**Steps for Running DME Server and Web on Local Machine**

# Setup the DME project using the following steps:

# Clone the code from HPC\_DME\_APIs repository on GitHub to your local machine.

git clone https://github.com/CBIIT/HPC\_DME\_APIs.git

The above step clones the master branch.

If a specific branch needs to be checkout out, use the following command:

git checkout <branch\_name>

e.g.: git checkout releases/3.0.0

To know the branch you are on, use the following command:

git branch

*From Eclipse, you can use the Import project from Git (Clone URI)*

*We need to create a development access token from Github and use username/token to connect to the repository.*

# Set HPC\_HOME environment variable to the <Path to HPC\_DME\_APIs> directory path in the source tree.

e.g. export HPC\_HOME=/Users/dinhys/dev/HPC\_DME\_APIs

# Install JDK 11:

https://www.oracle.com/java/technologies/javase/jdk11-archive-downloads.html

Set JAVA\_HOME environment variable accordingly.

You can add this to ~/.bash\_profile or ~/.profile so it takes effect every time e.g.:

export JAVA\_HOME=/Library/Java/JavaVirtualMachines/jdk-11.jdk/Contents/Home/

# Install Maven

Install Maven 3.9.6:

https://maven.apache.org/download.cgi  
  
Set MAVEN\_HOME environment variable accordingly. For e.g.

export MAVEN\_HOME=/Users/dinhys/dev/apache-maven-3.9.6

# Build the hpc-server and hpc-web war files using the following command:

# Build the hpc-server

cd $HPC\_HOME/src  
mvn clean install

# Make the following changes to the DME Web configuration file located at:

$HPC\_HOME/src/hpc-web/src/main/resources/appconfigs/local-env.conf

* Add values for the Google Drive client id and secret.

gov.nih.nci.hpc.drive.clientid=< Configure Me >

gov.nih.nci.hpc.drive.clientsecret=< Configure Me >

* Point the property gov.nih.nci.hpc.server to the URL of the appropriate DME Dev Server e.g.  
  gov.nih.nci.hpc.server=https://fsdsgl- dmeap02d.ncifcrf.gov:7738/hpc-server

# Build the hpc-web

cd $HPC\_HOME/src/hpc-web  
mvn clean install -DskipTests

# Install and Configure Tomcat

# Install Tomcat

Install Apache Tomcat 9.x:  
<https://tomcat.apache.org/download-90.cgi>

Set CATALINA\_HOME environment variable accordingly. For e.g.

export CATALINA\_HOME=/Users/dinhys/dev/apache-tomcat-9.0.84

# Configure Tomcat

To allow the server to communicate with iRODS and LDAP, we need to deploy/configure a keystore:

* Copy keystore and cacerts into $CATALINA\_HOME/conf/hpc-server

mkdir $CATALINA\_HOME/conf/hpc-server

cp $HPC\_HOME/src/hpc-server/hpc-ws-rs-impl/src/main/resources/cacerts.jks $CATALINA\_HOME/conf/hpc-server

cp $HPC\_HOME/src/hpc-server/hpc-ws-rs-impl/src/main/resources/keystore.jks $CATALINA\_HOME/conf/hpc-server

* Create file $CATALINA\_HOME/bin/setenv.sh and add the following:

JAVA\_OPTS="$JAVA\_OPTS -Djavax.net.ssl.trustStore=$CATALINA\_HOME/conf/hpc-server/cacerts.jks -Djavax.net.ssl.trustStorePassword=changeit"

* Copy properties file into $CATALINA\_HOME/conf/hpc-server

cp $HPC\_HOME/src/hpc-server/hpc-ws-rs-impl/src/main/resources/WEB-INF/spring/hpc-server.properties $CATALINA\_HOME/conf/hpc-server

* Update $CATALINA\_HOME/conf/hpc-server/hpc-server.properties as follows:

1. Set the hpc.bus.aspect.systemAdministratorUserId to be your nih username
2. Set the hpc.integration.ldap.password.
3. Set hpc.dao.oracle.password.
4. Check to see if you are pointed to the Dev Oracle instance

*Ensure that you have connectivity to the database. If not, it could be that you are not in the right VPN group.*

1. Check to see if you are pointed to the Dev iRODS instance

*You will need an iRODS account in the DEV environment. Ask a team member to register you as an HPC-DM user in the DEV environment. The registration in DEV will create your iRODS user.*

* Configure the SSL port 7738 for DME API Server

To enable SSL on port 7738, add the following connector to $CATALINA\_HOME/conf/server.xml:

<Connector protocol="org.apache.coyote.http11.Http11NioProtocol" port="7738" maxThreads="200" maxParameterCount="1000" sslEnabledProtocols="TLSv1.2" scheme="https" secure="true" SSLEnabled="true" keystoreFile="${catalina.base}/conf/hpc-server/keystore.jks" keystorePass="changeit" clientAuth="false" sslProtocol="TLS"/>

* To access the Web application directly at localhost:8080, add the following context in $CATALINA\_HOME/conf/server.xml.

<Context path="" docBase="hpc-web-<version>" privileged="true"/>

* To access the hpc-server at path /hpc-server, add the following context in $CATALINA\_HOME/conf/server.xml.

<Context path="/hpc-server" docBase="hpc-server-<version>" privileged="true"/>

# Deploy the war files to Tomcat

# Copy the war files into the webapps folder

* Copy the DME API Server war file

cp $HPC\_HOME/src/hpc-server/hpc-ws-rs-impl/target/hpc-server-<version>.war $CATALINA\_HOME/webapps

* Copy the DME Web Application war file

cp $HPC\_HOME/src/hpc-web/target/hpc-web-<version>.war $CATALINA\_HOME/webapps

* Optionally (please see note below first), copy the scheduler war files to Tomcat

cp $HPC\_HOME/src/hpc-server/hpc-scheduler/target/hpc-scheduler-<version>.war $CATALINA\_HOME/webapps

cp $HPC\_HOME/src/hpc-server/hpc-scheduler-migration/target/hpc-scheduler-migration-<version>.war $CATALINA\_HOME/webapps

***Important Note****: In a local development environment, we typically skip the 3rd step above (scheduler install) since we share the same iRODS server with DEV and the scheduler is running there. Having 2 schedulers running (pointing to the same DME DB) will cause issues in processing the asynchronous tasks.*

# Run tomcat and verify

* Run Tomcat from the console

$CATALINA\_HOME/bin/startup.sh

DME API Services are deployed to https://localhost:7738/hpc-server/<hpc-service>

This is the URL to be used for accessing the services from curl or SOAP-UI.

*For some versions of open ssl, you will need to add curl --ciphers ALL option when accessing the URL above.*

* Verify API Server using the authentication endpoint

curl -k -u <username> https://localhost:7738/hpc-server/authenticate

* Verify DME Web Application from the browser  
  http://localhost:8080/

# [Optional] Deploy the war files using eclipse IDE

* In the Servers View, add New->Server Tomcat v9.0 Server at localhost and select the Tomcat installation directory from Step 5.1
* Double click on the Tomcat v9.0 Server at localhost and select “Use Tomcat installation (takes control of Tomcat installation) for Server Location and save
* Right click on the server and click Add and Remove… to deploy the war files
* In the Project Explorer, open Servers/Tomcat v9.0 Server at localhost/server.xml add:  
  <Connector protocol="org.apache.coyote.http11.Http11NioProtocol" port="7738" maxThreads="200" maxParameterCount="1000" sslEnabledProtocols="TLSv1.2" scheme="https" secure="true" SSLEnabled="true" keystoreFile="${catalina.base}/conf/hpc-server/keystore.jks" keystorePass="changeit" clientAuth="false" sslProtocol="TLS"/>

Also set hpc-web path and hpc-server path to “” and “/hpc-server” respectively:  
<Context docBase="hpc-web" path=""…>

<Context docBase="hpc-ws-rs-impl" path="/hpc-server"…>

* In the Project Explorer, open ServersàRun asàRun Configurations…

Select Apache TomcatàTomcat v9.0 Server at localhost and open the Arguments tab. In the VM arguments, add the following:

-Djavax.net.ssl.trustStore="<path defined in $CATALINA\_HOME>/conf/hpc-server/cacerts.jks" -Djavax.net.ssl.trustStorePassword="changeit"

* Select hpc-web and hpc-ws-rs-impl (hpc-server) and click Add>
* In the Project Explorer, right click on Servers/Tomcat v9.0 Server at localhost/server.xml
* Start the server and verify access

curl -k -u <username> https://localhost:7738/hpc-server/authenticate

http://localhost:8080/

# [Optional] Start the DME Web application as standalone

The DME Web application is a Springboot application, and it can run with the embedded tomcat. To start the application:

cd $HPC\_HOME/src/hpc-web

mvn spring-boot:run

The URL of the local web client is

<https://localhost:9080>

*To start the Springboot app using eclipse, Right click on the hpc-web project and click Run As à Java application and select HpcWebApplication (which is the main class).*