Table of Contents

[RESOURCES 2](#_Toc373235356)

[Access GitHub’s IHTSDO/ISAAC project 2](#_Toc373235357)

[Ensure have JDK 1.7 or JDK 1.8 installed on system 3](#_Toc373235358)

[Ensure have Maven 3.0.5 installed on system 3](#_Toc373235359)

[Subfolders found under IHTSDO/ISAAC Git Project on GitHub to download: 3](#_Toc373235360)

[1. TK3 (Toolkit 3): 3](#_Toc373235361)

[2. Icons: 3](#_Toc373235362)

[3. (SIM) Simple Integrated Model 3](#_Toc373235363)

[Download a Berkeley Database 3](#_Toc373235364)

[General Setup 3](#_Toc373235365)

[1. Ensure your settings.xml file looks similarly 3](#_Toc373235366)

[2. Verify the validity of the reference to <java.home> in ttk.pom 6](#_Toc373235367)

[3. Identify javafx version associated with installed JDK 6](#_Toc373235368)

[Setup Projects in IDE 6](#_Toc373235369)

[Eclipse Workspace Setup 6](#_Toc373235370)

[NetBeans 6](#_Toc373235371)

[IntelliJ 6](#_Toc373235372)

[Command Line 6](#_Toc373235373)

[Create a Git local repository of ISAAC 6](#_Toc373235374)

[Import local Git repository into IDE 8](#_Toc373235375)

[Eclipse 8](#_Toc373235376)

[NetBeans 10](#_Toc373235377)

[IntelliJ 10](#_Toc373235378)

[Command Line 10](#_Toc373235379)

[Verify Maven projects ready for build 10](#_Toc373235380)

[Force inclusion of JavaFX into IDE (JDK 1.7 only) 10](#_Toc373235381)

[Eclipse 10](#_Toc373235382)

[NetBeans 12](#_Toc373235383)

[IntelliJ 12](#_Toc373235384)

[Command Line 12](#_Toc373235385)

[Build three ISAAC projects 12](#_Toc373235386)

[Eclipse 12](#_Toc373235387)

[NetBeans 14](#_Toc373235388)

[IntelliJ 14](#_Toc373235389)

[Command Line 14](#_Toc373235390)

[Install Glassfish 4.0 14](#_Toc373235391)

[Eclipse 15](#_Toc373235392)

[NetBeans 18](#_Toc373235393)

[IntelliJ 18](#_Toc373235394)

[Command Line 18](#_Toc373235395)

[Deploy Berkeley Database onto Server 18](#_Toc373235396)

[Start Glassfish Server 18](#_Toc373235397)

[Eclipse 18](#_Toc373235398)

[NetBeans 19](#_Toc373235399)

[IntelliJ 19](#_Toc373235400)

[Command Line 19](#_Toc373235401)

[Configure Glassfish Properties 19](#_Toc373235402)

[Deploy SIM Rest Server 19](#_Toc373235403)

[Via AutoDeploy 19](#_Toc373235404)

[Via Glassfish Administrator Web Page 19](#_Toc373235405)

[Test Deployment 20](#_Toc373235406)

# 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 credentials to the project
3. Access IHTSDO/ISAAC project at: <https://github.com/IHTSDO/ISAAC>

## Ensure have JDK 1.7 or JDK 1.8 installed on system

1. Download from <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
2. Install as instructed
3. Verify your version has jfxrt.jar under your jdk’s jre/lib folder
   1. jdk1.7.0\_45\jre\lib\jfxrt.jar

## Ensure have Maven 3.0.5 installed on system

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

## Subfolders found under IHTSDO/ISAAC Git Project on GitHub to download:

### TK3 (Toolkit 3):

1. Contains the refactored core libraries
2. Does not contain GUI elements as the existing workbench GUI has been removed based on the architecture review’s recommendations

### Icons:

1. Contains a collection of open source icons
2. Will eventually replace the icon library used by the workbench.

### (SIM) Simple Integrated Model

1. Contains a simple event-driven document model for representing patient records

## Download a Berkeley Database

1. <To Be Completed>

# General Setup

## Ensure your settings.xml file looks similarly

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

<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0 http://maven.apache.org/xsd/settings-1.0.0.xsd">

<localRepository>

C:/Code/.m2/repository-eclipse-test

</localRepository>

<servers>

<server>

<id>IHTSDO-maestro</id>

<username>*Archiva-User*</username>

<password> *Archiva-Password*</password>

</server>

<server>

<id>VA</id>

<username>*Archiva-User*</username>

<password> *Archiva-Password*</password>

</server>

<server>

<id>VA-SIM</id>

<username>*Archiva-User*</username>

<password> *Archiva-Password*</password>

</server>

<server>

<id>VA-3rdParty</id>

<username>*Archiva-User*</username>

<password> *Archiva-Password*</password>

</server>

</servers>

<profiles>

<profile>

<id>standard-extra-repos</id>

<activation>

<activeByDefault>true</activeByDefault>

</activation>

<repositories>

<repository>

<id>maven2</id>

<name>Java.net Repository for Maven2</name>

<url>http://download.java.net/maven/2</url>

</repository>

<repository>

<id>oracleReleases</id>

<name>Oracle Released Java Packages</name>

<url>http://download.oracle.com/maven</url>

</repository>

<repository>

<id>EclipseLink</id>

<name>eclipselink</name>

<url>http://download.eclipse.org/rt/eclipselink/maven.repo</url>

</repository>

<repository>

<id>VA</id>

<name>VA DataFiles Maestro</name>

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

</repository>

<repository>

<id>VA-SIM</id>

<name>VA DataFiles Maestro</name>

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

</repository>

<repository>

<id>VA-3rdParty</id>

<name>VA DataFiles Maestro</name>

<url>https://mgr.servers.aceworkspace.net/apps/va-archiva/repository/3rdParty/</url>

</repository>

<repository>

<id>netbeans</id>

<name>Netbeans-based Maven</name>

<url>http://bits.netbeans.org/maven2/</url>

</repository>

</repositories>

</profile>

</profiles>

<mirrors>

<mirror>

<id>IHTSDO-maestro</id>

<mirrorOf>external:\*,!VA,!maven2,!netbeans,!VA-SIM,!VA-3rdParty,!EclipseLink</mirrorOf>

<name>VA</name>

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

</mirror>

</mirrors>

</settings>

## Verify the validity of the reference to tools.jar in ttk-mmb.pom

If have error in the pom line: <systemPath>${java.home}/../lib/tools.jar</systemPath>

where the artifact “maven-jaxb-schemagen-plugin” is defined

1. Try changing Path Separator (Unix vs Windows)
2. Use env.JAVA\_HOME property
   1. <systemPath >${env.JAVA\_HOME}</systemPath >
3. Hard-Code tools.jar location
   1. <systemPath>C:\Program Files\Java\jdk1.7.0\_45\lib\tools.jar</systemPath>

## Identify javafx version associated with installed JDK

1. Search the java release notes associated with your jdk version to identify the version of javafx available in your java installation
2. <http://www.oracle.com/technetwork/java/javase/7u-relnotes-515228.html>
3. Locate your jdk version of release notes
4. Search on webpage for string “JavaFX Release Notes”
5. Under this section of document, locate JavaFX version
   1. JavaFX is now part of JDK. JDK 7u45 release includes JavaFX version **2.2.45**
6. Record version for later use (2.2.45)

# Setup Projects in IDE

## Eclipse Workspace Setup

1. Via **Windows-Preferences**, ensure **Java-Installed JREs** version is using latest 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
4. Via **Projects**, disable **Build Automatically**

## NetBeans

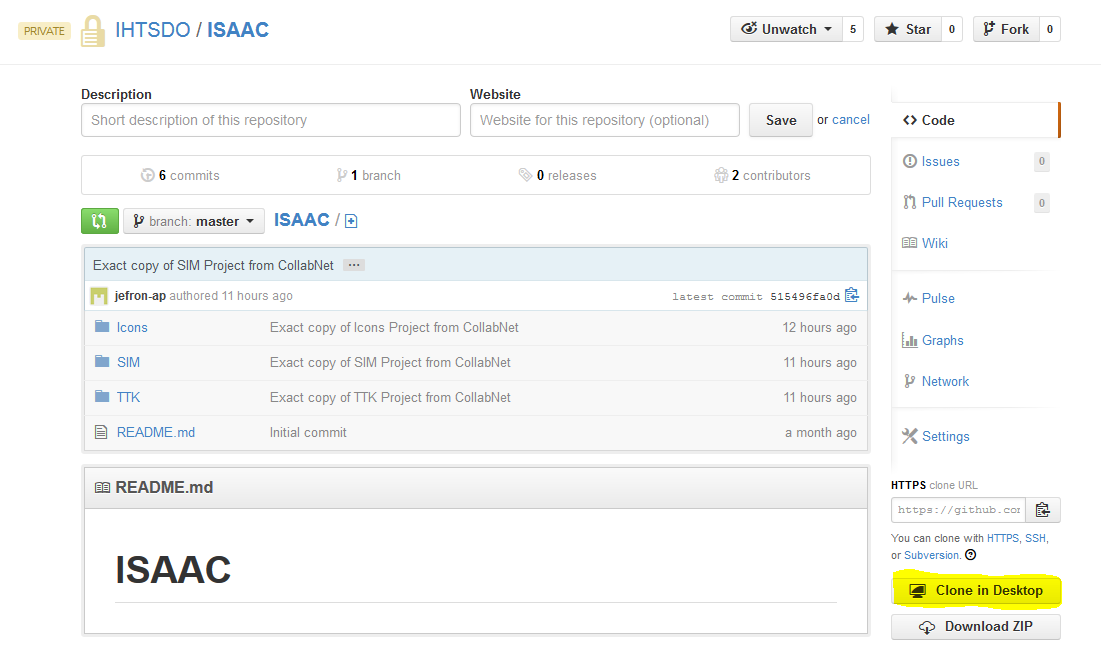
## IntelliJ

## Command Line

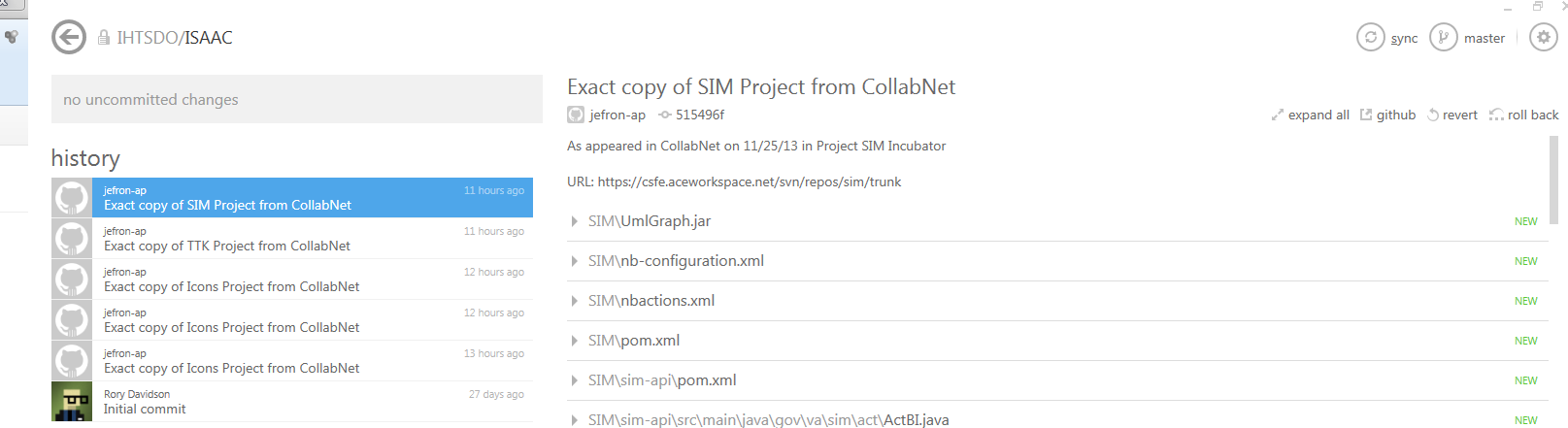
# Create a Git local repository of ISAAC

Demonstrated for Windows via Windows GitHub client

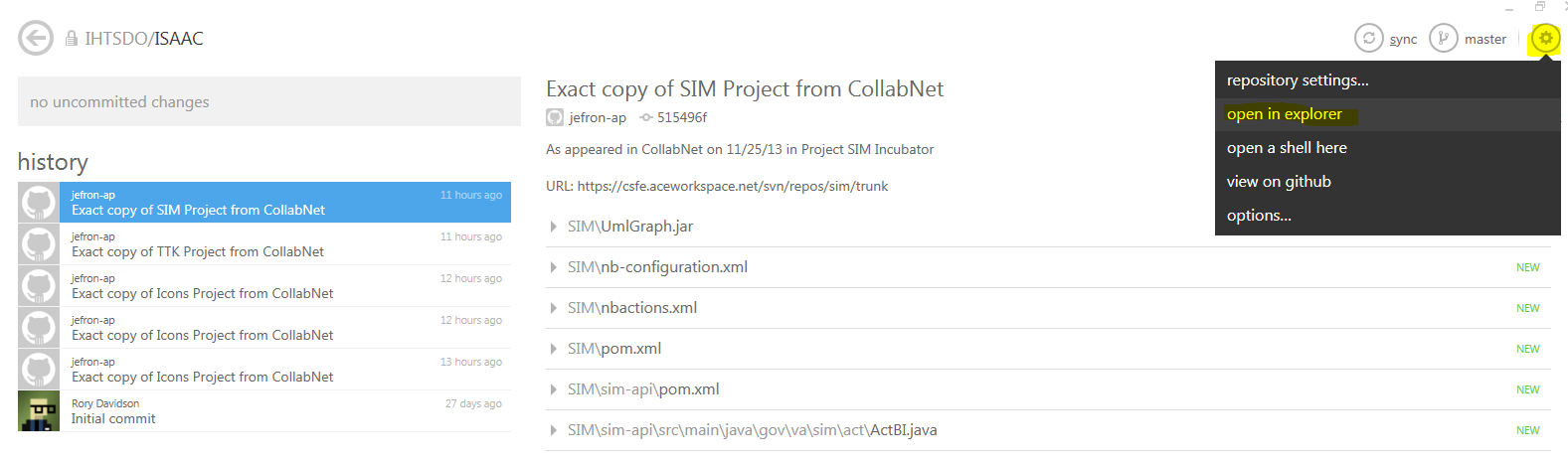
1. Goto <https://github.com/IHTSDO/ISAAC>
2. Select ‘Clone in Desktop’ button at lower-right



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



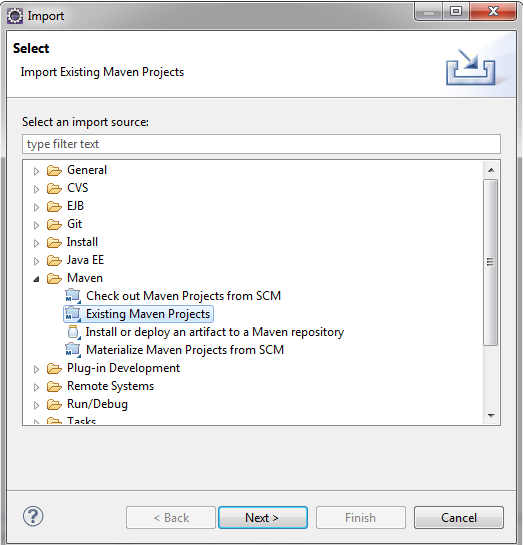
1. Locate local repository
   1. Click on settings button
   2. Select **open in explorer** option
   3. Explorer opens in directory containing local Git repository



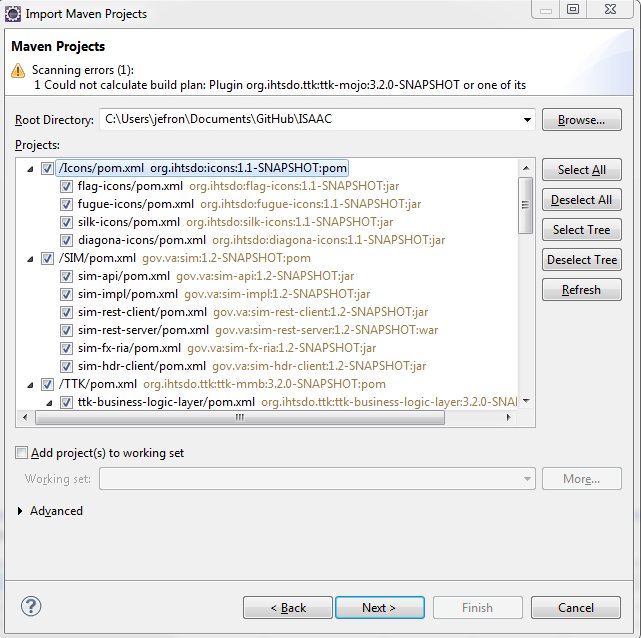
# Import local Git repository into IDE

## Eclipse

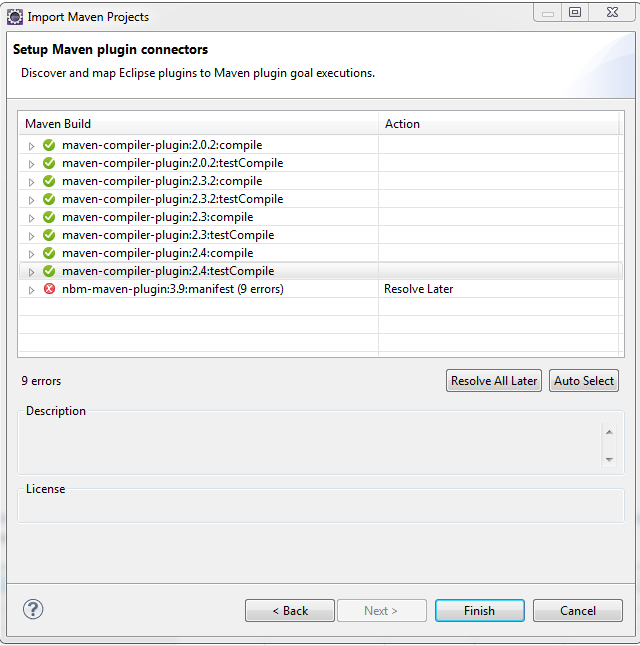
1. Select **File-Import**
2. Select **Maven-Existing Maven Project** option



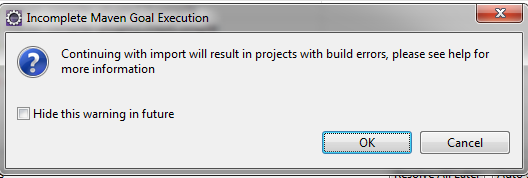
1. Point Root Directory to ISAAC’s local GitHub repository



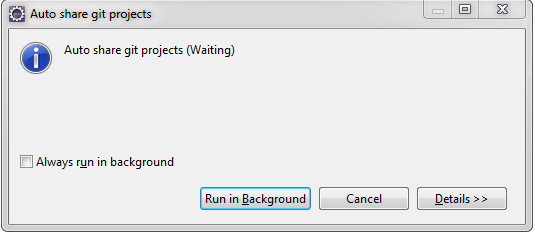
1. Once project finished loading up, select **Next** button
2. Even if **Setup Maven plugin connectors** shows errors, select **Finish** button



1. Select **OK** button if warning of build errors dialog box shows



1. Wait for project to import, will import Maven dependent artifacts to local Maven repository
2. No action to take when **Auto share git projects** dialog box is displayed



## NetBeans

## IntelliJ

## Command Line

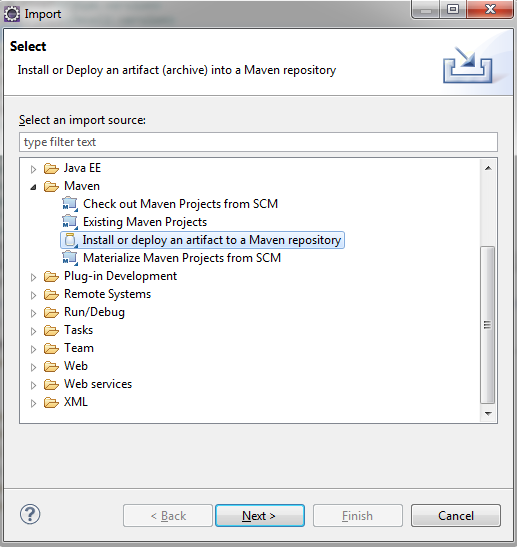
# Verify Maven projects ready for build

## Force inclusion of JavaFX into IDE (JDK 1.7 only)

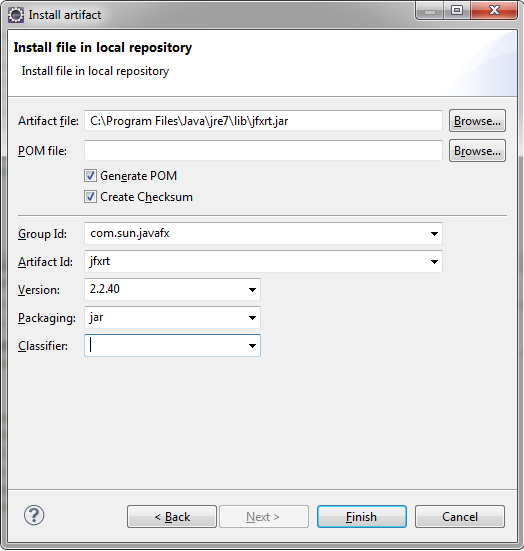
This is an issue that is universal, not just ISAAC based

### Eclipse

1. Select **File-Import** from Menu
2. Select **Maven-Install or deploy an artifact to a Maven repository** option



1. Select **Next** button
2. Fill out values
   1. Point **Artifice file** to jre\lib\jfxrt.jar
   2. Leave **Pom** file blank
   3. Set **Group Id** as “com.sun.javafx”
   4. Set **Artifact Id** as “jfxrt”
   5. Set **Version** to javafx version identified under *General Setup* portion of this document
   6. Verify **Packaging** set to “jar”



* 1. Select **Finish** button

### NetBeans

### IntelliJ

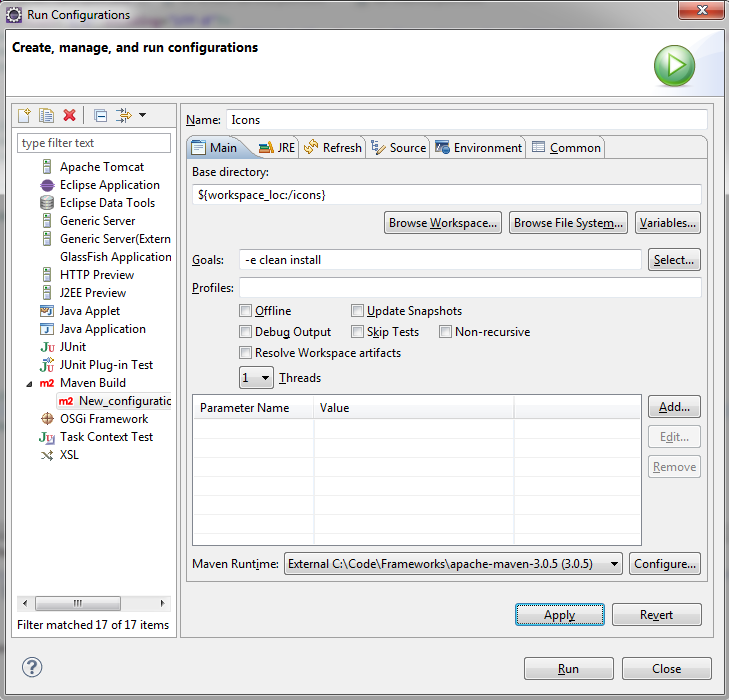
### Command Line

# Build three ISAAC projects

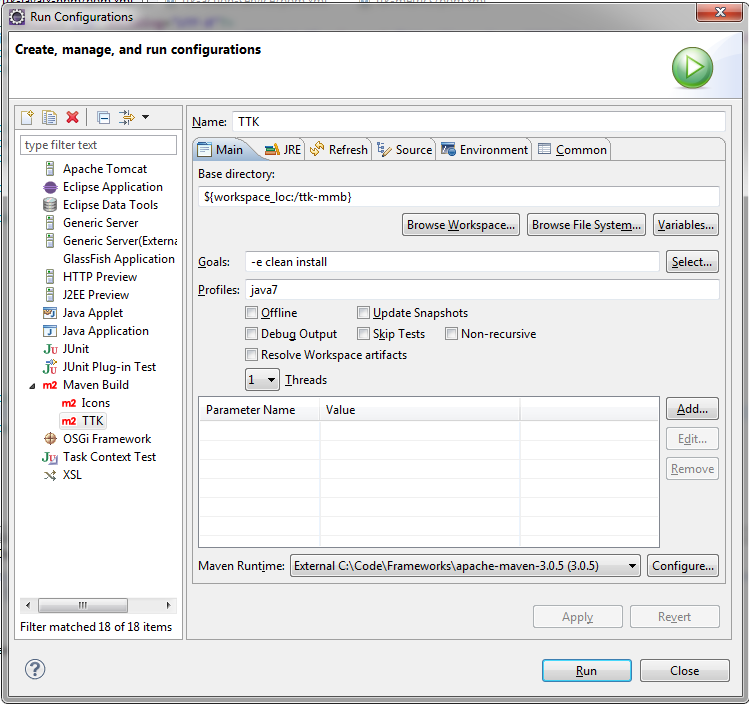
1. Icons
2. TTK
3. SIM

## Eclipse

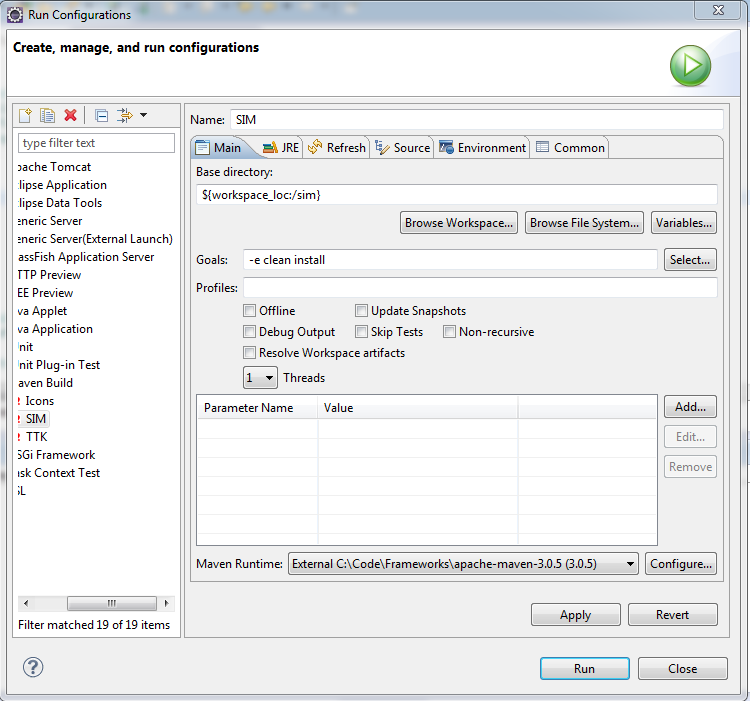
1. Create Run Configurations for each separate project
   1. Select **Run-Run Configurations** from Menu
   2. For each project
      1. Double click **Maven Build** option
      2. Name configuration
      3. Select Workspace
      4. Enter “-e clean install” for Goals
      5. Enter “java7” for Profiles
         1. For TTK only
         2. If building with jre 1.8, profile needs to be “java8”
      6. Save by selecting **Apply** button
   3. Close
   4. Icons Run Configuration (select maven icons project)



* 1. TTK Run Configuration (select maven ttk-mmb project)



* 1. SIM Run Configuration (select maven sim project)



1. Build Icons
   1. If not already open, select **Run-Run Configurations** from menu
   2. Select Icons under **Maven Build**
   3. Select **Run** button
2. Build TTK
   1. If not already open, select **Run-Run Configurations** from menu
   2. Select TTK under **Maven Build**
   3. Select **Run** button
3. Build SIM
   1. If not already open, select **Run-Run Configurations** from menu
   2. Select SIM under **Maven Build**
   3. Select **Run** button

## NetBeans

## IntelliJ

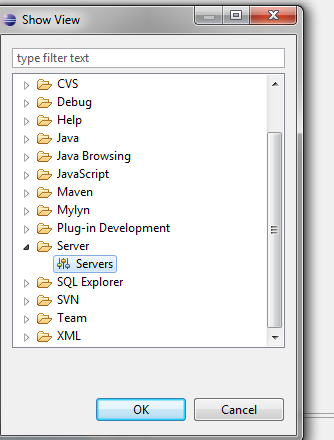
## Command Line

# Install Glassfish 4.0

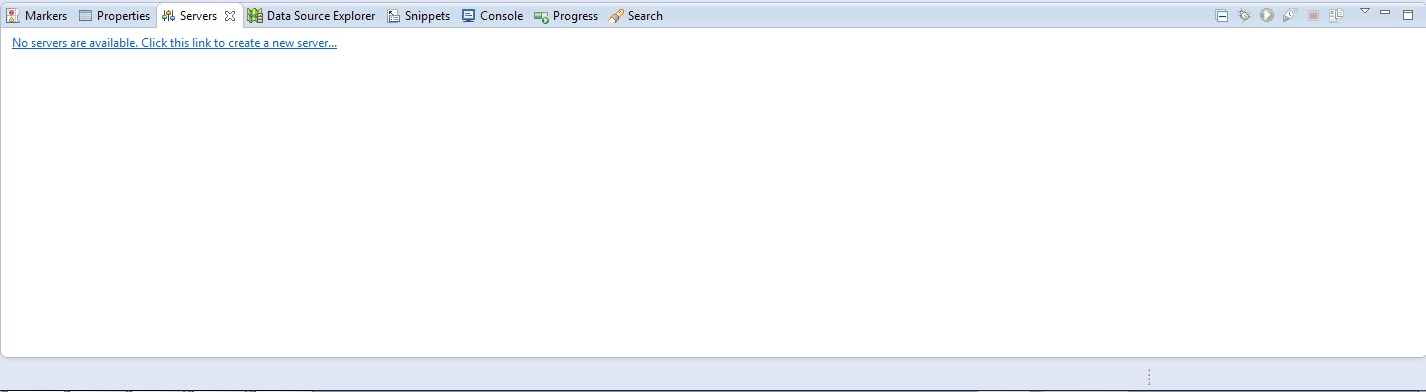
If already installed, skip to section Deploy SIM Rest Server

## Eclipse

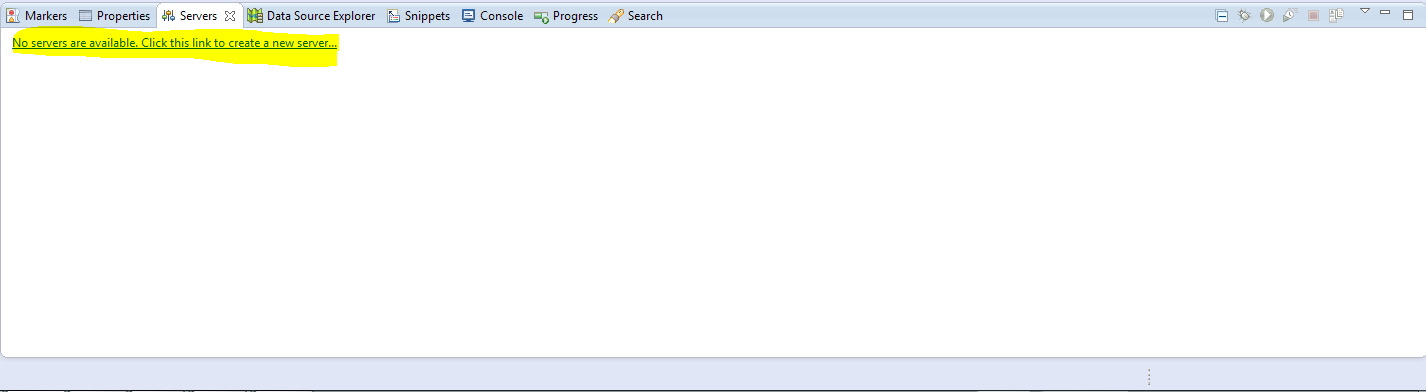
1. Bring up Servers View
   1. Open Window – Open View
   2. Select Others
   3. Select Server – Servers



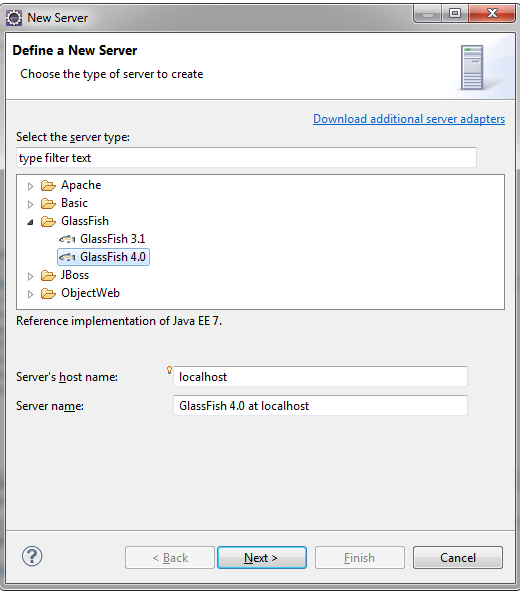
* 1. Select Servers tab



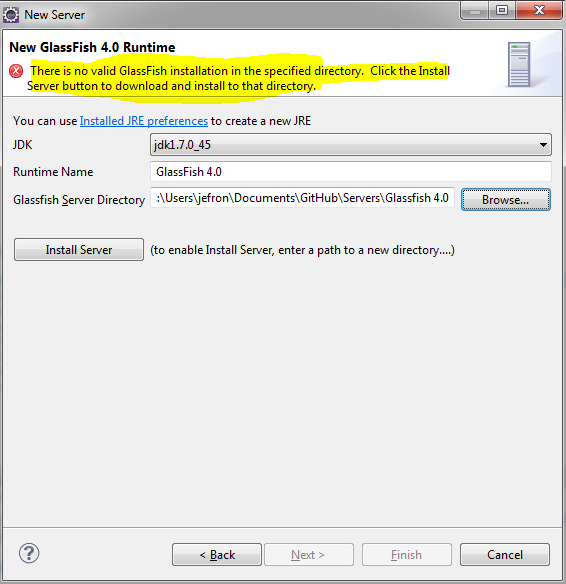
1. Create new Glassfish 4.0 Server instance
   1. Click “No servers are available. Click this link to create a new server” link



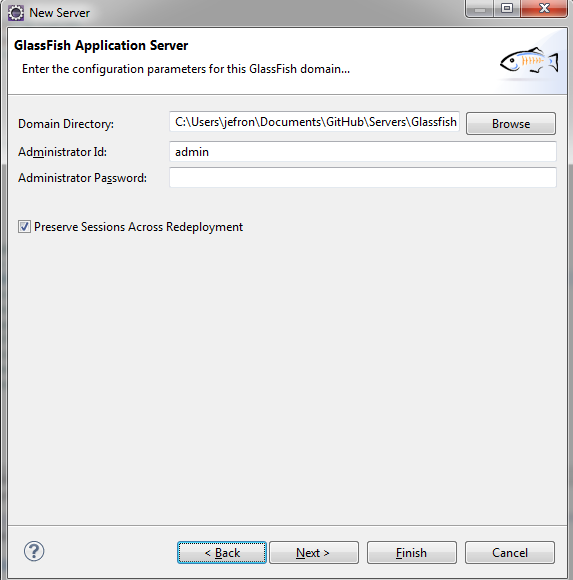
* 1. Select **Glassfish-Glassfish 4.0** option



* 1. Select **Next** button
  2. Define Glassfish Runtime Directory
     1. For JDK, select one specified in workspace
     2. For Runtime Name, select any name
     3. For Glassfish Server Directory, select Glassfish 4.0 installation location
        1. If Glassfish 4.0 has already been installed in specified location, proceed to step ‘F’
  3. Install Glassfish 4.0
     1. The Dialog Box will complain that a Glassfish 4.0 installation cannot be located in the specified Glassfish Server Directory



* + 1. Select **Install Server** button
    2. Select **Next** button ensuring that license is accepted
    3. Glassfish 4.0 will be downloaded and installed
  1. Configure Glassfish 4.0 instance
     1. Select Domain Directory (default is appropriate)
     2. Select Administrator Id (default is appropriate)
     3. Select Administrator Password (default is appropriate)



* + 1. Select **Next** button
    2. May deploy sim-rest-server at this point, but ignore for now
       1. Will present deployment at later stage
    3. Click **Finish** button

## NetBeans

## IntelliJ

## Command Line

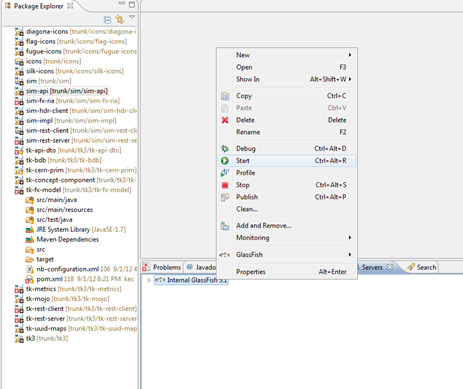
# Deploy Berkeley Database onto Server

<To Be Completed>

# Start Glassfish Server

## Eclipse

1. Go to server panel
2. Right-Click on Glassfish 4.0 Server
3. Select Start



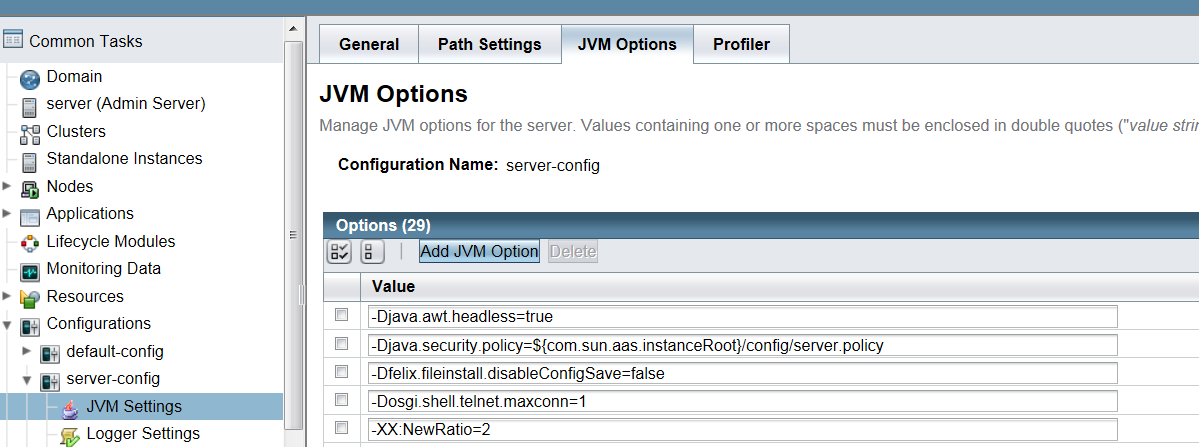
## NetBeans

## IntelliJ

## Command Line

# Configure Glassfish Properties

1. Goto Admin Console <http://localhost:4848/common/index.jsf>
2. Open **Configurations** in tree
3. Open **server-config** under Configurations
4. Select **JVM Settings under** server-config
5. Select **JVM Options** Tab



1. Change –XX:MaxPermSize variable to 512m
2. Change –Xmx variable to –Xmx1600m
3. Add JVM Option
   1. Select **Add JVM Option** button
   2. Add string “-XX:-UseGCOverheadLimit”
4. Save

# Deploy SIM Rest Server

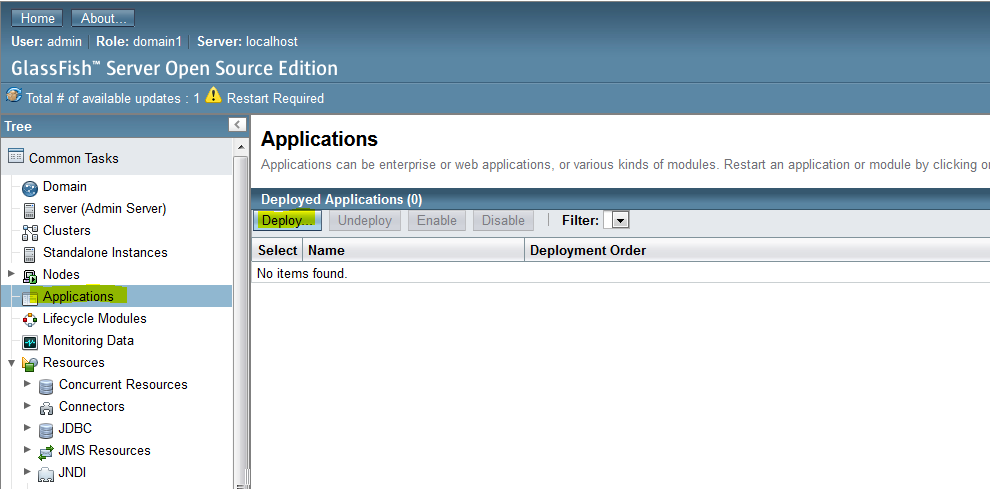
## Via AutoDeploy

1. Goto SIM/sim-rest-server/target folder
2. Copy sim-rest-server-1.2-SNAPSHOT.war file
3. Paste sim-rest-server-1.2-SNAPSHOT.war file to:

<Glassfish Installation Home>\ domains\domain1\autodeploy folder

## Via Glassfish Administrator Web Page

1. Goto Admin Console <http://localhost:4848/common/index.jsf>
2. Open **Applications** in tree
3. Select **Deploy** button



1. Select **Browse** button
2. Add the SIM/sim-rest-server/target/sim-rest-server-1.2-SNAPSHOT.war file
3. Select **OK** button

## Test Deployment

1. Open WebPage to: <http://localhost:8080/sim/>
2. Should be able to view:

