceforge.net/	BSD http://xdoclet.sourceforge.net/xdoclet/licenses/xdoclet-license.html
xjavadoc	1.0	http://xdoclet.sourceforge.net/xjavadoc/	
db-ojb-1.0.rc4-src.jar	1.0.rc4		
jakarta regexp	1.2		http://jakarta.apache.org/regexp/
Java API for Servlets - javax.servlet.jar			
javax.servlet.jsp.jar			
JBOSS -j2ee		http://www.jboss.org/	
Java Data Objects	jdo.jar	http://www.javaranch.com/newsletter/200401/IntroToJDO.html	
Sun's reference implementation of the JDO specification	jdori.jar		
antlr.jar		http://www.antlr.org/	http://www.antlr.org/license.html
appStateClient.jar			

Table A.1 Third-Party Tools (Continued)

Third-Party Tool	*Version*	*URL*	*License Info/URL*
`batik-awt-util.jar`		http://xml.apache.org/batik/	
`batik-dom.jar`		http://xml.apache.org/batik/	
`batik-svggen.jar`		http://xml.apache.org/batik/	
`batik-util.jar`		http://xml.apache.org/batik/	
`batik-xml.jar`		http://xml.apache.org/batik/	
`c3p0-0.8.4.5.jar`			
caBIO	`caBIO.jar`	http://ncicb.nci.nih.gov/NCICB/infrastructure/cacore_overview/caBIO	
`cglib-full-2.0.2.jar`	2.0.2		
`commons-beanutils.jar`		http://jakarta.apache.org/commons/beanutils/	
`commons-collections-3.1.jar`	3.1	http://jakarta.apache.org/site/downloads/downloads_commons-collections.cgi	
`commons-dbcp.jar`			
`commons-digester.jar`		http://jakarta.apache.org/commons/digester/	
`commons-fileupload.jar`			
`commons-lang-2.0.jar`			
`commons-logging`	`1.0.4.jar`	http://jakarta.apache.org/commons/logging/	
`commons-pool.jar`			
`commons-validator.jar`			
`csmapi.jar`		http://ncicb.nci.nih.gov/NCICB/infrastructure/cacore_overview/csm/	
`jaas.jar`		http://java.sun.com/products/jaas/install_notes.html	

Table A.1 Third-Party Tools (Continued)

Third-Party Tool	Version	URL	License Info/URL
jakarta-oro.jar		http://jakarta.apache.org/oro/	
jcommon-1.0.0-rc1.jar	1.0.0		
jcs.jar			
jdo.jar			
odmg-3.0.jar	3.0		
ojdbc14.jar			

Table A.1 Third-Party Tools (Continued)

INDEX