JBoss ESB 4.6

SOA Software Integration

JBESB-GS-7/17/09

Legal Notices

The information contained in this documentation is subject to change without notice.

JBoss Inc. makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. JBoss Inc. shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Java™ and J2EE is a U.S. trademark of Sun Microsystems, Inc. Microsoft® and Windows NT® are registered trademarks of Microsoft Corporation. Oracle® is a registered U.S. trademark and Oracle9™, Oracle9 Server™ Oracle9 Enterprise Edition™ are trademarks of Oracle Corporation. Unix is used here as a generic term covering all versions of the UNIX® operating system. UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

Copyright

Software Version

JBoss ESB 4.6

Restricted Rights Legend

Configuration

In order to communicate with the SOA Software registry, you will have to configure JBoss ESB to communicate with the SOA Software registry. In order to do this, the first thing you need to do is to point the JBoss ESB towards the SOA Software registry.

This text assumes you have a running JBoss Application/ESB Server, version 4.6.GA.

jbossesb-properties.xml

Edit server/<profile>/deploy/jbossesb.sar/jbossesb-properties.xml and change the following lines – if you are deploying to JBoss AS 5.1.0 or higher, this file is located in server/<profile>/deployers/esb.deployer/jbossesb-properties.xml :

```
<property name="org.jboss.soa.esb.registry.queryManagerURI" value="org.apache.juddi.registry.local.InquiryService#inquire"/>
<property name="org.jboss.soa.esb.registry.lifeCycleManagerURI" value="org.apache.juddi.registry.local.PublishService#publish"/>
```

To the following (and make sure to edit the queryManagerURI and the lifeCycleManagerURI to match the URLs from your installation of SOA Workbench) :

```
<property name="org.jboss.soa.esb.registry.uddi.maxRows" value="100"/>
<property name="org.jboss.soa.esb.registry.queryManagerURI" value="http://stacks2:9901/uddi/inquiry_v2"/>
<property name="org.jboss.soa.esb.registry.lifeCycleManagerURI" value="http://stacks2:9901/uddi/publish_v2"/>
```

Change jbossesb-properties.xml:

```
<property name="org.jboss.soa.esb.scout.proxy.transportClass" value="org.apache.ws.scout.transport.LocalTransport"/>
```

To

```
<property name="org.jboss.soa.esb.scout.proxy.transportClass" value="org.apache.ws.scout.transport.AxisTransport"/>
```

Also, within jbossesb-properties.xml, change :

```
<property name="org.jboss.soa.esb.registry.user" value="administrator"/>
<property name="org.jboss.soa.esb.registry.password" value="password"/>
```

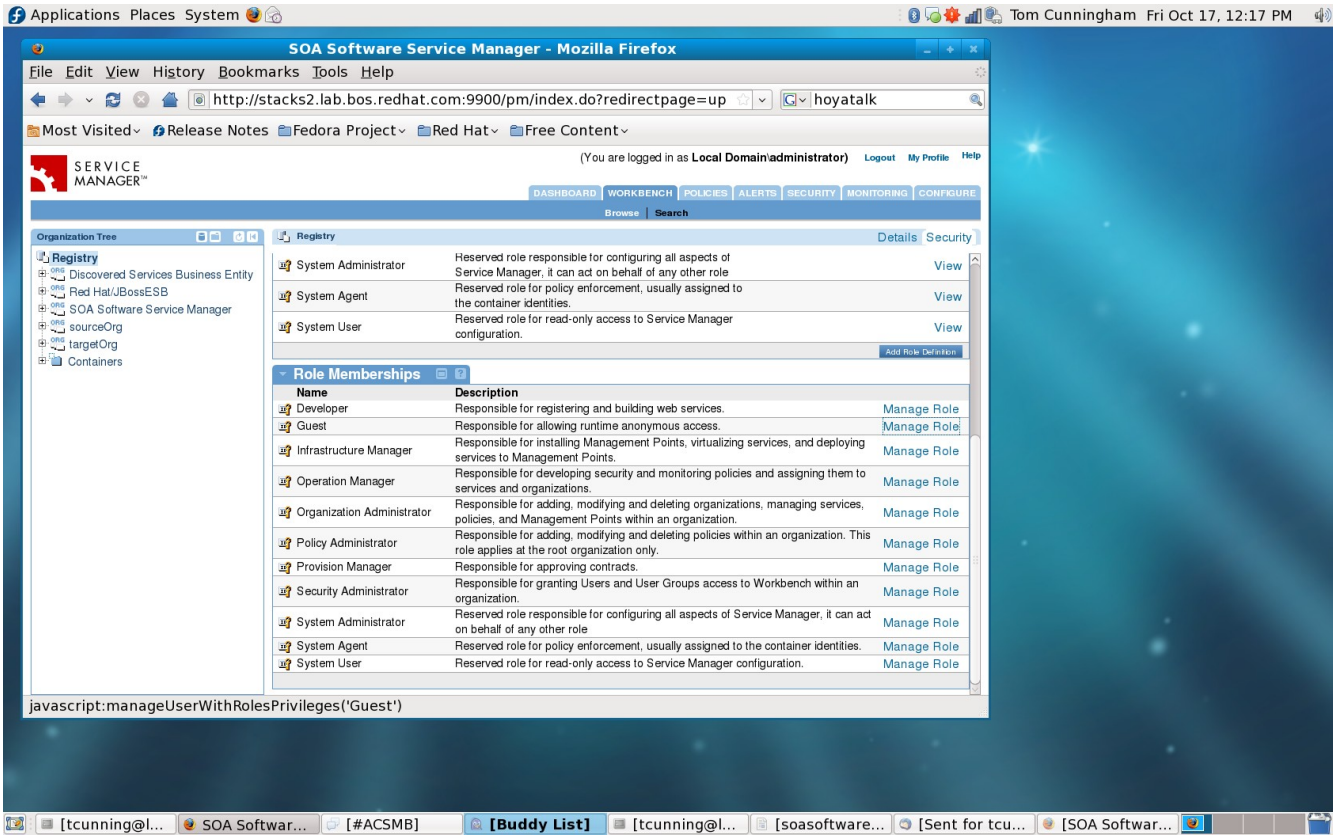
to match the SOA Workbench user and password you'll be using.

Add Axis and commons-discovery JARs

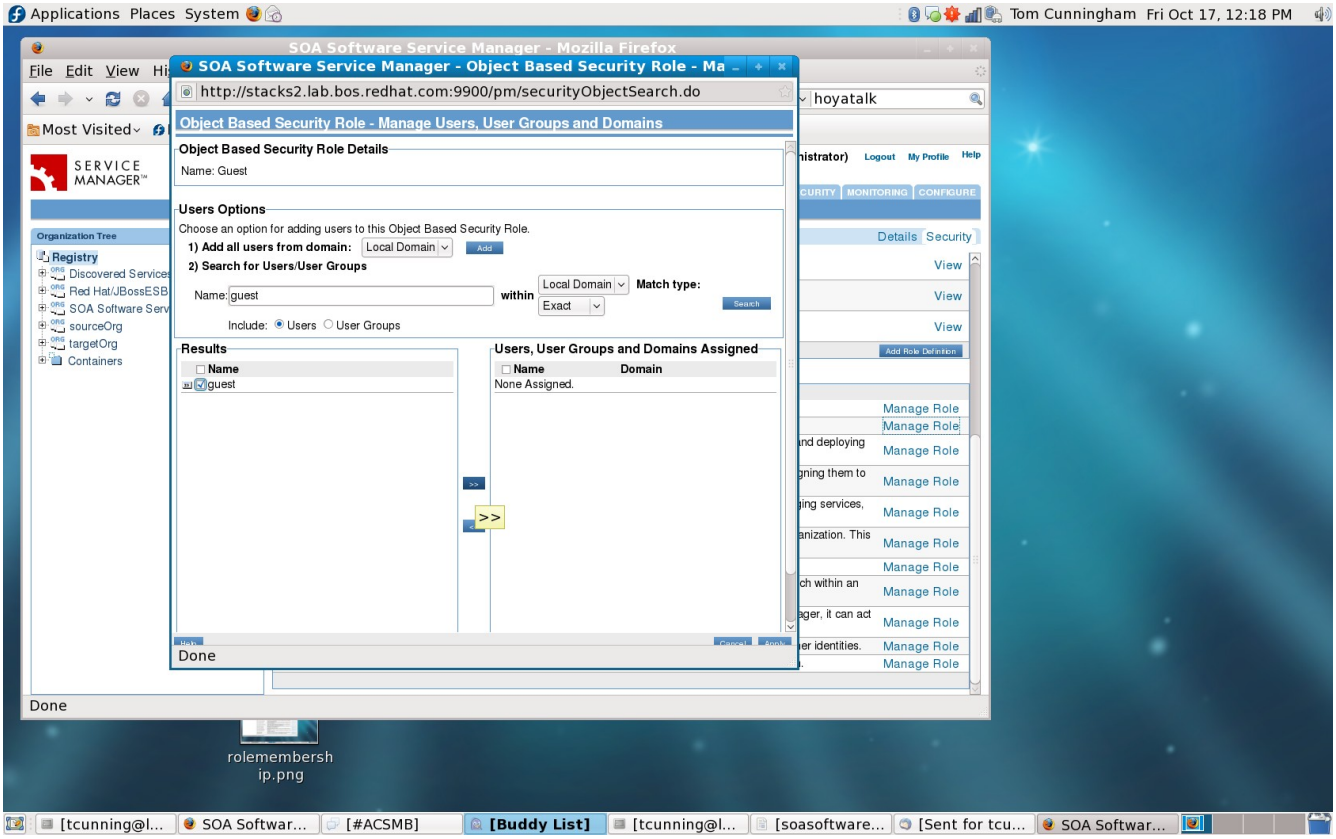
Copy a recent version of Apache Axis and Apache Commons Discovery into server/<profile>/lib. The integration was tested using axis-1.4.jar and commons-discovery-0.2.jar. Download axis-bin-1_4.tar.gz from the Apache Axis site (http://ws.apache.org/axis/) to find these JARs.

SOA Workbench Configurations

Configure the guest user in Workbench to the guest role. Click on the top level registry node in Workbench (in the Organization Tree) and click on the second-level “Security” tab (within the Workbench hierarchy). Click “Manage Role” for “Guest” under “Role Memberships :



Search for the user “guest”, and add them to the Guest Role by selecting the checkbox next to “guest” clicking “Apply” :



Workflow Configurations

Change the Workflow so that ESB services are published upon creation. Click on the “Configure” tab and then select “Workflow”. View the Workflow Definition, save it to your local disk, and then change the function type line under @create to <function type="publish" /> and the step attribute to “100”. After saving, Update the Workflow to the local copy you have saved.

Before:

```
<action id="1" name="@create">
    <results>
        <unconditional-result old-status="Created" status="Draft" step="100" owner="${caller}"/>
    </results>
    <post-functions>
        <function type="setLifecycleStage">
            <arg name="stage">Design</arg>
        </function>
    </post-functions>
</action>
```

After:

```
<action id="1" name="@create">
  <results>
    <unconditional-result old-status="Created" status="Draft" step="100" owner="${caller}"/>
  </results>
  <post-functions>
    <function type="publish" />
  </post-functions>
</action>
```

Running

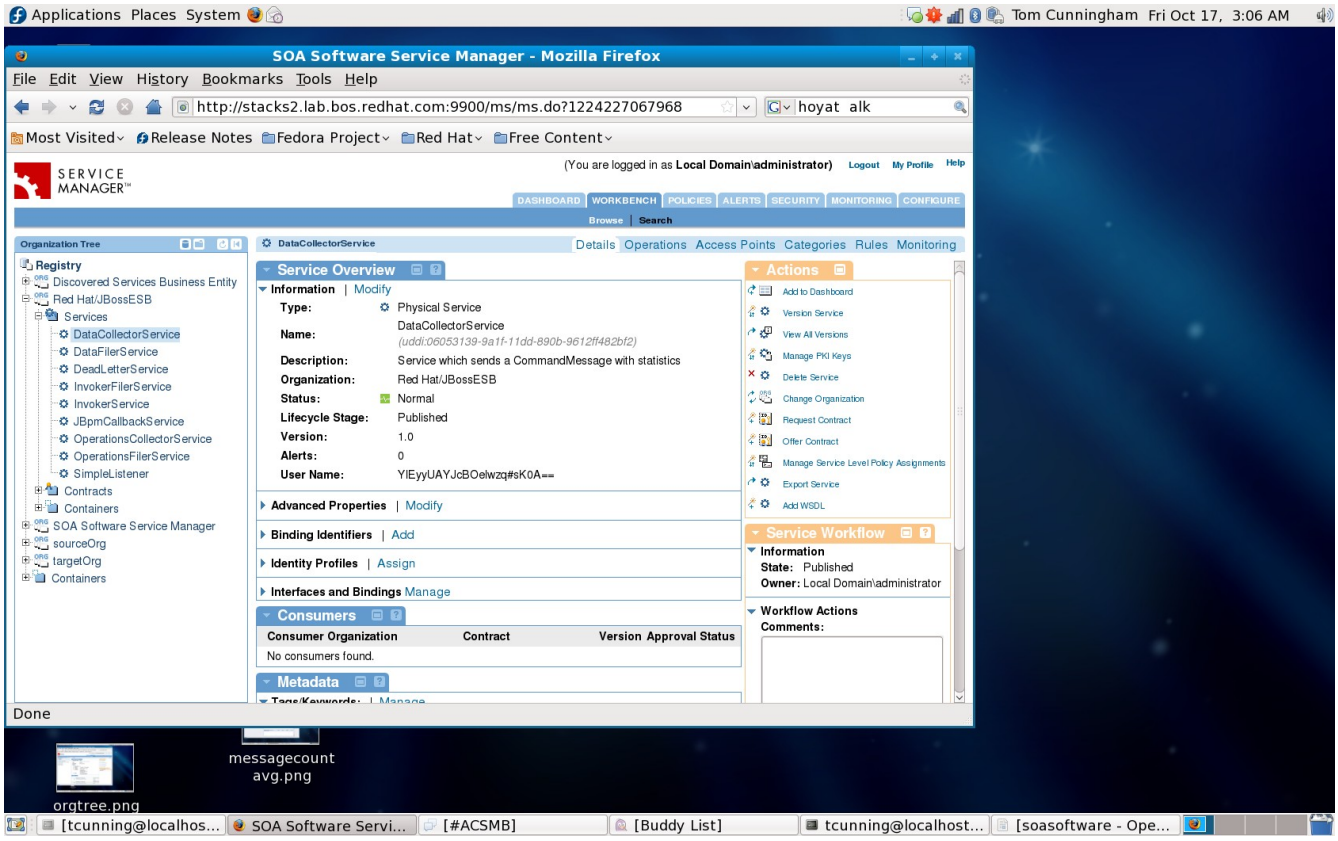
Start JBossESB :

```
cd bin
run.bat (if on Windows) or run.sh (on Unix)
```

Read product/docs/GettingStarted.pdf and follow the directions for running the Helloworld QS.

Managing Your Services

You can manage your services through Workbench – as you notice in the left frame of the screenshot below, services register with the SOA Software Workbench.



Search:

The “Search” option within Workbench allows the user to manage their services and shows the state, version, description, and number of alerts produced from those services.