**SIRSi INSTALLATION INSTRUCTIONS**

Contents

[A. Eclipse Setup 2](#_Toc404676432)

[B. Chrome Setup 3](#_Toc404676433)

[C. Folders and Files to be downloaded from NCCD 3](#_Toc404676434)

[D. MIMIC Setup 3](#_Toc404676435)

[E. AllegroGraph Setup 4](#_Toc404676436)

[F. SIRSi Setup 4](#_Toc404676437)

[G. Tomcat Security Settings 10](#_Toc404676438)

[H. Tomcat Deployment 11](#_Toc404676439)

[APPENDICES 13](#_Toc404676440)

[App1. Modifying the Models 13](#_Toc404676441)

[App2. Run locally for development 14](#_Toc404676442)

[App3. Details on Setting Changes after Update Maven Project 16](#_Toc404676443)

## A. Eclipse Setup

|  |  |
| --- | --- |
| 1. Eclipse Kepler | Download ‘Eclipse Kepler SR2-IDE for Java EE Developers’ from  <http://www.eclipse.org/downloads/packages/release/Kepler/SR2>  Extract files |
| 2. Java Development Kit | Download JDK 1.7 from  <http://www.oracle.com/technetwork/java/javase/downloads/index.html>  Install |
| 3. Environment Variables | a. Add the ‘JAVA\_HOME’ Environment Variable:  *C:\Program Files (x86)\Java\jdk1.7.0\_xx*  b. Append JDK bin to the ‘Path’ variable  e.g. append: *;%JAVA\_HOME%\bin* |
| 4. GWT Plugin | a. In Eclipse, go to ‘Help→Install New Software’  b. In the ‘Work with’ text field, enter:  <http://dl.google.com/eclipse/plugin/4.3>  c. On the next selection window, select:  Google Plugin for Eclipse  SDKs |
| 5. Subversive | a. In Eclipse, go to ‘Help→Eclipse Marketplace’  b. In the ‘Find’ text field, enter;  *Subversive Team Provider*  c. On the next selection window, select:  Subversive SVN Team Provider |
| 6. SVN Connectors | a. In Eclipse, go to ‘Help→Install New Software’  b. In the ‘Work with’ text field, enter:  <http://community.polarion.com/projects/subversive/download/eclipse/3.0/kepler-site/>  c. On the next selection window, select:  Subversive SVN Connectors |
| 7. SVN Integrations | a. In Eclipse, go to ‘Help→Install New Software’  b. In the ‘Work with’ text field, enter:  <http://community.polarion.com/projects/subversive/download/integrations/kepler-site/>  c. On the next selection window, select:  Subversive Integration for the M2Eclipse Project |
| 8. SVN Connector Kit | a. In Eclipse, go to‘Help→Install New Software’  b. In the‘Work with’ text field, enter:  <http://community.polarion.com/projects/subversive/download/eclipse/3.0/kepler-site/>  c. On the next selection window, select:  Native JavaHL 1.7.9 for 64bit Windows  If not available, select any version and allow for Eclipse to compute alternative solution. Select whatever option becomes available. |

## 

## B. Chrome Setup

|  |  |
| --- | --- |
| 1. Chrome Setup | a. In the Chrome browser, click on the top rightmost icon  b. Select ‘Settings’  c. In the left panel, click on ‘Extensions’  d. Click on ‘Get more extensions’  e. Search for ‘GWT Developer Plugin’  f. If unavailable, download from  <http://www.gwtproject.org/missing-plugin/> |

## C. Folders and Files to be downloaded from NCCD

https://sbmi.uth.edu/nccd/

1. AllegroGraph file: sharp2a.trix

2. ‘.m2’ folder

of structure:

a. ‘com’ folder

i. ‘franz’ folder

b. ‘org’ folder

i. ‘sharpc\_backoffice’ folder

ii. ‘sharpc\_frontoffice’ folder

iii. ‘sharp2a’ folder

c. ‘SMArtClient’ folder

## D. MIMIC Setup

The physiologic dataset that was used in this project is the MIMIC Database.

|  |  |
| --- | --- |
| 1. Gaining Access | a. Follow the instructions for gaining access to the database  <http://physionet.org/mimic2/mimic2_access.shtml>  b. Once you have gained access to the website, follow the instructions for setting up the database in PostGreSQL.  You will need to download the MIMICII v25 files needed for this project. |
| 2. Username and Password | a. Create a username and password  b. At a later set-up stage, these will be entered into properties files. |

## E. AllegroGraph Setup

|  |  |
| --- | --- |
| 1. AllegroGraph Server | a. Follow the instructions at:  <http://franz.com/agraph/support/documentation/current/agraph-quick-start.html#aqs-intro>  to download and install AllegroGraph  b. Start the server by using a command similar to:  /home/[user]/agraph-4.14.1/bin/agraph-control --config /home/[user]/agraph-4.14.1/lib/agraph.cfg start |
| 2. AGWebView | a. Follow instructions at:  [http://franz.com/agraph/support/documentation/current/agraph-quick-start.html#aqs-webview-start](http://franz.com/agraph/support/documentation/current/agraph-quick-start.html%23aqs-webview-start)  to access the server.  b. Create a username and password  c. Create a new repository:  i. On the mainpage, type in a repository name, i.e. “*sharp2a*”  ii. Click on Create*.* |
| 3. Upload KnowledgeModel | a. In AGWebView, click on your newly created repository, ‘sharp2a’  b. Under ‘Import RDF’, click on ‘from an uploaded file’  c. Click on Choose Files  d. Navigate to your downloaded ‘sharp2a.trix’ file (downloaded in Section C above)  Click on Open  e. Click on OK in the bottom right corner of the top blue box |

## F. SIRSi Setup

|  |  |
| --- | --- |
| 1. Maven artifacts | a. Go to: ‘C:\Users\[*username*]\’  b. If there is no folder entitled ‘.m2’,  paste the previously downloaded folder ‘.m2’ (Section C above) into the ‘C:\Users\[*username*]\’ folder  If the ‘.m2’ folder is already present:  paste into ‘C:\Users\[*username*]\.m2\repository’:  ‘SMArtClient’  paste into the ‘C:\Users\[*username*]\.m2\repository\org’ folder:  ‘sharp2a’  ‘sharpc\_backoffice’  ‘sharpc\_frontoffice’  paste into the ‘C:\Users\[*username*]\.m2\repository\com’ folder:  ‘franz’ |

|  |  |
| --- | --- |
| 2. Allegrograph dependency | Downloading the jar  a. Go to:  <http://franz.com/agraph/allegrograph/clients.lhtml>  b. Go to first row in the table ‘Java - all platforms - Javadoc’  and click on ‘all platforms’  c. This will download the latest version of agraph, as of this document:  ‘agraph-4.14.1-client-java.tar.gz’  d. Extract the files  e. Navigate to the ‘lib’ folder and copy the ‘agraph-4.14.1.jar’ file  f. Paste the ‘agraph-4.14.1.jar’ file into the folder:  C:\Users\[*username*]\.m2\repository\com\franz\agraph\4.14.1 |
| 3. SmartClient dependency | Downloading the jar  a. Go to: <https://github.com/chb/smart_client_java>  b. On the right-hand side at the bottom, Click on Download ZIP  c. This will download the latest version of SmartClient, as of this document:  ‘smart\_client\_java-master.zip’  d. Extract files  Compile java files  One way to compile the java files to classes is by using Eclipse  a. Open new Eclipse workspace  b. Select ‘File→New→Project…’  c. In the ‘New Project’ dialog, select ‘Java Project’ and click on Next  d. In the ‘New Java Project’ dialog, enter a Project Name and click on Finish  e. From Windows Explorer, copy the folder:  C:\Users\[*username*]\Downloads\smart\_client\_java-master\src\org  f. In Eclipse, in the ‘Project Explorer’ tab, paste the copied folder into: ‘[*Project Name*]→src’  In the ‘File and Folder Operation’ dialog, select ‘Copy files and folders’  g. ‘Window→Show View→Navigator’  h. In ‘Navigator’ tab, navigate to ‘[*Project Name*]→bin→org’  This folder will contain the needed class files  i. Copy the ‘org’ folder  j. Paste the folder into Windows Explorer into some temporary location, e.g.  C:\Users\[*username*]\Downloads\smart\_client\_java-master\  k. Compress the ‘org’ folder  l. Rename the compressed folder to ‘SMArtClient-0.2.1.jar’.  m. Paste the ‘SMArtClient-0.2.1.jar’ file into the folder:  C:\Users\[*username*]\.m2\repository\SMArtClient\SMArtClient\0.2.1 |
| 4. SVN | a. Start Eclipse and open a new workspace  b. Select ‘Window→Open Perspective→Other…’  c. In the ‘Open Perspective’ dialog, select:  ‘SVN Repository Exploring’  Click on OK  d. Click on the ‘New Repository Location’ icon  e. In the ‘New Repository Location’ dialog, enter:  URL: *svn://epiphanet.uth.tmc.edu/svn/SIRSiFINNEGAN*  Click on Finish  f. In the ‘SVN Repositories’ tab, right-click on the repository ‘svn://epiphanet…’ and  select ‘Check out as Maven Project…’  g. In the ‘Checkout as Maven project from SCM’ dialog:  Click on Finish  Note: The checkout process will take some time, approximately 10 minutes  Note: There will be errors, but these will be cleared by later steps in the installation process |
| 5. Java Workspace Settings | a. In Eclipse, select ‘Window→Preferences’  b. In the ‘Preferences’ dialog, navigate to and select ‘Java→Installed JREs’  c. In the ‘Installed JREs’ dialog  Click on Add…  d. In the ‘Add JRE’ dialog, select ‘Standard VM’ and click on Next>  e. Click on Directory… and navigate to your JDK installation folder  likely C:\Program Files (x86)\Java\jdk1.7.0\_xx  Click on OK  f. Back in the ‘Add JRE’ dialog  Click on Finish  g. Back in the ‘Installed JREs’ dialog, check the box next to ‘jdk1.7.0\_xx’  jdk1.7.0\_xx  Click on OK |
| 6. Maven Install | a. Bring up the Navigator View by navigating to  ‘Window→Show View→Navigator’  b. Navigate to the ‘pom.xml’ file in ‘SIRSiBackOffice’  c. Right-click on the ‘pom.xml’ and select:  ‘Run As→3 Maven clean’  d. Upon completion, right-click on the same ‘pom.xml’ file and select:  ‘Run As→4 Maven generate-sources’  e. Upon completion, right-click on the same ‘pom.xml’ file and select:  ‘Run As→5 Maven install’ |
| 7. Update Maven Project | a. In Eclipse, in the ‘Project Explorer’ tab, right-click on any of the projects and select:  ‘Maven→Update Project…’  b. In the ‘Update Maven Project’ dialog, Select All available codebases  Leave the default settings as they are  Click on OK |
| 8. Reverting Settings | The update process alters some Eclipse settings on account of the particular folder structure that has been used. Files that have been altered are denoted with a preceding ‘>’ in the tree structure.  a. In the ‘Navigator’ tab, navigate to:  ‘KnowledgeEngineService→.settings→org.eclipse.wst.common.component’  b. Right-click on that file and select ‘Team→Revert’  c. In the ‘Revert’ dialog, click on OK  d. Likewise, in the ‘Navigator’ tab, navigate to  ‘KnowledgeEngineService→.settings→org.eclipse.wst.common.project.facet.core.xml’  Right-click on that file and select ‘Team→Revert’  In the ‘Revert’ dialog, click on OK  e. Likewise, perform ‘Team→Revert’ on the file ‘KnowledgeEngineService→.classpath’  f. Perform the same revert steps for the parallel files in the ‘pRECISE\_Data\_Engine’ project  ‘pRECISE\_Data\_Engine→.settings→org.eclipse.wst.common.component’  ‘pRECISE\_Data\_Engine→.settings→org.eclipse.wst.common.project.facet.core.xml’  ‘pRECISE\_Data\_Engine→.classpath’  Note: the specific changes to the files that have been made are denoted in Appendix 3. |
| 9. Allegrograph Properties | a. Click on ‘Search→File’ and search for the term *temporaryAg*  b. Result should yield two files:  DataImport/src/main/java/properties/i2b2.properties  and  KnowledgeEngineService/WebContent/WEB-INF/properties/knowledgeengine.properties  c. Replace the temporary placeholders in “i2b2.properties” with the Allegrograph parameters of your setup:  SERVER\_URL = *[http://[ipaddress of AllegroGraph server]:10035]*  CATALOG\_ID = *[catalog from Section E Step 2 above]*  REPOSITORY\_ID = *[repository from Section E Step 2 above]*  USERNAME = *[username from Section E Step 2 above]*  PASSWORD = *[password from Section E Step 2 above]*  for example:  SERVER\_URL = http://129.305.21.587:10035  CATALOG\_ID = system  REPOSITORY\_ID = sharp2a  USERNAME = sharp-user  PASSWORD = qjsiape  d. Set the same configuration properties in:  KnowledgeEngineService/WebContent/WEB-INF/properties/knowledgeengine.properties  for example:  TS\_SERVER\_URL = http://129.305.21.587:10035  TRIPLE\_STORE\_ID = sharp2a  CATALOG\_ID = system  USERNAME = sharp-user  PASSWORD = qjsiape |
| 10. MIMIC Properties | a. Click on ‘Search→File’ and search for the term *temporaryMimic*  b. Result should yield three files:  DataImport/src/main/java/properties/i2b2.properties  pRECISE\_Data\_Engine/WebContent/WEB-INF/classes/dataengine.properties  pRECISE\_Data\_Engine/WebContent/WEB-INF/properties/dataengine.properties  c. Replace the temporary placeholders in all files with the MIMC parameters of your setup:  serverName = *[ipaddress of Postgres server]*  portNumber = 5432  sid = MIMIC2  username = *[username from Section D Step 2 above]*  password = *[password from Section D Step 2 above]*  for example:  serverName = 179.58.249.376  portNumber = 5432  sid = MIMIC2  username = mimicUser  password = test1000 |
| 11. Migrate MIMIC data | a. Create a new Web Application run configuration by clicking on  ‘Run→Run configurations…’  b. In the tree in the left panel, double-click on ‘Java Application’  c. In the ‘Main’ tab, enter:  Name: *MigrateMimicData*  Project: *DataImport*  Main class: *test.testMigrateMimicData*  d. Click on Apply  e. Run MigrateMimicData |

|  |  |
| --- | --- |
| 12. Maven Install FrontOffice | a. Navigate to the file ‘pom.xml’ in the ‘SIRSiFrontOffice’ project  b. Right-click on the ‘pom.xml’ and select:  ‘Run As→4 Maven clean’  c. Upon completion, right-click on the same ‘pom.xml’ file and select:  ‘Run As→5 Maven generate-sources’  d. Upon completion, right-click on the same ‘pom.xml’ file and select:  ‘Run As→6 Maven generate-sources’ |
| 13. Update Maven Project | a. In Eclipse, in the ‘Navigator’ tab, right-click on the ‘SIRSiFrontOffice’ project and select:  ‘Maven→Update Project…’  Leave the default settings as they are  Click on OK |
| 14. Updating Settings | The update process alters some Eclipse settings on account of the particular folder structure that has been used  a. In the ‘Navigator’ view, navigate to and open:  ‘SIRSiFrontOffice→.settings→com.google.gdt.eclipse.core.prefs’  b. Right-click on that file and select ‘Team→Revert’  c. In the ‘Revert’ dialog, click on OK  d. Likewise, perform ‘Team→Revert’ on the file ‘SIRSiFrontOffice→.classpath’ |
| 15. Path for logging | a. Find and note the location of the ‘WebContent’ folder, e.g.  C:\Users\[*username*]\[*workspace*]\maven.*xxxxxxxxxx*\SIRSiFrontOffice\WebContent  b. In the ‘Navigator’ view, navigate to and open:  ‘SIRSiFrontOffice→Java Resources→src→org→sharpc→sirsi→server→RemoteClientLogger.java’  c. At line 53, change logging String path to your WebContent location, e.g.  *String path = “C:\\Users\\[username]\\[workspace]\\maven.xxxxxxxxxx\\SIRSiFrontOffice\\WebContent”;*  Note the need for double forward slashes |

## 

## G. Tomcat Security Settings

|  |  |
| --- | --- |
| 1. Download | a. Download Tomcat 7 from:  <http://tomcat.apache.org/download-70.cgi>  b. Extract the folder to, for example:  C:\Users\[*username*]\Documents\apache-tomcat-7.0.55 |
| 2. tomcat-users.xml | a. In your File Browser, navigate to and open for editing:  C:\Users\[*username*]\Documents\apache-tomcat-7.0.55\conf\tomcat-users.xml  b. Add:  *<role rolename="authenticated-user"/>*  *<user username="sirsiUser" password="sirsiPassword" roles="authenticated-user"/>*  c. Change “sirsiUser” and “sirsiPassword” to ones of your choosing  d. Save the file |
| 3. SIRSI Security properties | a. In Eclipse, navigate to:  ‘SIRSiFrontOffice→WebContent→WEB-INF→classes→SIRSiSecurity.properties’  b. Change the USERNAME and PASSWORD to be the ones you had chosen in Step 1 above  c. Save the file  d. If no such file exists, create a new file in that location, containing the following lines:  USERNAME = [username from Step 2 above]  PASSWORD = [password from Step 2 above]  e. Save the file |
| 4. Create keystore | a. Open a Windows Command terminal in Administrator mode by:  i. Opening to the Start Menu  ii. Selecting ‘All Programs’.  iii. Selecting ‘Accessories’  iv. Right-clicking on ‘Command Prompt’  v. Selecting ‘Run as Administrator’  b. Create a keystore by using a command of the form:  $JAVA\_HOME$\bin\keytool -genkey -alias localhost -keyalg RSA -keystore C:\users\[username]\keystore.jks  c. When prompted for first and last name, just use *localhost*  d. At the 2nd prompt for a password, hit [ENTER] to retain the same password as used in the earlier step |
| 5. Export certificate | a. In the Windows Command terminal, enter:  $JAVA\_HOME$\bin\keytool -export -alias localhost -file localhost.crt -keystore C:\users\[username]\keystore.jks |
| 6. Import certificate | a. In the Windows Command terminal in Administrator Mode, enter  $JAVA\_HOME$\bin\keytool -import -trustcacerts -file localhost.crt -alias localhost -keystore $JAVA\_HOME$\jre\lib\security\cacerts |

|  |  |
| --- | --- |
| 7. server.xml | a. In your File Browser, navigate to and open for editing:  C:\Users\[*username*]\Documents\apache-tomcat-7.0.55\conf\server.xml  b. Within <Host name …></Host>, comment out the line starting with:  *<Valve className="org.apache.catalina.valves…”*  c. Add the following line:  *<Valve className="org.apache.catalina.authenticator.SingleSignOn" />*  d. Within <Service name=”Catalina”></Service>, uncomment and add keystore information (from Step 4 above) to the  <Connector port =”8443” …> line:  In the form of:  *<Connector port="8443" protocol="org.apache.coyote.http11.Http11Protocol" sslProtocol="TLS" clientAuth="false"* ***keyAlias="localhost" keystorePass="[your password from ]" keystoreFile="[your keystore.jks location]"*** *secure="true" scheme="https" SSLEnabled="true" maxThreads="150"/>*  *e.g.:*  *<Connector port="8443" protocol="org.apache.coyote.http11.Http11Protocol" sslProtocol="TLS" clientAuth="false" keyAlias="localhost" keystorePass="ashygqit" keystoreFile="C:\keys\keystore.jks" secure="true" scheme="https" SSLEnabled="true" maxThreads="150"/>*  e. If you have an LDAP server, you can add the following line in <Realm></Realm>:  *<Realm className="org.apache.catalina.realm.JNDIRealm" connectionURL="ldaps://xxx.xxx.xxx.xxx:xxx" userBase="ou=PEOPLE,dc=zzz,dc=edu" userSearch="(uid={0})" userRoleName="uid" />*  f. This line can be commented out for when no internet connection is active, defaults to tomcat user file |

## H. Tomcat Deployment

|  |  |
| --- | --- |
| 1. Compiling FrontOffice | a. In Eclipse, right-click on the ‘SIRSiFrontOffice’ project folder and  Select ‘Google→GWT Compile’  b. In the ‘GWT Compile’ dialog:  Click on Compile  c. In the ‘WAR Directory Selection’ dialog, navigate to and select the following folder  ‘[workspace]→maven.xxxxxxxxxxxxx→SIRSiFrontOffice→WebContent’  Click on OK  d. Upon completion of the compilation, in the Navigator view, right-click on and copy the ‘WebContent’ folder  e. Paste the ‘WebContent’ folder into your Tomcat webapps folder, e.g.:  C:\Users\[*username*]\Documents\apache-tomcat-7.0.55\webapps  f. Rename the pasted ‘WebContent’ folder to be *SHARPC* |
| 2. Exporting kb | a. In Eclipse, right-click on the BackOffice ‘KnowledgeEngineService’ folder  Select ‘Export’  b. In the ‘Export’ dialog, select ‘Web→WAR file’  Click on Next  c. Next to the ‘Destination’ field, click on Browse…  d. In the ‘Save As’ dialog, navigate to  C:\Users\[*username*]\Documents\apache-tomcat-7.0.55\webapps  Change the filename to ‘kb.war’  Click on Save  e. Now back in the ‘Export’ dialog:  Click on Finish |
| 3. Exporting smartapp | a. In Eclipse, right-click on the BackOffice ‘pRECISE\_Data\_Engine’ folder  Select ‘Export’  b. In the ‘Export’ dialog, select ‘Web→WAR file’  Click on Next  c. Next to the ‘Destination’ field, click on Browse…  d. In the ‘Save As’ dialog, navigate to  C:\Users\[*username*]\Documents\apache-tomcat-7.0.55\webapps  Change the filename to ‘smartapp.war’  Click on Save  e. Now back in the ‘Export’ dialog:  Click on Finish |
| 4. Starting Tomcat | a. Open a Windows Command terminal and navigate to the folder:  C:\Users\[*username*]\Documents\apache-tomcat-7.0.55\webapps\bin  b. At the prompt, execute: *startup.bat*  c. A separate Tomcat window should open, should cycle through some text and end with:  ‘Server startup in xxxx ms’ |
| 5. Running SIRSi | a. Direct the Chrome browser to:  https://localhost:8443/SHARPC/SIRSI.html  b. A warning message will come up ‘Your connection is not private…’  i. Click on ‘Advanced’  ii. Then click on ‘Proceed to localhost’*.*  c. Password challenge popup should appear  These are the username and password that were selected in Section G Part 1 above and that were entered into ‘tomcat-users.xml’  d. Upon successful authentication, the patient selection screen should appear  e. If you would like to use the Standard Display (Vitals, Laboratory, etc.), check the box next to ‘Standard’ in the lower right corner  f. Select a patient by clicking anywhere on that patient’s row.  g. This will initiate preloading of data which will take approximately 1 minute.  To monitor loading, right-click on any blank space in the Chrome window,  Select ‘Inspect element’  Click on the ‘Network’ tab  Progress of the preloading can also be seen in the ‘Tomcat’ window.  The last data to be preloaded will be ‘INTUBATED’ |
| 6. Stopping Tomcat | a. Return to the Command Window and execute: *shutdown.bat* |

# APPENDICES

## App1. Modifying the Models

|  |  |
| --- | --- |
| 1. Run configurations | If model is modifies, these are the steps for uploading the model into the TripleStore  a. Create a new Web Application run configuration by clicking on  ‘Run→Run configurations…’  b. In the tree in the left panel, double-click on ‘Java Application’  c. In the ‘Main’ tab, enter:  Name: *KEModelUpload*  Project: *KnowledgeEngineService*  Main class: *precise.knowledge.utils.KEModelUpload*  d. In the ‘Arguments’ tab:, enter  Program Arguments:  *http://[IPAddress of AllegroGraph server] [AllegroGraph username] [AllegroGraph password] system sharp2a C:\Users\[username]\[workspace]\maven.xxxxxxxxxxx\PreciseEngine\src\main\java\cognitiveModel.owl*  e. Click on Apply  f. In the tree in the left panel, double-click on ‘Java Application’  g. In the ‘Main’ tab, enter:  Name: *MimicModelMapping*  Project: *KnowledgeEngineService*  Main class: *precise.knowledge.utils.MimicModelMapping*  h. In the ‘Arguments’ tab:, enter  Program Arguments:  *http://localhost:10035 [agraph username] [agraph password] system sharp2a C:\Users\[username]\[workspace]\maven.xxxxxxxxxxx\PreciseEngine\src\main\java\cognitiveModel.owl*  i. Click on Apply |
| 2. Uploads | a. Run KEModelUpload  b. Run MimicModelMapping |

## App2. Run locally for development

|  |  |
| --- | --- |
| 1. Adding Tomcat7 | Download Apache Tomcat  [http://archive.apache.org/dist/tomcat/tomcat-7/v7.0.55/bin/](http://http:/archive.apache.org/dist/tomcat/tomcat-7/v7.0.55/bin/)  Setup Tomcat in Eclipse  a. In Eclipse, click on the ‘Servers’ tab  b. Click on ‘No servers are available. Click this link …’  c. In the ‘New Server’ dialog, navigate to and select ‘Apache→Tomcat v7.0 Server’  Click on Next  d. Click on Browse… and navigate and select your Apache Tomcat7 folder,  e.g. C:\Users\[*username*]\Documents\apache-tomcat-7.0.55  Click on Next>  e. On the ‘Move Resources’ dialog, click on Add All>> , then click on Finish  f. In the ‘Servers’ tab, double-click on ‘Tomcat v7.0 Server at localhost’  Switch to the ‘Modules’ tab  Confirm that the Path for pRECISE\_Data\_Engine is: */kb*  Confirm that the Path for KnowledgeEngineService is: */smartapp*  g. In the ‘Servers’ tab, Right-Click on the ‘Tomcat v7.0 Server…’  and select ‘Start’ |
| 2. Test BackOffice Function | With the Allegrograph server running, the MIMIC server running, and the BackOffice running on Tomcat as described in Step 1 above, test the functioning of SIRSiBackOffice KnowledgeEngineService by directing a web browser to the following URLs:  <https://localhost:8443/kb/rest/kb/patientlist>  This should yield a patient list  <https://localhost:8443/kb/rest/kb/mimic/test/11686/Hemodynamic_instability_hypothesis>  This should yield a knowledge model for “Hemodynamic\_instability\_hypothesis”  Test the functioning of SIRSiBackOffice pRECISE\_Data\_Engine by directing a web browser to the URL:  <https://localhost:8443/smartapp/rest/patients/11686/observations/mimic/MONOS>  This should yield MONOS data for patient 11686 |
| 3. FrontOffice Run config | a. Create an new Web Application run configuration by clicking on  ‘Run→Run configurations…’  b. In the tree in the left panel, double-click on ‘Web Application’  c. In the ‘Main’ tab, enter:  Name: *SIRSi.html*  Project: *SIRSiFrontOffice*  Main class: *com.google.gwt.dev.DevMode*  d. In the ‘Arguments’ tab:, enter  Program Arguments:  *-war C:\Users\[username]\[workspace] \maven.xxxxxxxxxxx\SIRSiFrontOffice\target\SIRSiFrontOffice-1.0 -remoteUI "${gwt\_remote\_ui\_server\_port}:${unique\_id}" -logLevel INFO -codeServerPort 9997 -port 8888 org.sharpc.sirsi.SIRSI*  e. Click on Apply  f. Click on Close |
| 3. Running the application | a. Start the Allegrograph server running, the MIMIC server running, and the BackOffice running on Tomcat as described in Step 1 above  b. Set Chrome as the default browser in Eclipse  i. Select ‘Window→Preferences’  ii. Navigate to and select ‘General→Web Browser’  iii. Click on New  iv. Enter:  Name: *Chrome*  Location: Chrome application location, e.g.  *C:\Program Files (x86)\Google\Chrome\Application*  v. Select ‘Use external web browser’ and ‘Chrome’  c. Run the FrontOffice application  i. Select ‘Run→Configurations’  ii. In the filter tree on the right, expand the ‘Web Application’ node, and select ‘SIRSi.html’  iii. Click on Run  d. In the SIRSiFrontOffice workspace, In the ‘Development Mode’ tab,  double-click on the URL, e.g. ‘http://127.0.0.1:8888/SIRSi.html…’ |

## App3. Details on Setting Changes after Update Maven Project

|  |  |
| --- | --- |
| 1. KnowledgeEngine Settings | The Update Maven Project process alters some Eclipse settings on account of the particular folder structure that has been used  a. In the ‘Navigator’ tab, navigate to:  ‘KnowledgeEngineService→.settings→org.eclipse.wst.common.component’  b. Change Line 4 from:  *<wb-resource deploy-path="/" source-path="/target/m2e-wtp/web-resources"/>*  to  *<wb-resource deploy-path="/" source-path="/WebContent"/>*  c. Change Line 7 from:  *<property name="context-root" value="KnowledgeEngineService"/>*  to  *<property name="context-root" value="kb"/>*  d. Navigate to  ‘KnowledgeEngineService→.settings→org.eclipse.wst.common.project.facet.core.xml’  e. Change Line 9 from:  *<installed facet="jst.web" version="3.1"/>*  to  *<installed facet="jst.web" version="2.5"/>*  f. Navigate to ‘KnowledgeEngineService→.classpath’  g. Change Lines 11 to 15 from:  *<classpathentry kind="con" path="org.eclipse.jdt.launching.JRE\_CONTAINER/org.eclipse.jdt.internal.debug.ui.launcher.StandardVMType/JavaSE-1.7">*  *<attributes>*  *<attribute name="maven.pomderived" value="true"/>*  *</attributes>*  *</classpathentry>*  to  *<classpathentry kind="con" path="org.eclipse.jdt.launching.JRE\_CONTAINER"/>* |
| 2. pRECISE\_Engine Settings | a. In the ‘Navigator’ tab, navigate to:  ‘pRECISE\_Data\_Engine→.settings→org.eclipse.wst.common.component’  b. Change Line 4 from:  *<wb-resource deploy-path="/" source-path="/target/m2e-wtp/web-resources"/>*  to  *<wb-resource deploy-path="/" source-path="/WebContent"/>*  c. Change Line 7 from:  *<property name="context-root" value="pRECISE\_Data\_Engine"/>*  to  *<property name="context-root" value="smartapp"/>*  d. Navigate to  ‘pRECISE\_Data\_Engine→.settings→org.eclipse.wst.common.project.facet.core.xml’  e. Change Line 9 from:  *<installed facet="jst.web" version="3.1"/>*  to  *<installed facet="jst.web" version="2.5"/>*  f. Navigate to ‘pRECISE\_Data\_Engine→.classpath’  g. Change Lines 11 to 15 from:  *<classpathentry kind="con" path="org.eclipse.jdt.launching.JRE\_CONTAINER/org.eclipse.jdt.internal.debug.ui.launcher.StandardVMType/JavaSE-1.7">*  *<attributes>*  *<attribute name="maven.pomderived" value="true"/>*  *</attributes>*  *</classpathentry>*  to  *<classpathentry kind="con" path="org.eclipse.jdt.launching.JRE\_CONTAINER"/>* |
| 3. FrontOffice Settings | a. In the ‘Navigator’ view, navigate to and open:  ‘SIRSiFrontOffice→.settings→com.google.gdt.eclipse.core.prefs’  b. Delete the line that starts with ‘lastWarOutDir…’  c. Change the line  *warSrcDir=src/main/webapp*  to  *warSrcDir=WebContent*  d. Save and close the file.  e. Navigate to ‘SIRSiFrontOffice→.classpath’  f. Change Line 20 from:  *<classpathentry kind="output" path="target/classes"/>*  to  *<classpathentry kind="output" path="WebContent/WEB-INF/classes"/>* |