*HPC Data MANAGEMENT*

DEVELOPMENT ENVIRONMENT SETUP

Version *1.3*

*06/12/2017*

**Version History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Number** | **Implemented**  **By** | **Revision**  **Date** | **Description of Change** |
| 1.0 | Eran Rosenberg | *5/17/2015* | Initial Draft |
| 1.1 | Eran Rosenberg | 8/17/2015 | Create local Keystore |
| 1.2 | Eran Rosenberg | 1/1/2017 | Updates after lib/tools upgrades |
| 1.3 | Eran Rosenberg | 6/12/2017 | Configure servicemix w/ Keystore + Git repo |

TABLE OF CONTENTS

Git Code Repository 4

JDK 4

MAVEN 4

SERVICEMIX 4

PostgreSQL DB 5

IRODS 5

PATH 5

BUILD 5

DEPLOY to SERVICEMIX 5

ENDPOINTS 6

SOAP-UI 6

# Git Code Repository

Check out the source code from Git:

Repository URL: <https://github.com/CBIIT/HPC_DME_APIs>

Branch name: master

Set HPC\_HOME environment variable to the ‘hpc/src’ directory path in the source tree.

e.g /Development/HPC-DM/src/hpc

# JDK

Install JDK 1.8.x: <http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html>

Set JAVA\_HOME environment variable accordingly.

# MAVEN

Install Maven 3.3.9: <http://maven.apache.org/download.cgi>

Set MAVEN\_HOME environment variable accordingly.

# SERVICEMIX

Install Servicemix 7.0.0.: <http://servicemix.apache.org/downloads.html>

Set SERVICE\_MIX\_HOME environment variable accordingly.

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

1. Obtain the keystore file used in the DEV environment (file name is hpc-keystore.jks).
2. Copy the keystore file to $SERVICE\_MIX\_HOME/etc
3. Add the following to the end of $SERVICE\_MIX\_HOME/etc/system.properties

# HPC-DM keystore

javax.net.ssl.keyStore=${karaf.home}/etc/hpc-keystore.jks

javax.net.ssl.keyStorePassword=hpc-server-store-pwd

javax.net.ssl.trustStore=${karaf.home}/etc/hpc-keystore.jks

javax.net.ssl.trustStorePassword=hpc-server-store-pwd

# PostgreSQL DB

Install PostgreSQL DB 9.6.3 <https://www.postgresql.org/download/>

1. Execute all SQL scripts in $HPC\_HOME/hpc-server/hpc-dao-impl/src/main/scripts/schema **except** hpc\_hierarchical\_metadata.sql (this script is depending on having iRODS installed which we don’t have on local development environment). Please note that the hpc\_hierarchical\_metadata\_local\_dev\_env.sql will create these hierarchical views as foreign tables in your local DB.
2. Execute SQL scripts intended for DEV environment (they end with ‘dev’ in the script name’) in $HPC\_HOME/hpc-server/hpc-dao-impl/src/main/scripts/restore.
3. Manually insert a row into HPC\_USER table using your info (NIH user-id, first-name, last-name, etc)

# IRODS

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. Use your NIH user-id and first/last name. The registration in DEV will create your iRODS user.

# PATH

Set your $PATH environment variable

PATH=$MAVEN\_HOME/bin:$SERVICE\_MIX\_HOME/bin:$PATH

# BUILD

cd $HPC\_HOME

mvn clean install

# DEPLOY to SERVICEMIX

Run Servicemix console: servicemix

Install hpc-server (in Servicemix console):

features:repo-add mvn:gov.nih.nci.hpc/hpc-features/1.0.0-SNAPSHOT/xml/features

feature:install hpc-server-rest-services

feature:install hpc-server-scheduler

Note: in a local development environment, we typically don’t run the scheduler since we share the same iRODS server with DEV and the scheduler is running there. Having 2 scheduler running (pointing to the same iRODS but having different HPC DB) will cause issues in the async upload of data objects.

Also – you can start servicemix in a debug mode by ‘servicemix debug’, and then attach a debugger to a remote Java application on port 5005.

# ENDPOINTS

HPC DM Services are deployed to [https://localhost:7338/hpc-server/<hpc-service](https://localhost:7338/hpc-server/%3chpc-service)>

# SOAP-UI

Soap UI workspace and project can be found under $HPC\_HOME/hpc-soap-ui folder.

Note – after pointing soap-ui to the workspace, you will need to correct the path to the project based on the directory you pulled the code into.