Building Apromore

Apromore Build Instructions with Eclipse Virgo

REQUIREMENTS

Apromore has been successfully installed on the following platforms:
- Windows 7
- Linux (specifically ubuntu only when running OracleJVM)
- Mac OSX 10.8 to 10.11

Ensure the following software is installed:
- Java 1.7
- Maven 3.3.9
- Ant 1.9.2
- Git 2.3.5

Additional software required for PQL support:
- MySQL 5.6.x
- LoLA 2.0 (http://service-technology.org/lola/)

Check out the Apromore source tree from the GitHub repository at https://github.com/apromore/ApromoreCode

CONFIGURATION

Almost all configuration occurs in the top level site.properties file. The default version of this file from a fresh git checkout contains reasonable defaults that use H2 as the main database, but disable PQL (which requires MySQL or Postgres and more intricate configuration).

H2 running from a flat file is the default database for the sake of zero-configuration.
However our development is done chiefly on MySQL; instructions for reconfiguring Apromore to use MySQL appear below.
We do have plugins for Postgres and Oracle, but some extra setup will be required since we only have sql scripts to create the database for H2 and MySQL.

Some of Apromore's features are implemented as Java applets running client-side in the browser. If you possess an code-signing certificate (not an SSL certificate), you can edit the top-level codesigning.properties file to use your certificate rather than the self-signed certificate included in the source tree. This will avoid browser warnings.

DATABASE SETUP

Ensure MySQL is configured to accept local TCP connections on port 3306 in its .cnf file; "skip-networking" should not be present.

Start MySQL
$ sudo mysqld_safe

Set the root password of MySQL to the default used by Apromore
$ mysqladmin -u root password MAcri

Create a database named 'apromore' in your MySQL server
$ mysqladmin --user=root --password=MAcri create apromore

Create a user named 'apromore' with the required permissions
$ mysql --user=root --password=MAcri
CREATE USER 'apromore'@'localhost' IDENTIFIED BY 'MAcri';
GRANT SELECT, INSERT, UPDATE, DELETE, LOCK TABLES, EXECUTE, SHOW VIEW ON apromore.* TO 'apromore'@'localhost';

Create and populate the database tables.
$ mysql --user=apromore --password=MAcri < Supplements/database/db-mysql.sql

At the end of the db-mysql.sql script is where we populate some of the system data including user information. Currently, we have a few users setup that are developers or affiliates and they can be used or you can choose to add your own. All passwords are encrypted but they

are 'password' for the time being and we don't have a facility just yet to allow a user to change their password. This is coming soon as well as a setup utility to allow the creation of a user on first running.

Edit the top-level site.properties file, replacing the H2 declarations in "Database and JPA" with the commented-out MySQL properties.
Stop and restart the server so that it picks up the changes to site.properties.

PQL SETUP

PQL queries over the process store are only supported on MySQL. Create and populate the database with additional tables for PQL:
$ mysql --user=root --password=MAcri < Supplements/database/PQL.MySQL-1.2.sql

In site.properties, perform the following changes:
* Change pql.numberOfIndexerThreads to at least 1
* Change pql.numberOfQueryThreads to at least 1
* Change pql.lola.dir to the location of your LoLA 2.0 executable
* Change the various pql.mysql.* properties to match your MySQL database

In build.xml, uncomment the inclusion of the following PQL components in the "pickupRepo" fileset:
* APQL-Portal-Plugin/target/apql-portal-plugin-1.1.war
* Apromore-Assembly/PQL-Indexer-Assembly/src/main/resources/103-pql-indexer.plan
* PQL-Logic/target/pql-logic-1.1.jar
* PQL-Logic-WS/target/pql-logic-ws-1.1.war
* PQL-Portal-Plugin/target/pql-portal-plugin-1.1.jar

Also, uncomment the following component in the "copy-virgo" target:
* PQL-Indexer-Portal-Plugin/target/pql-indexer-portal-plugin-1.1.jar

BUILD IT

* Check out the Apromore source tree using git: 'git clone https://github.com/apromore/ApromoreCode.git'
* Change to the root of the project you checked out of git. 'cd ApromoreCore'
* Run the maven command 'mvn clean install'. This will build the Apromore manager, portal and editor and all the extra plugins.
* Create an empty H2 database 'ant create-h2'. Only do this once, unless you just want to reset to a blank database later on.
* Run the ant command 'ant start-virgo'. This will install, configure and start Eclipse Virgo, and deploy Apromore.
* Apromore runs on all modern browsers, browse http://localhost:9000/portal
* Ctrl-C on the command line will shut the server down.

SAMPLE DATA

You can upload some sample data into the system with the following command:
$ ant install-sample-data

/airport contains a Configurable BPMN process models which demonstrate configurability
/pql contains Petri nets in PNML format from the PQL test suite
/repair contains a BPMN model which demonstrates log animation

TRY IT OUT

Apromore runs on all modern browsers, browse http://localhost:9000/portal

NOTE: The current configuration of Apromore running in Virgo does show errors on the startup. These can be ignored as they aren't related to Apromore.

Please keep track of this page as we are constantly changing the build to make it better. Also in the near future, we will be releasing artifacts for all different system for an even easier setup.