Table of Contents

[RESOURCES 3](#_Toc379061773)

[Access GitHub’s IHTSDO/ISAAC project 3](#_Toc379061774)

[Ensure have JDK 1.8 installed on system 3](#_Toc379061775)

[Ensure have Maven 3.0.5 installed on system 3](#_Toc379061776)

[Download the entire IHTSDO/ISAAC project from GitHub: 3](#_Toc379061777)

[Berkeley Database 3](#_Toc379061778)

[General Setup 4](#_Toc379061779)

[Setup your settings.xml file 4](#_Toc379061780)

[Setup Project in IDE 5](#_Toc379061781)

[Eclipse 5](#_Toc379061782)

[NetBeans 5](#_Toc379061783)

[IntelliJ 5](#_Toc379061784)

[Command Line 5](#_Toc379061785)

[Create a Git local repository of ISAAC 5](#_Toc379061786)

[Desktop Client 5](#_Toc379061787)

[Command Line 7](#_Toc379061788)

[Import local Git repository into IDE 7](#_Toc379061789)

[Eclipse 7](#_Toc379061790)

[NetBeans 9](#_Toc379061791)

[IntelliJ 9](#_Toc379061792)

[Command Line 9](#_Toc379061793)

[Build project 9](#_Toc379061794)

[Eclipse 9](#_Toc379061795)

[NetBeans 10](#_Toc379061796)

[IntelliJ 10](#_Toc379061797)

[Command Line 10](#_Toc379061798)

[Deploy Berkeley Database 10](#_Toc379061799)

[Eclipse 10](#_Toc379061800)

[NetBeans 11](#_Toc379061801)

[IntelliJ 11](#_Toc379061802)

[Command Line 11](#_Toc379061803)

[Run application 11](#_Toc379061804)

[Eclipse 11](#_Toc379061805)

[NetBeans 12](#_Toc379061806)

[IntelliJ 12](#_Toc379061807)

[Command Line 12](#_Toc379061808)

# RESOURCES

## Access GitHub’s IHTSDO/ISAAC project

1. Create a GitHub account at <http://github.com>
2. Request access to the IHTSDO /ISAAC Project on GitHub from a project administrator
   1. Currently, only Rory Davidson may provide you with permissions to the project
3. Access IHTSDO/ISAAC project at: <https://github.com/IHTSDO/ISAAC>

## Ensure have JDK 1.8 installed on system

1. Download from <https://jdk8.java.net/download.html>
2. Install as instructed

## Ensure have Maven 3.0.5 installed on system

1. Download from <http://maven.apache.org/download.cgi>
2. Install as instructed

## Download the entire IHTSDO/ISAAC project from GitHub:

1. ISAAC (top level folder)
   1. Contains the multi-module master pom file.
2. app-util
   1. Contains shared application utilities
3. data-model
   1. Contains shared application data models
4. import-export
   1. Contains Import/Export view code
5. isaac-app
   1. Main application project
   2. Aggregates other view projects
6. lego-view
   1. Contains LEGO view code
7. resources
   1. Contains documents & resources
8. search-view
   1. Contains search view code
9. taxonomy-view
   1. Contains taxonomy view code

## Berkeley Database

1. Log into VA-Archiva at <https://mgr.servers.aceworkspace.net/apps/va-archiva/index.action>
2. Download <https://mgr.servers.aceworkspace.net/apps/va-archiva/repository/all/org/ihtsdo/otf/tcc-test-data/3.0/tcc-test-data-3.0.zip>

# General Setup

## Setup your settings.xml file

1. Pull sample settings.xml file from <https://github.com/IHTSDO/ISAAC/tree/master/resources> folder.
2. Replace <username> and <password> with your VA-Archiva credentials:

…..

<servers>

<server>

<id>va-maestro</id>

<username>VA-ARCHIVA\_USER\_NAME</username>

<password>VA-ARCHIVA\_USER \_PASSWORD</password>

</server>

</servers>

…..

1. Define your repositories

…..

<profiles>

<profile>

<id>my-profile</id>

<activation>

<activeByDefault>true</activeByDefault>

</activation>

<repositories>

<repository>

<id>va-maestro</id>

<name>VA Maestro</name>

<url>https://mgr.servers.aceworkspace.net/apps/va-archiva/repository/all/</url>

</repository>

</repositories>

<pluginRepositories>

<pluginRepository>

<id>va-maestro</id>

<name>VA Maestro</name>

<url>https://mgr.servers.aceworkspace.net/apps/va-archiva/repository/all/</url>

</pluginRepository>

</pluginRepositories>

</profile>

</profiles>

…..

# Setup Project in IDE

## Eclipse

1. Via **Windows-Preferences**, ensure **Java-Installed JREs** version is using Java 8 JDK.
2. Via **Windows-Preferences**, ensure **Maven-Installations** version is using local maven 3.0.5
3. Via **Windows-Preferences**, ensure **Maven-User Settings** has the User Settings pointing to the proper settings.xml file updated in General Setup portion of this document

## NetBeans

## IntelliJ

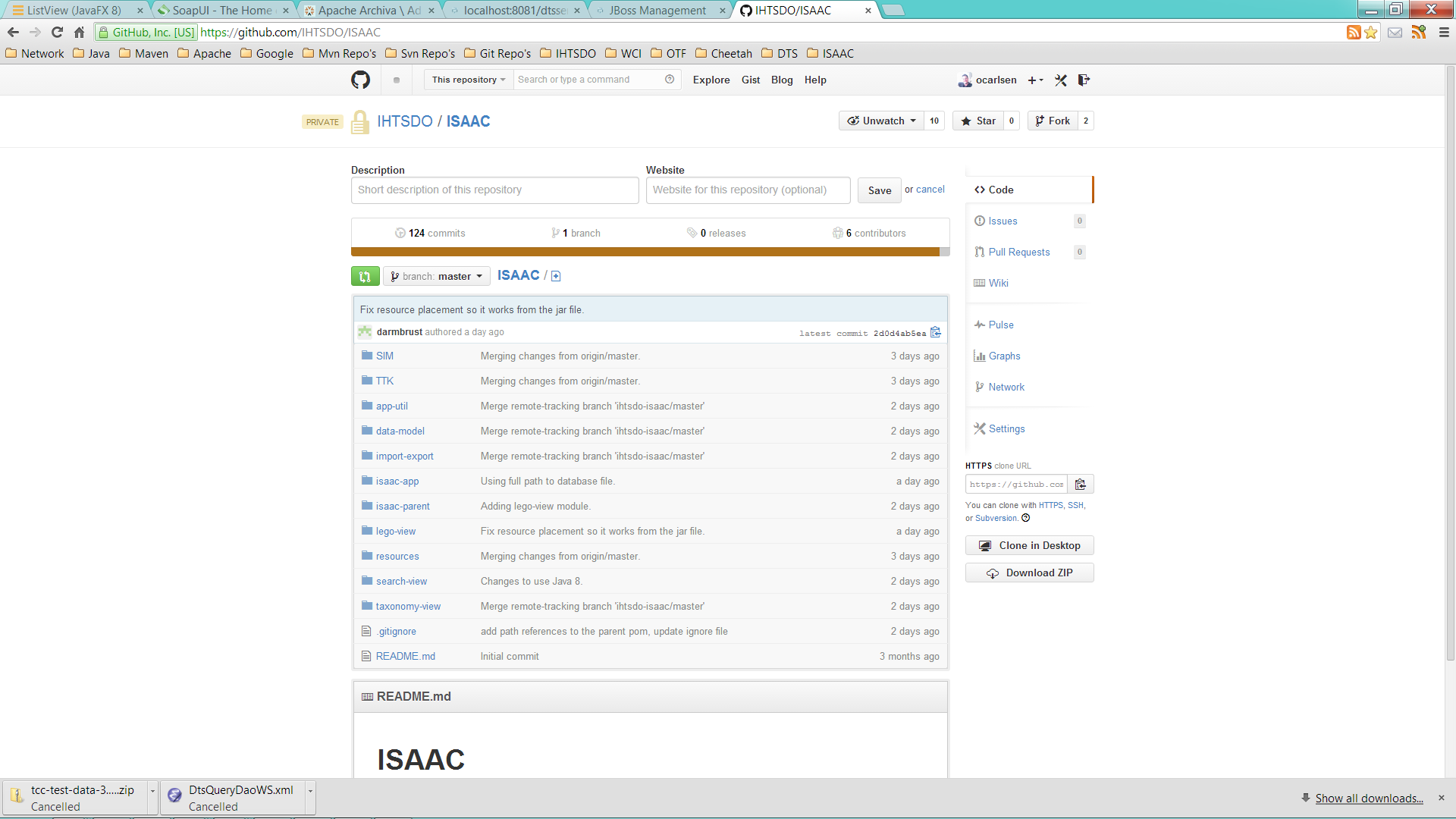
## Command Line

# Create a Git local repository of ISAAC

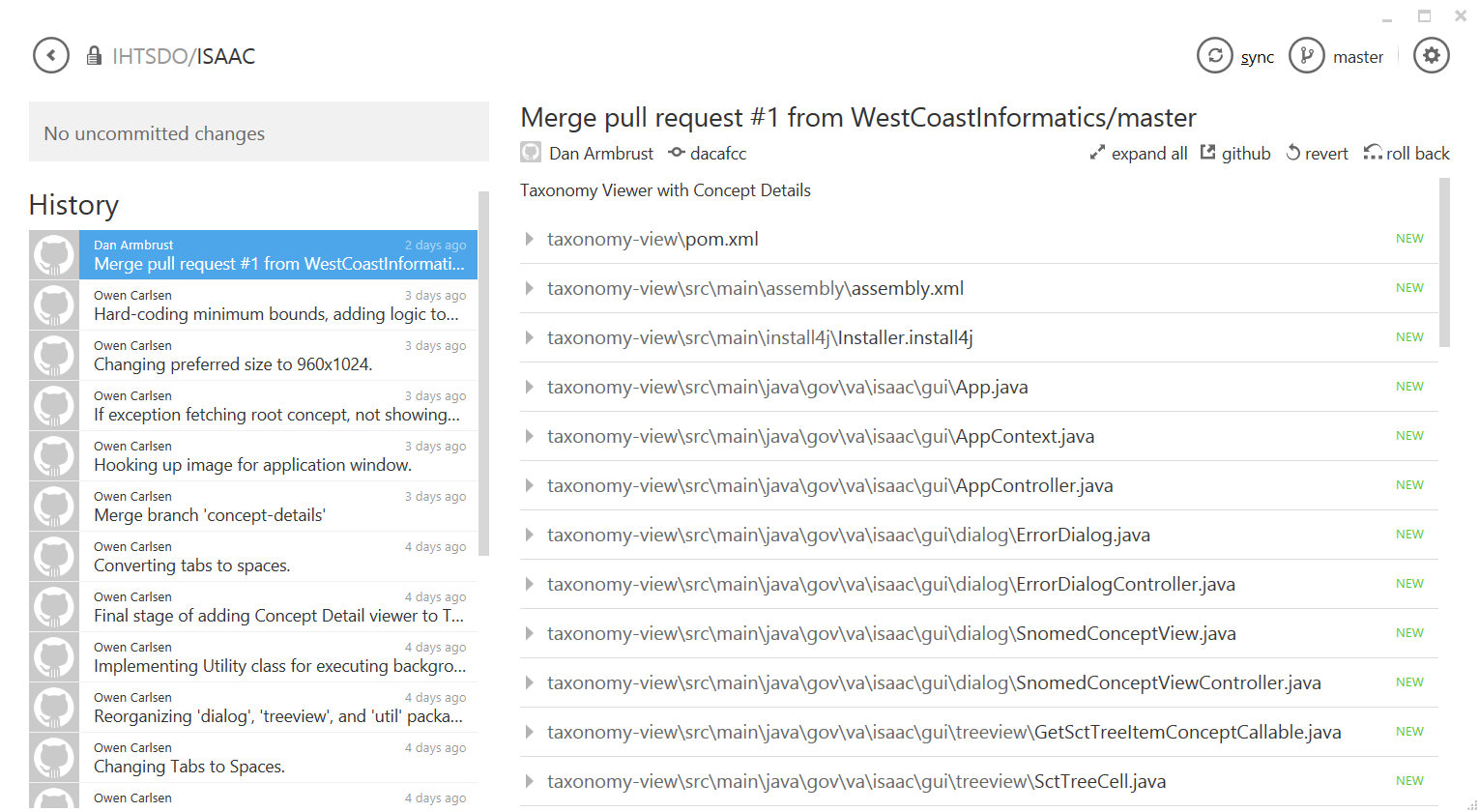
## Desktop Client

Demonstrated for Windows via Windows GitHub client

1. Go to <https://github.com/IHTSDO/ISAAC>
2. Select ‘Clone in Desktop’ button at lower-right



1. Opens GitHub client
2. Content will download
3. Once done, GitHub client will look like this



1. Locate local repository
   1. Click on settings button
   2. Select **open in explorer** option
   3. Explorer opens in directory containing local Git repository

## Command Line

Demonstrated in a Unix/Linux terminal

1. Change directory to the location you wish to clone the repository.

cd /Projects/ISAAC

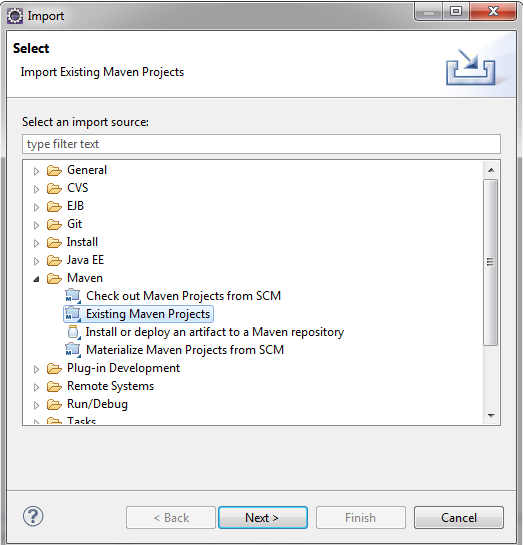
1. Download the project with the Git “clone” command.

git clone https://github.com/IHTSDO/ISAAC

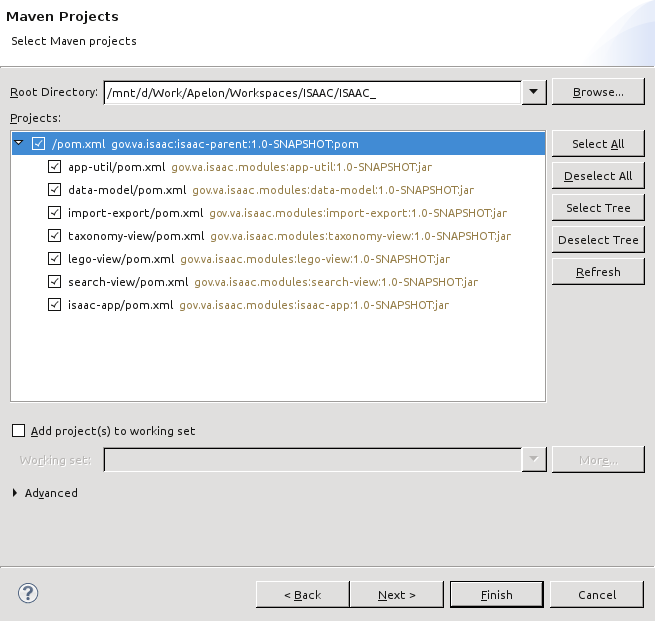
# Import local Git repository into IDE

## Eclipse

1. Select **File-Import**
2. Select **Maven-Existing** Maven Project option



1. Point **Root Directory** to the “ISAAC” folder in your local GitHub repository and select all the sub-projects.



1. Click the **Finish** button when done.

## NetBeans

## IntelliJ

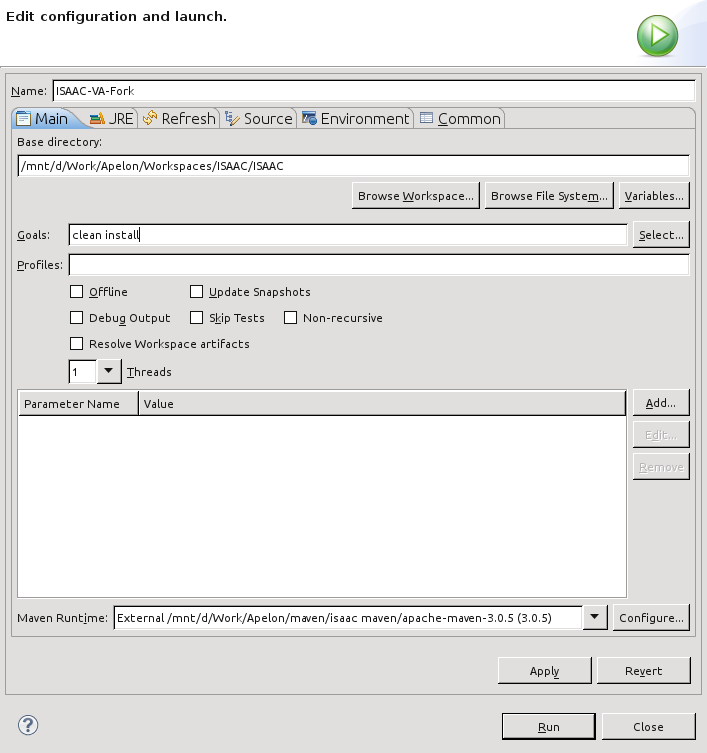
## Command Line

# Build project

The following sections describe how to build the ISAAC application.

## Eclipse

1. Create a Run Configuration for the project
   1. Right-click the “ISAAC” project and select **Run As-Maven Build…** from the menu
   2. Type “clean install” in the **Goals** field of the **Edit Configuration** dialog:



1. Click the **Run** button to build the ISAAC application and all its modules.

## NetBeans

## IntelliJ

## Command Line

Demonstrated in a Unix/Linux terminal

1. Change directory to the location containing the ISAAC repository cloned in [Create a Git local repository of ISAAC](#_Create_a_Git).

cd /Projects/ISAAC/

1. Build the project with the Maven as usual.

mvn clean install

1. The ISAAC application will be built into a ZIP file in the “isaac-app/target” folder.

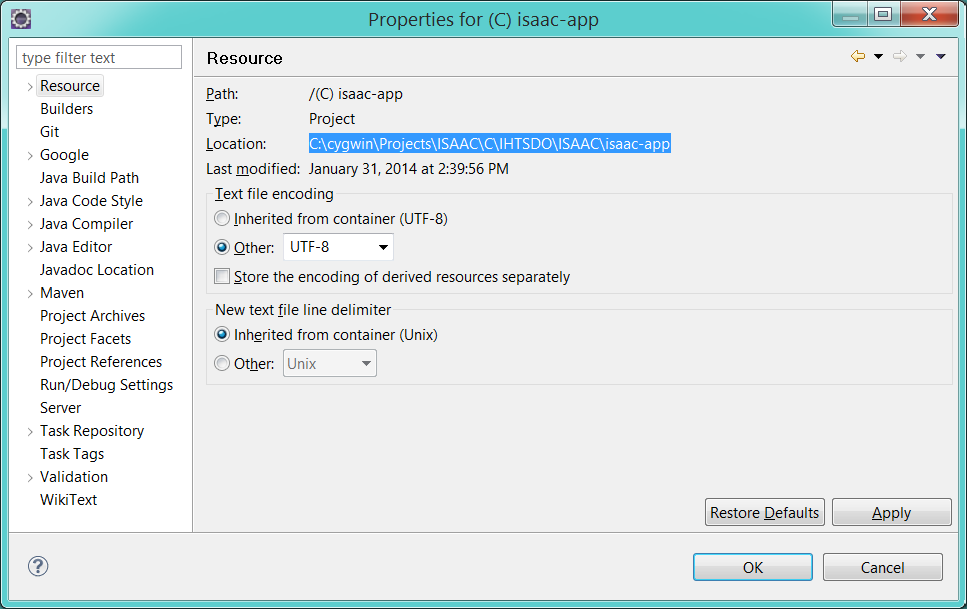
# Deploy Berkeley Database

The ISAAC application will need a database to connect to. The following steps describe how to deploy the database described in the [Berkeley Database](#_Berkeley_Database) section.

1. Locate the tcc-test-data-3.0.zip file you downloaded.
2. Extract the ZIP file into a directory called tcc-test-data-3.0.
3. Note the extracted “berkeley-db” folder inside. This contains the database content.

## Eclipse

1. Right-click the “isaac-app” project and select **Properties**.
2. Select the **Resources** group on the left.
3. In the Properties dialog, make a note of the **Location** field:



1. Copy the “berkeley-db” folder you extracted above into this directory.

## NetBeans

## IntelliJ

## Command Line

Demonstrated in a Unix/Linux terminal

1. Return to the directory containing the “target” folder of the “isaac-app” project.

cd /Projects/ISAAC/isaac-app/target

1. Copy the “berkeley-db” folder you extracted above into this directory.

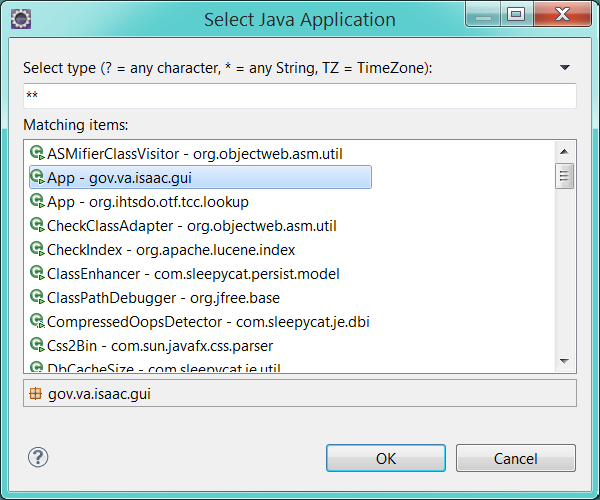
cp -r /tcc-test-data-3.0/berkeley-db .

# Run application

The following sections describe how to run the ISAAC application.

## Eclipse

1. Create a Run Configuration for the app
   1. Right-click the “isaac-app” project and select **Run As-Java Application** from the menu.
   2. Select “App – gov.va.isaac.gui” from the **Matching Items** list of the **Select Java Application** dialog:



1. Click the **Ok** button to start the ISAAC application. An Eclipse run configuration will be automatically created for you.

## NetBeans

## IntelliJ

## Command Line

Demonstrated in a Unix/Linux terminal

1. Change directory to the location containing the “target” folder of the “isaac-app” project you built in [Build project](#_Build_project).

cd /Projects/ISAAC/isaac-app/target

1. Unzip the ISAAC App ZIP file.

jar xvf isaac-app.zip

1. This will extract several scripts and a lib folder of JAR files into the “target” folder. Locate the appropriate script and execute it.

./isaac-app64.sh