# Measure Authoring Tool Installation

The Measure Authoring Tool (MAT) utilizes the Google Web Toolkit (GWT) framework. GWT allows one to write client side code in Java and then converts it to JavaScript while running. Also, the MAT uses MySQL as its backend database server. The IDE used by the MAT development team is Eclipse 3.6 (Helios). Finally, the MAT is deployed on the Glassfish 3.1.1 application server.

The sections below describe each of the above mentioned components in detail.

Java

Please ensure that you have Java 6 (JDK) on your machine for successfully running the MAT. Also, verify that JAVA\_HOME and PATH system variables are pointing to the Java 6 JDK. The application has not been tested with any newer version above JAVA 6 and hence the application is not guaranteed to work.

Eclipse

Eclipse IDE is a free, open source IDE for writing Java applications. Download Eclipse 3.6 (Helios) for Java EE developers from the Eclipse download site at <http://www.eclipse.org/downloads/index-helios.php> . There is no installation involved. You download a zip file and extract it to wherever you want Eclipse to reside on your hard drive.

Google Web Toolkit (GWT)

Google Web Toolkit is a free, open-source development toolkit used for developing complex browser based applications. More about GWT can be found at <https://developers.google.com/web-toolkit/>. MAT has recently upgraded to GWT version 2.5.1 currently.

Instead of installing GWT separately, we recommend using the Google Plugin for Eclipse which allows you to write, compile and run your code all through Eclipse.

After starting Eclipse IDE, navigate to the workbench and select,

Help 🡪 Install New Software

Use the following URL:

For Google Web Toolkit SDK, use the following URL: <http://dl.google.com/eclipse/plugin/3.6>

For Google Plugin for Eclipse 3.6, please refer to the following page for installation instructions.

<https://developers.google.com/eclipse/docs/download_older>

Please remember that we need plugin for Eclipse 3.6. If you are having trouble downloading the plugin direct from Eclipse, please see the section

“Installing from a local update archive” on the page.

Google App Engine content isn’t needed.

Currently MAT requires version 2.5.1 for successful compilation. This version is downloaded from:

<https://developers.google.com/web-toolkit/versions>

Extract the version 2.5.1 to a folder of your choice.

Go back to Eclipse and select,

Window 🡪Preferences 🡪 Google 🡪 Web Toolkit 🡪 Add

Browse to the folder where you previously extracted GWT 2.5.1 and select this version.

MySQL

MAT currently uses MySQL community version 5.5. Download MySQL community server 5.5 installer for your Operating System. Installing MySQL workbench (which comes with the download) can be useful.

Take note of the MySQL username and password when installing.

Under the ‘mat’ folder find ‘Blank DB Scripts.zip’ and extract it. This file will contain the .sql files to create an initial database. Within this zip is also a file called ‘ReadMe.txt’ which will indicate the order in which you would need to run the .sql files.

Code Base

Extract the code base into your Eclipse workspace folder.

Import the code base into an Eclipse project,

File 🡪 Import 🡪 General 🡪 Existing Project Into Workspace 🡪 Browse to and Select <<workspace>> 🡪 Finish

Compile the code,

Google Services and Development Tools (Google Icon button) 🡪 GWT Compile Project

In the GWT Compile wizard, add the entry point modules Login and Mat

Select log level as Debug and click on compile. The project should be successfully compiled.

Run the build to create a war file:

build.xml (right click and Run As 🡪 Ant Build)

After the build has run, the war file will be placed in,

mat/work/artifacts/MeasureAuthoringTool.war

Setting up your development environment

You need to make some changes to some of the files in the code base so that MAT will connect to your local MySQL db.

Go to mat/war/WEB-INF/mat-persistence.xml

In this file set the “dataSource” bean to point to your local MySQL 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=*"****<<your db user name>>****"*></property>

<property name=*"password"* value=*"****<<your db password>>****"*></property>

</bean>

Next go to,

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

Make the change as marked in black below.

<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>

Next go to,

mat/war/WEB-INF /applicationContext-mail

Make changes to the ‘mailSender’ bean to point to proper mail server and then to the ‘templateMessage’ bean to set the correct From address.

Run MAT

Make sure that your database server is running.

In the Eclipse IDE select,

Run 🡪 Run Configurations 🡪 Web Application 🡪 New

Set Main class:

com.google.gwt.dev.DevMode

Select Apply and then Run.

Logging In to MAT

For logging in to MAT, you will need to get onto MySQL and run the following queries,

SELECT \* FROM USER where USER\_ID='Admin';

Look at the LOGIN\_ID column, the value there is your User Id.

The password for this user is already set as ‘Ursaminor\_10’.

Use the User ID and password on the MAT login screen and log on to MAT.

Once logged in, go to the “MAT Account” tab and enter other your details under the “Personal Information” tab.

Go to the “Security Questions” tab and setup your security questions.

Also if you want to set your password to something new, use the “Password” tab.

This is an admin user which will allow you to create a regular user (one that can create,edit, delete measures).

Setting up VSAC communication parameters

MAT uses RESTful web-services to connect to VSAC system to pull in element lookup data. To set up MAT to work correctly

and connect with the VSAC, we need to specify the following VM arguments.

-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

This can be set in the Dev Eclipse by doing the following,

Run -> Run Configurations -> Select your MAT project on the LHS and then on the RHS on the 'Arguments' tab -> Add the above in the VM arguments box.