Using Simple API for Workflow with Java CAPS 6 Configuration Guide

Sun Project Number:	SWIFT Demonstration
Sun Document Reference:	
Version Number:	D-0.1
Author:	Patrice Goutin
Document Date:	Aug 21, 2008

Table of Contents

1. Overview..... 4
1.1. WebApp approach to manage Human Workflow activities in a Java CAPS 6 Repository Based Application..... 4
1.2. Portal approach to manage Human Workflow activities in a Java CAPS 6 Repository Based Application..... 4
1.3. How-to step by step..... 5

Document Control

Revision History

Version	Author	Review	Reason For Issue	Dat
D-0.1	Patrice Goutin		Initial release	August 2008

Document Distribution

Copy	Owner	Location / Address
1	XXXX	xxxx@xxxx

Document References

Ref	Referenced Item

Document Access Control

Section	Access Restrictions
	Confidential - Sun Microsystems

1. Overview

This blog illustrates how to use a WLM portlet to manage Human Workflow activities in a Java CAPS 6 Repository Based Application.

1.1. WebApp approach to manage Human Workflow activities in a Java CAPS 6 Repository Based Application

If you want to use Human Workflow activities in a Java CAPS 6 Repository Based application, you can develop a Web application as the following to manage the tasks assigned to a given user :

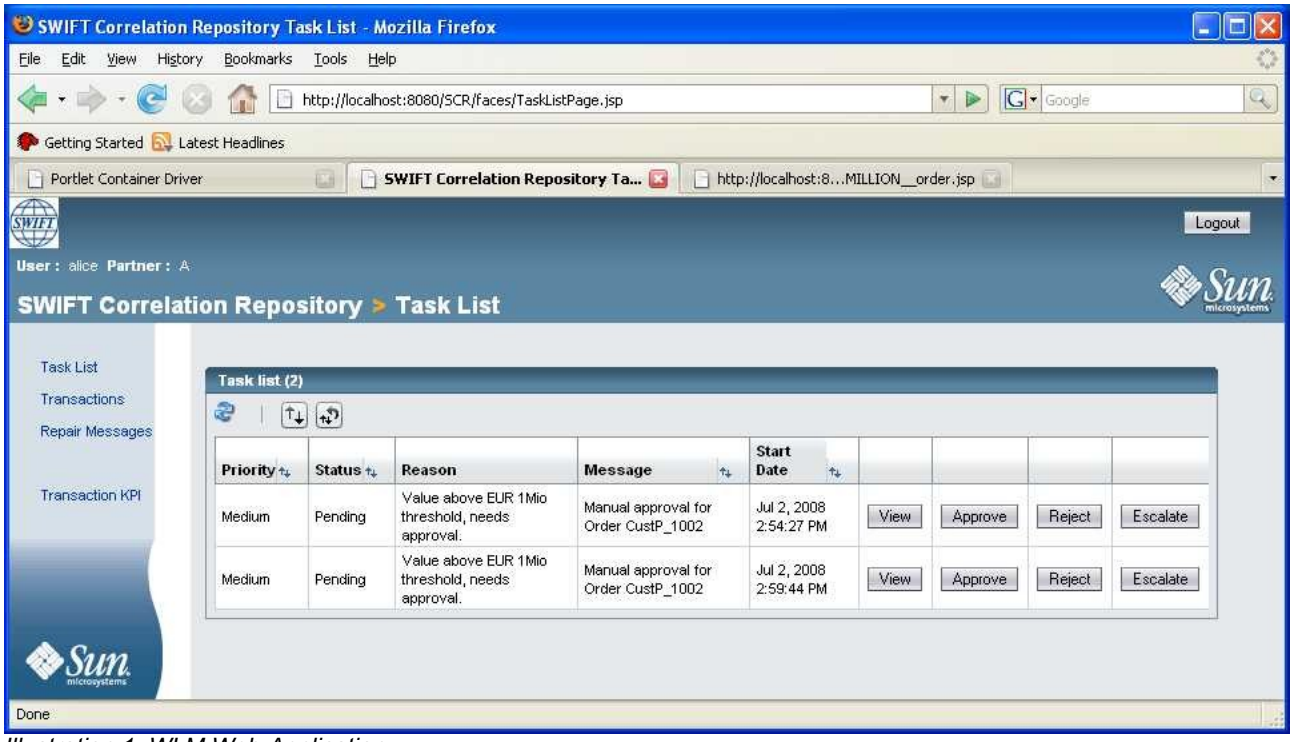

Illustration 1: WLM Web Application

A couple of blog illustrates how to create such Web application and how to configure Java CAPS 6 appserver to handle Authentication/Authorization policies.

1.2. Portal approach to manage Human Workflow activities in a Java CAPS 6 Repository Based Application

The WLM portlet can be easily deployed on any JSR168 portal. As you are working with Java CAPS 6 it could be a good idea to consider deploying the WLM portlet in a light portlet container as OpenPortlal Container (OPC). This portlet container can be deployed in a couple of clicks inside the Java CAPS 6 appserver instance.

For this blog, we will focus on the OPC deployment mode. By the way, in the near future, I will demonstrate how to deploy this WLM portlet with Web Sysnergy portal.

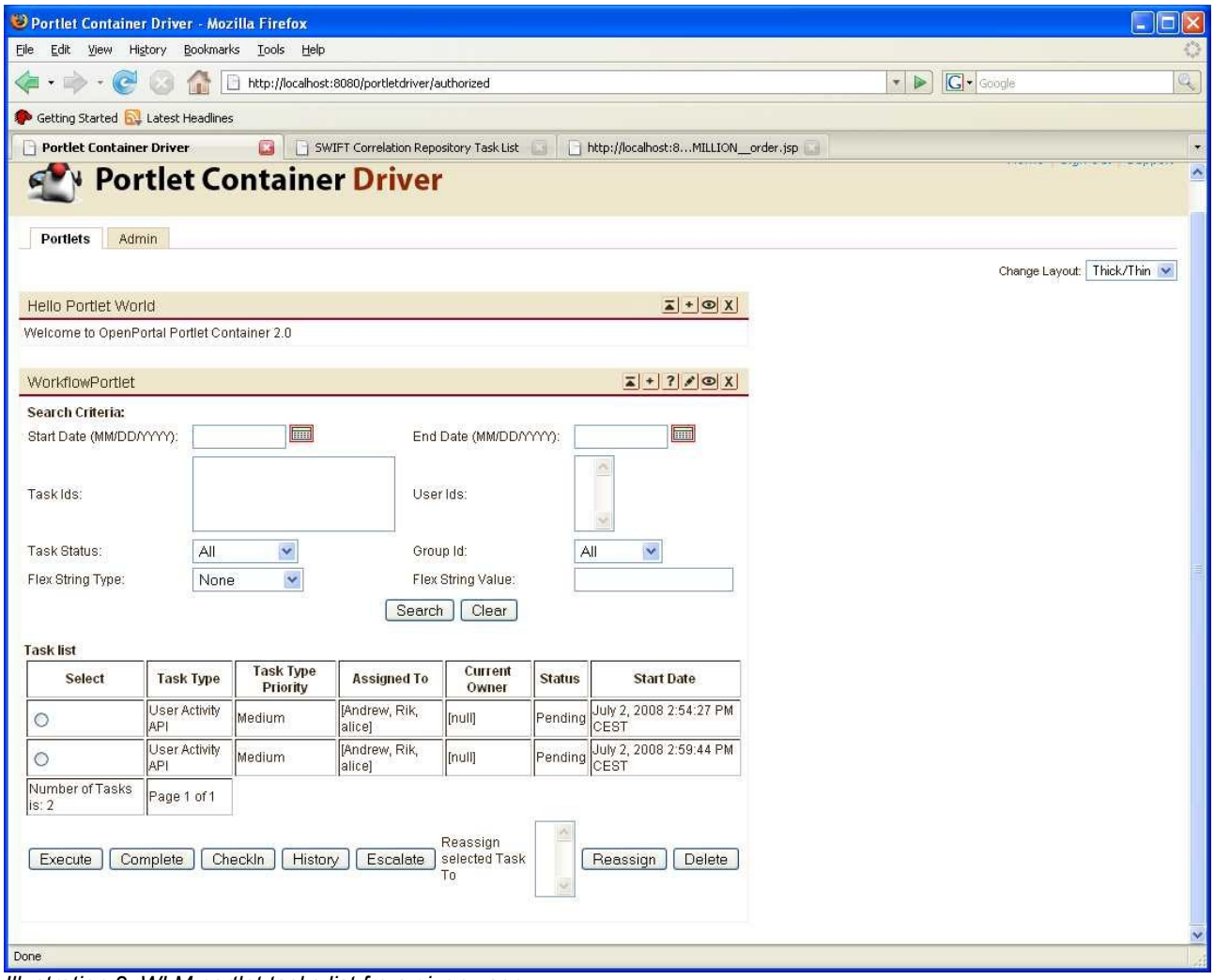

Illustration 2: WLM portlet tasks list fro a given user

1.3. How-to step by step

The following is a reminder of the main configuration steps to be performed to get the WLM portlet up & running.

1. Setup the WLM capabilities for your Java CAPS 6 Repository based application. Of course, this include also some configuration for the Java CAPS 6 apperserver instance. These steps are well-known and fully documented in the Java CAPS6 documentation.

2. Get the WLM portlet artifacts
   You need to get the following components :

Components	URLs
SAW API & SAW API Java Docs for Java CAPS 6	https://saw.dev.java.net/docs/SAWDownloadPage.html
WLM portlet The sample portlet will showcase how to use SAW to manage task with the help of workflow engine provided by Sun Java CAPS. It can be extended to use other workflow engines(see saw for more details) or customized to offer specific task management functionalities.	https://portlet-repository.dev.java.net/public/Download.html
Tooling Plugins for developer IDE's which empower the developer to use the SAW API and Tag Libraries with the	

greatest of ease. - Portal Pack 2.0.1 - OpenPortal Container 2.0	 http://portalpack.netbeans.org/nb6/download.html https://portlet-container.dev.java.net/

3. Installation instruction for Portal Pack 2.0.1 for NetBeans 6.1
   Follow the installation instruction from http://portalpack.netbeans.org/nb6/download.html

4. Installation instruction for Portlet Container 2.0
   Follow the instruction from
   https://portlet-container.dev.java.net/public/Download.html#Instructions_to_install_Portlet_

5. Configuration of the WLM portlet
   This configuration is required to reflect your specific setup for the Java CAPS 6 WLM Service and the way the WLM portlet is going to handle Authentication/Authorization. Follow the instruction from
   http://wiki.java.net/bin/view/OpenPortal/WorkflowPortlet

   By the way, as I encountered some minor issues between the documentation available so far and the portlet itself, I provide a description of my own modifications to the initial WLM portlet

1. Extract the workflowPortlet.war into a local directory : [working-dir].
2. Substitute the WorkflowServiceClient.jar file in [working-dir]\WEB-INF\lib\ by your own jar file generated during the Java CAPS 6 WLM configuration
3. Modify the workflowportlet.properties file in [working-dir]\WEB-INF\classes directory giving suitable values to the following properties :
   # The authenticationRepository used. The value should be accessManager for Open Portal and appServer for OSPC.
   authenticationRepository = appServer
4. Modify the WorkflowConfig.properties file in [working-dir]\WEB-INF\classes directory giving suitable values to the following properties :

   # the saw workflow impl class to use
   sawworkflowimplclass = **com.sun.saw.impls.jcaps.JCAPSWorkflow**

   # Properties that are needed by the JCAPS Implementation of SAW.

   # The host where jcaps server is running. Eg. test.domain.com
   com.sun.saw.impls.jcaps.JCAPSWorkflow.appserverhost = localhost

   # The iiop port where the ejb look up happens. Eg. 18002
   com.sun.saw.impls.jcaps.JCAPSWorkflow.iiopport = **3100**

   # Admin user name of the jcaps server Eg. Administrator
   com.sun.saw.impls.jcaps.JCAPSWorkflow.appserverusername = **admin**

   # Admin password of the jcaps server
   com.sun.saw.impls.jcaps.JCAPSWorkflow.appserverpassword = **adminadmin**

   # Context factory Eg.com.sun.jndi.cosnaming.CNCtxFactory
   com.sun.saw.impls.jcaps.JCAPSWorkflow.contextfactory = **com.sun.jndi.cosnaming.CNCtxFactory**

   # Jndi look up name. Eg WorkflowService
   com.sun.saw.impls.jcaps.JCAPSWorkflow.serviceJndi = **WorkflowService**
5. Recreate the workflowPortlet.war

6. Java CAPS 6 appserver Admin-Reamls configuration
   My WLM portlet Authentication/Authorization is based on an admin-reamls configuration

1. Login to the Java CAPS 6 admin -> under configuration -> security -> Realms -> click on admin-realm and click on Manager users button ,
2. add [user_name], group list : asadmin and passowrd as [user_password]
3. repeat step 2 as required

7. Java CAPS 6 appserver classpath configuration
   The saw-api-0.7.jar should be available in the container classpath of the Java CAPS 6 appserver.

   The saw.tld, saw-impl-jcaps-0.7.jar and WorkflowServiceClient.jar should be bundled with the WLM portlet.

8. Deploying the WLM portlet

   Access the portlet like this : http://machine:port/portletdriver/dt.
   With the Admin tab install the WLM portlet in the OPC.

Illustration 3: OpenPortal Container Admin tab

9. Running the WLM portlet
   Access the portlet like this. http://machine:port/portletdriver/dt. With the Portlets tab log in to the WLM portlet with a valid credential user, and the tasks assigned to this user would be displayed on the workflow portlet. Read the WORKFLOW PORTLET - USER GUIDE to learn how to manage WLM tasks.

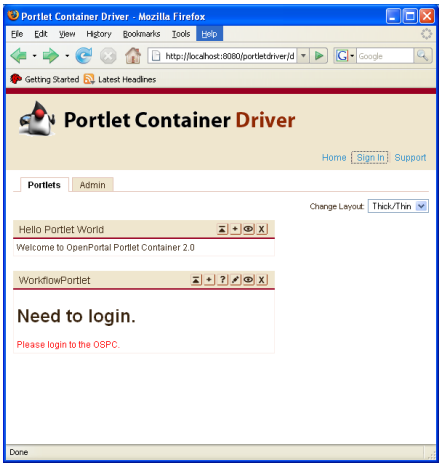

Illustration 4: OpenPortal Container Portlets tab

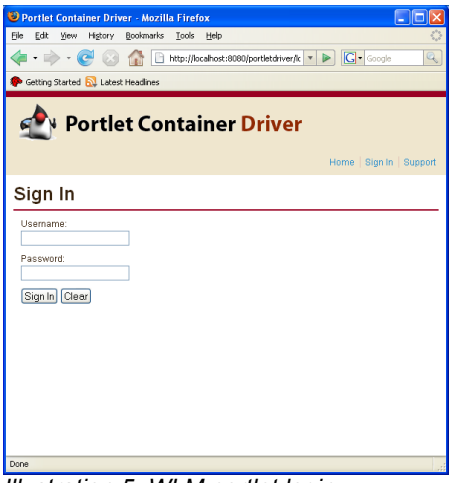

Illustration 5: WLM portlet login

You should be able to get the WLM tasks list :

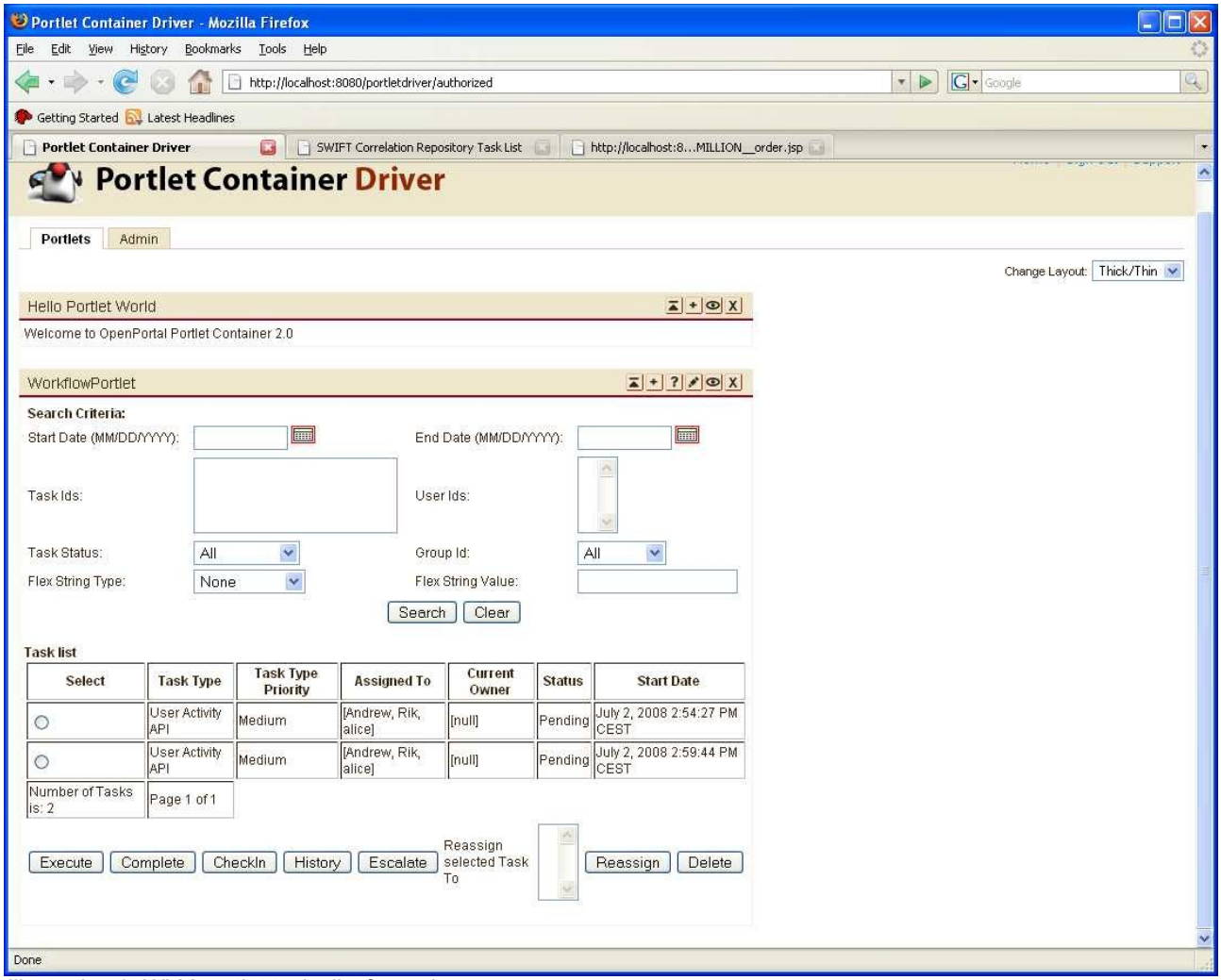

Illustration 6: WLM portlet tasks list for a given user