Measure Authoring Tool Latest Release Installation Guide

The Measure Authoring Tool (MAT) was designed using many open source products including the Google Web Toolkit (GWT) framework, Java JDK, MySQL and Eclipse. GWT allows a developer to write client side code in Java and where GWT converts it to JavaScript. The MAT uses MySQL as its backend database server and the IDE is Eclipse Mars.2 for Java EE Developers. Finally, the MAT is deployed on the Glassfish 3.1.2.2 application server.

Due to the use of these open source products, a working knowledge of Java development and some research into how the products work with each other in your environment may be necessary. Research from discussions on product forums, help documents, internet searches, and knowledge of the local environment where the MAT will be running may all need to be checked if there are errors during the install.

Installation Guidelines

1. Install Eclipse

1. Download Eclipse Mars.2 for Java EE Developers. Eclipse IDE is a free, open source IDE for writing Java applications found on the Eclipse download site: http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/mars2.

2. Select the version that best fits your operating system.

3. Extract the zip file to a location on the hard drive you would like Eclipse to be installed.

4. Run the application file.

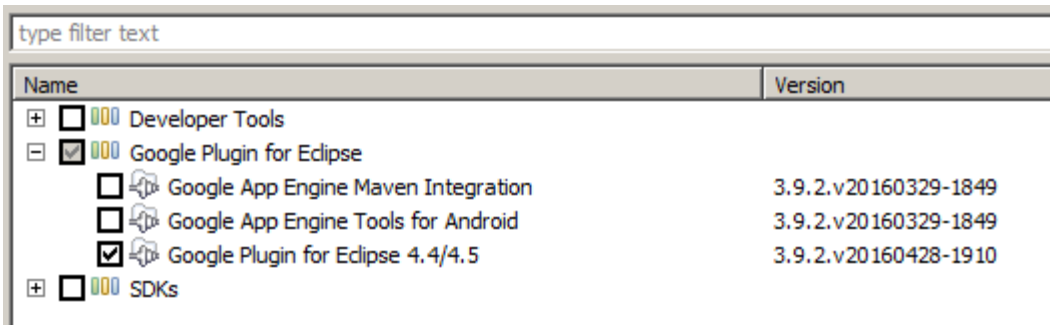
2. Install Google Web Toolkit (GWT)

Google Web Toolkit is a free, open source development toolkit used for developing complex browser based applications. The MAT has been designed and tested with GWT version 2.6.0.

Note: GWT can be installed and run through multiple methods; however, the MAT support team recommends using the Google Plugin for Eclipse along with GWT. Brief instructions on how to use the plugin can be found below. By using GWT through Eclipse, the user has the ability to write, compile, and run code with just one product.

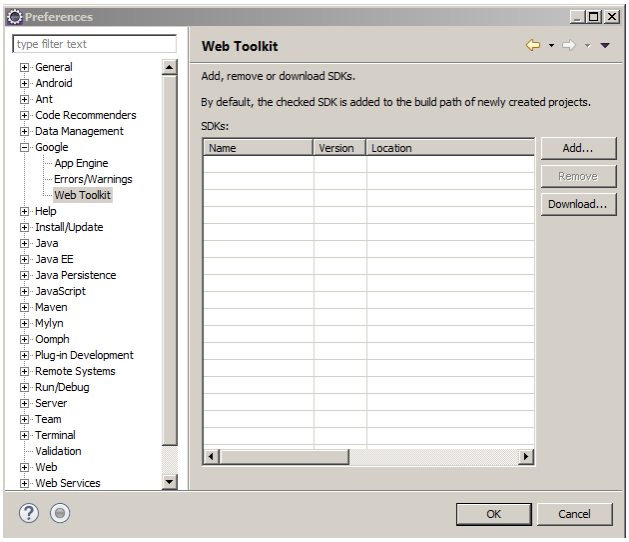
1. Install GWT Plugin for Eclipse.

1. Open Eclipse IDE, navigate to the workbench, and select **[Help] -> [Install New Software].**
2. In the **Work With** text box, past in the following URL: https://dl.google.com/eclipse/plugin/4.5 .
3. Expand **Google Plugin for Eclipse.** Select only. **Google Plugin for Eclipse 4.4/4.5.**



2. Install GWT SDK 2.6.0

1. Open a browser and navigate to http://www.gwtproject.org/versions.html.
2. Scroll and find Version 2.6.0 and download gwt-2.6.0.zip.
3. Extract the zip file to a location on the hard drive you would like GWT to be saved.
4. Open up Eclipse IDE, navigate to workbench and select **[Window] -> [Preferences] -> [Google] -> [Web Toolkit].**



5. Click the add button, click browse, then navigate to where you saved the gwt-2.6.0 folder, and select gwt-2.6.0 folder, press the OK Button.
6. If adding was successful, you should now see the SDK show up in the list. If it's in the list, press the OK button.

3. Configure Java (JDK)

The application has **not** been tested with version above Java 1.8. Please ensure this version is in the environment.

Verify that JAVA_HOME and PATH system variables are pointing to the proper folder(s).

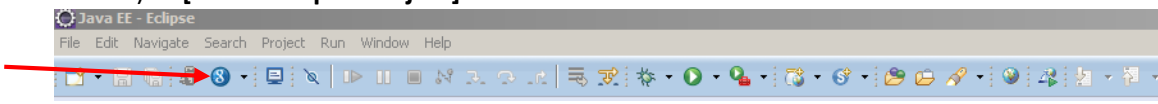
For example, in a Windows environment, the JAVA_HOME (Environment Variables under Advanced System Settings) should point to the Java DK 1.8.x folder and PATH should point to the Java 1.8.x/bin.

4. MAT Code Base Import

1. From the MAT Github source code link, extract the code base into the Eclipse workspace folder in the MAT Environment.
2. Import the code base into an Eclipse project:
   [File] -> [Import] -> [General] -> [Existing Project into Workspace] -> Browse to and Select << workspace >> -> **[Finish]**

5. Compile the MAT Code

1. In eclipse, select Google Services and Development Tools (Google Icon Button) **-> [GWT Compile Project].**



2. In the GWT Compile Wizard:
   1. Browse to find the MAT Project.
   2. Set the Log Level to Debug.
   3. Add the Login and Mat Entry Point Modules.

6. Create MAT Database

1. Install MySQL (MAT currently uses MySQL Community Version 5.5) available from MySQL.com: http://dev.mysql.com/downloads/installer/5.5.html.
2. Run the MySQL community server 5.5 installer for your operating system and the MySQL workbench (which comes with the download).
3. Create a new MySQL Connection to localhost.
   Note: Make sure to keep track of the username and password you used as you will need this later.
4. From the MAT source code in the Eclipse Workspace folder, navigate to the sql folder, and extract the 'Blank DB Scripts.zip' folder and extract it. This folder contains the .sql files to create the initial MAT database. The order the files need to be executed in is located in the 'ReadMe.txt' file. Open and execute each .sql file in the order 'ReadMe.txt' specifies.

7. For App Server Use (Optional)

Note: If MAT is to be run on an application server, the developer will need to run the build to create a .war file.

1. In the Eclipse project explorer, navigate to 'build.xml'.
2. Right-Click and **[Run As] -> [Ant Build].**
3. After the build has run, the file is placed into an artifacts folder under MAT as follows: mat/work/artifacts/MeasureAuthoringTool.war.

8. Configure MAT Files for Local Development Environment

Changes will need to be made to some of the files in the code base to allow the MAT to connect to the local MySQL Database.

1. mat/war/WEB-INF/mat-persistence.xml

1. In this file, set the 'dataSource' bean to provide login credential to the local MySQL database. The box below is an example of the mat-persistence.xml file. The username and password values indicated with '**username' and 'password' (bolded)** need to be changed to the "username" and "password" you used in the previous step to set up your local database.

```
<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">
        <property name="driverClassName" value="com.mysql.jdbc.Driver"/>
        <property name="url" value="jdbc:mysql://localhost:3306/MAT_APP"></property>
        <property name="username" value="username"></property>
        <property name="password" value="password"></property>
</bean>
```

2. mat/war/WEB-INF/applicationContext-security.xml

1. Make the **default target URL (bolded)** changes in the box below in the applicationContext-security.xml file.

```
<http use-expressions="true">
        <intercept-url pattern="/Mat.html" access="isAuthenticated()" />
        <intercept-url pattern="/mat/**" access="isAuthenticated()" />
        <form-login default-target-url="/Mat.html?gwt.codesvr=127.0.0.1:9997" login-page="/Login.html?gwt.codesvr=127.0.0.1:9997"/>
        <logout />
        <session-management invalid-session-url="/Login.html">
                <concurrency-control max-sessions="1" error-if-maximum-exceeded="true" />
        </session-management>
</http>
```

3. mat/war/WEB-INF/applicationContext-mail.xml

1. Make the **host value property name (bolded)** changes in the box below to the 'mailSender' bean to point to the proper mail server.

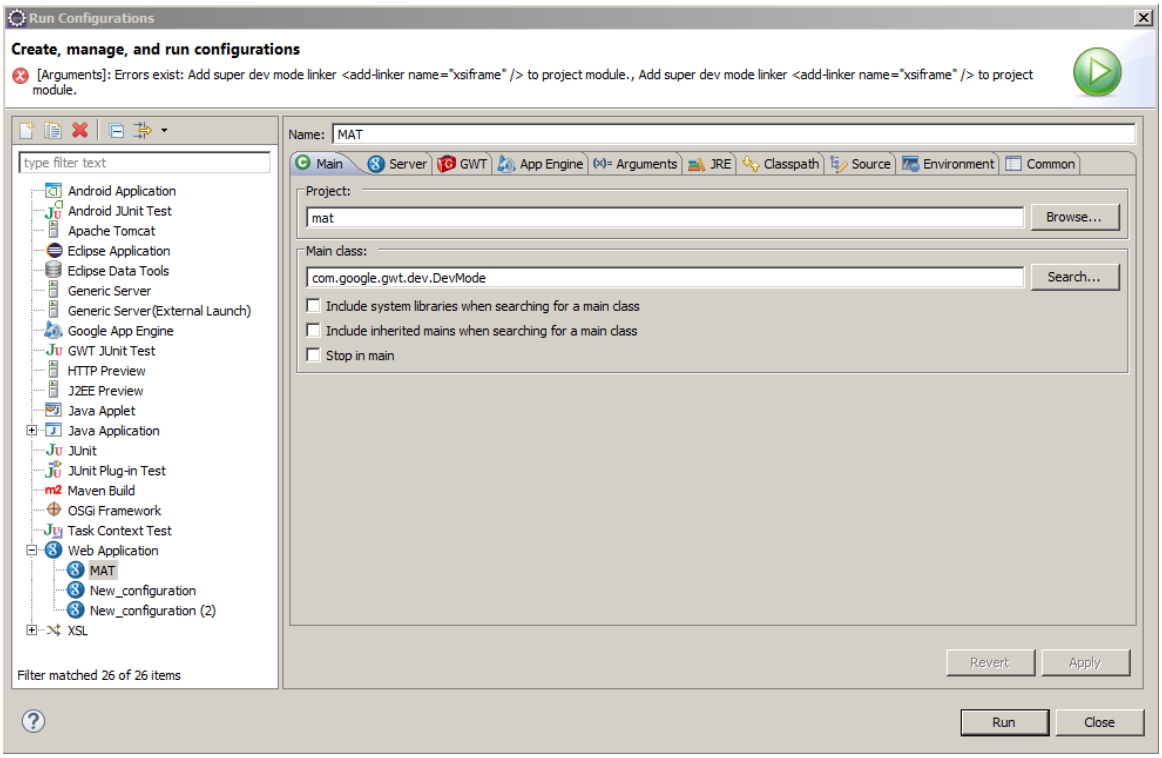
```
<bean id="mailSender" class="org.springframework.mail.javamail.JavaMailSenderImpl">
        <property name="host" value="mail.some-server.org"/>
</bean>
```

2. Make the **from value property name (bolded)** changes in the box below to the 'templateMessage' bean to set the correct 'From' email address.

```
<bean id="templateMessage" class="org.springframework.mail.SimpleMailMessage">
        <property name="from" value="someaddr@someaddr.org"/>
</bean>
```

9. Run Mat

1. Make sure your database server is running.
2. In the eclipse IDE Select,
   [Run] -> [Run Configurations] -> [Web Application] -> [New].
3. Set Main Class: **"**com.google.gwt.dev.DevMode**".**
4. Select **[Apply]** and followed by **[Run].**



10. Logging in To Mat

1. To login to MAT, open MySQL Workbench and run the following queries:
    1. **SELECT * FROM USER where USER_ID='Admin'** and
       Look at the **LOGIN_ID** column, the value there is your UserID.
       The password default is **'Ursaminor_10'.**
2. Navigate to the MAT log in page GUI and use the UserID and password from the previous step and log in to MAT. In the Security Code box, just enter random numbers.
3. Once logged in, navigate to the **[Mat Account]** tab and enter the Admin user details under the **[Personal Information]** tab and the **[Security Questions]** tab to setup user's security questions.
4. To change the password to something new, use the **[Password]** tab.

Note: This is an admin user login which provides a user the ability to create a regular user (one that can create, edit, and delete measures).

11. Setting up VSAC Communication Parameters for use with new MAT

MAT uses RESTful web-services to connect to VSAC system to pull in element lookup data. To set up MAT to connect with the VSAC, specify the following VM arguments in Eclipse by doing the following:

1. **[Run] -> [Run Configurations]** -> Select your MAT project on the LHS and RHS on the **[Arguments]** tab -> add the following information below into the VM arguments box.

```
-Xmx512m
-Dvsac_proxy_host=<<your proxy host>>
-Dvsac_proxy_port=8080 -DSERVER_TICKET_URL=https://vsac.nlm.nih.gov/vsac/ws/Ticket
-DSERVER_SINGLE_VALUESET_URL=https://vsac.nlm.nih.gov/vsac/ws/RetrieveValueSet?
-DSERVER_MULTIPLE_VALUESET_URL=https://vsac.nlm.nih.gov/vsac/ws/RetrieveMultipleValueSets?
-DSERVICE_URL=http://umlsks.nlm.nih.gov
-DENVIRONMENT=DEV
-D2FA_AUTH_CLASS=mat.server.twofactorauth.DefaultOTPValidatorForUser
```