

Hi,

I'm using Talend Platform for MDM 5.4.2 on Windows Server 2012 R2 and am attempting to configure the Drools Guvnor Repository in Talend Admin Centre, using instructions at I used Drools -> Administration -> Repository Configuration to generate a repository.xml for connecting to an SQL Server database, placed this in tac\apache-tomcat\webapps\drools-guvnor\WEB-INF and edited the components.xml in the same location and edited tac\apache-tomcat\webapps\drools-guvnor\WEB-INF\workspaces\default\workspace.xml to use the same driver and JDBC URL.

When TAC is restarted tables are created in the SQL Server database and some data populated. However the Drools Guvnor web app is no longer available (HTTP 404) and the following error is logged in tac\apache-tomcat\logs\localhost.<date>.log:

```
Dec 09, 2014 4:51:52 PM org.apache.catalina.core.StandardContext listenerStart
SEVERE: Exception sending context initialized event to listener instance of class org.jboss.seam.servlet.SeamListener
org.jboss.seam.InstantiationException: Could not instantiate Seam component: repositoryConfiguration
  at org.jboss.seam.Component.newInstance(Component.java:2066)
  at org.jboss.seam.contexts.Contexts.startup(Contexts.java:304)
  at org.jboss.seam.contexts.Contexts.startup(Contexts.java:278)
  at org.jboss.seam.contexts.ServletLifecycle.endInitialization(ServletLifecycle.java:112)
  at org.jboss.seam.init.Initialization.init(Initialization.java:727)
  at org.jboss.seam.servlet.SeamListener.contextInitialized(SeamListener.java:34)
  at org.apache.catalina.core.StandardContext.listenerStart(StandardContext.java:4206)
  at org.apache.catalina.core.StandardContext.start(StandardContext.java:4705)
  at org.apache.catalina.core.ContainerBase.addChildInternal(ContainerBase.java:799)
  at org.apache.catalina.core.ContainerBase.addChild(ContainerBase.java:779)
  at org.apache.catalina.core.StandardHost.addChild(StandardHost.java:601)
  at org.apache.catalina.startup.HostConfig.deployWAR(HostConfig.java:943)
  at org.apache.catalina.startup.HostConfig.deployWARs(HostConfig.java:778)
  at org.apache.catalina.startup.HostConfig.deployApps(HostConfig.java:504)
  at org.apache.catalina.startup.HostConfig.start(HostConfig.java:1317)
  at org.apache.catalina.startup.HostConfig.lifecycleEvent(HostConfig.java:324)
  at org.apache.catalina.util.LifecycleSupport.fireLifecycleEvent(LifecycleSupport.java:142)
  at org.apache.catalina.core.ContainerBase.start(ContainerBase.java:1065)
  at org.apache.catalina.core.StandardHost.start(StandardHost.java:840)
  at org.apache.catalina.core.ContainerBase.start(ContainerBase.java:1057)
  at org.apache.catalina.core.StandardEngine.start(StandardEngine.java:463)
  at org.apache.catalina.core.StandardService.start(StandardService.java:525)
  at org.apache.catalina.core.StandardServer.start(StandardServer.java:754)
  at org.apache.catalina.startup.Catalina.start(Catalina.java:595)
  at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
  at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
  at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
  at java.lang.reflect.Method.invoke(Method.java:606)
  at org.apache.catalina.startup.Bootstrap.start(Bootstrap.java:289)
  at org.apache.catalina.startup.Bootstrap.main(Bootstrap.java:414)
Caused by: java.lang.NullPointerException: dispatcher
  at org.apache.jackrabbit.core.observation.ObservationManagerImpl.<init>(ObservationManagerImpl.java:97)
  at org.apache.jackrabbit.core.WorkspaceImpl.getObservationManager(WorkspaceImpl.java:748)
  at org.apache.jackrabbit.core.SessionImpl.removeRegisteredEventListeners(SessionImpl.java:1193)
  at org.apache.jackrabbit.core.SessionImpl.logout(SessionImpl.java:1222)
  at org.apache.jackrabbit.core.RepositoryImpl$WorkspaceInfo.doDispose(RepositoryImpl.java:2194)
  at org.apache.jackrabbit.core.RepositoryImpl$WorkspaceInfo.dispose(RepositoryImpl.java:2149)
  at org.apache.jackrabbit.core.RepositoryImpl.doShutdown(RepositoryImpl.java:1175)
  at org.apache.jackrabbit.core.RepositoryImpl.shutdown(RepositoryImpl.java:1127)
  at org.apache.jackrabbit.core.RepositoryImpl.initStartupWorkspaces(RepositoryImpl.java:544)
  at org.apache.jackrabbit.core.RepositoryImpl.<init>(RepositoryImpl.java:366)
  at org.apache.jackrabbit.core.RepositoryImpl.create(RepositoryImpl.java:673)
  at org.apache.jackrabbit.core.TransientRepository$2.getRepository(TransientRepository.java:231)
  at org.apache.jackrabbit.core.TransientRepository.startRepository(TransientRepository.java:279)
```

```
at org.apache.jackrabbit.core.TransientRepository.login(TransientRepository.java:375)
at org.apache.jackrabbit.commons.AbstractRepository.login(AbstractRepository.java:123)
at org.drools.repository.JCRRepositoryConfigurator.login(JCRRepositoryConfigurator.java:95)
at org.drools.repository.RulesRepositoryConfigurator.login(RulesRepositoryConfigurator.java:85)
at org.drools.guvnor.server.repository.RepositoryStartupService.newSession(RepositoryStartupService.java:213)
at org.drools.guvnor.server.repository.RepositoryStartupService.create(RepositoryStartupService.java:93)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:606)
at org.jboss.seam.util.Reflections.invoke(Reflections.java:22)
at org.jboss.seam.intercept.RootInvocationContext.proceed(RootInvocationContext.java:31)
at org.jboss.seam.intercept.SeamInvocationContext.proceed(SeamInvocationContext.java:56)
at org.jboss.seam.transaction.RollbackInterceptor.aroundInvoke(RollbackInterceptor.java:28)
at org.jboss.seam.intercept.SeamInvocationContext.proceed(SeamInvocationContext.java:68)
at org.jboss.seam.core.MethodContextInterceptor.aroundInvoke(MethodContextInterceptor.java:44)
at org.jboss.seam.intercept.SeamInvocationContext.proceed(SeamInvocationContext.java:68)
at org.jboss.seam.intercept.RootInterceptor.invoke(RootInterceptor.java:107)
at org.jboss.seam.intercept.BeanInterceptor.interceptInvocation(BeanInterceptor.java:166)
at org.jboss.seam.intercept.BeanInterceptor.invoke(BeanInterceptor.java:102)
at org.drools.guvnor.server.repository.RepositoryStartupService_$$javassist_1.create(RepositoryStartupService_$$javassist_1.java)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:606)
at org.jboss.seam.util.Reflections.invoke(Reflections.java:22)
at org.jboss.seam.util.Reflections.invokeAndWrap(Reflections.java:138)
at org.jboss.seam.Component.callComponentMethod(Component.java:2171)
at org.jboss.seam.Component.callCreateMethod(Component.java:2094)
at org.jboss.seam.Component.newInstance(Component.java:2054)
... 29 more
```

Is there a way to prevent this error so we can use SQL server as the repository?
Or can the Drools Guvnor Repository be integrated into the Talend repository (SVN)?

Thanks

Below are the 3 xml files (I don't seem to be able to upload them)

components.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<components xmlns=""
  xmlns:core="" xmlns:security=""
  xmlns:web="" xmlns:xsi=""
  xmlns:transaction=""
  xsi:schemaLocation=""

  ">

  <core:init transaction-management-enabled="false" />
  <transaction:no-transaction />
  <component name="repositoryConfiguration">
    <!-- JackRabbit -->

    <property name="properties">
```

```

    <key>org.drools.repository.configurator</key><value>org.drools.repository.jackrabbit.JackrabbitRepositoryConfigurator</value>
    <!-- the root directory for the repo storage the directory must exist. -->
    <key>repository.root.directory</key><value>E:\Talend\5.4.2\tac\apache-tomcat\webapps\drools-guvnor\WEB-INF\</value>
</property>
<!-- ModeShape
passwords for the background users (logInAdmin and mailman), these need to match the setting
you provided for JAAS (used by ModeShape only).
<property name="properties">
    <key>org.drools.repository.configurator</key>    <value>org.drools.repository.modeshape.ModeShapeRepositoryConfigurator</value>

    <key>org.modeshape.jcr.URL</key>                <value>jndi:jcr/local?repositoryName=brms</value>
    <key>org.drools.repository.secure.passwords</key><value>>false</value>
    <key>org.drools.repository.logInAdmin.password</key> <value>logInAdmin</value>
    <key>org.drools.repository.mailman.password</key><value>mailman</value>
</property>
-->

</component>
<!-- SECURITY IDENTITY CONFIGURATION -->
<!--
    default (will take any username, useful if you want to keep track of
    users but not authenticate

<security:identity authenticate-method="#{defaultAuthenticator.authenticate}" />
-->
<!--
    NO authentication. This will bypass the login screen when you hit the
    app. Everyone is "guest"
-->
    <security:identity authenticate-method="#{authenticator.authenticate}" remember-me="false" />
<!--
    FOR EXAMPLE: the following one will use the jaas configuration called
    "other" - which in jboss AS means you can use properties files for
    users:

    <security:identity authenticate-method="#{authenticator.authenticate}"
    jaas-config-name="other"/>
-->
<!--
    as JAAS is used you can use container specific ones to link up to your
    login services, eg LDAP/AD etc
-->
<!-- SECURITY AUTHORIZATION CONFIGURATION -->
<!--
    This is used to enable or disable role-based authorization. By default
    it is disabled.
-->
<component name="org.jboss.seam.security.roleBasedPermissionResolver">
    <property name="enableRoleBasedAuthorization">>false</property>
</component>
</components>

```

respository.xml:

```

<?xml version="1.0"?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<!DOCTYPE Repository PUBLIC "-//The Apache Software Foundation//DTD Jackrabbit 1.4//EN"
    "">
<!-- Example Repository Configuration File -->
<Repository>
  <!--
    virtual file system where the repository stores global state
    (e.g. registered namespaces, custom node types, etc.)
  -->
  <FileSystem class="org.apache.jackrabbit.core.fs.db.MSSqlFileSystem">
    <param name="driver" value="com.microsoft.sqlserver.jdbc.SQLServerDriver"/>
    <param name="url" value="jdbc:sqlserver://<IP address>:1433"/>
    <param name="schema" value="mssql"/>
    <param name="user" value="drools"/>
    <param name="password" value="drools"/>
    <param name="schemaObjectPrefix" value="FS_"/>
  </FileSystem>
  <!--
    security configuration
  -->
  <Security appName="Jackrabbit">
    <!--
      access manager:
      class: FQN of class implementing the AccessManager interface
    -->
    <AccessManager class="org.apache.jackrabbit.core.security.SimpleAccessManager">
      <!-- <param name="config" value="{rep.home}/access.xml"/> -->
    </AccessManager>
    <LoginModule class="org.apache.jackrabbit.core.security.SimpleLoginModule">
      <!-- anonymous user name ('anonymous' is the default value) -->
      <!-- <param name="anonymousId" value="anonymous"/> -->
      <!--
        default user name to be used instead of the anonymous user
        when no login credentials are provided (unset by default)
      -->
      <!-- <param name="defaultUserId" value="superuser"/> -->
    </LoginModule>
  </Security>
  <!--
    location of workspaces root directory and name of default workspace
  -->

```

```

<Workspaces rootPath="\${rep.home}/workspaces" defaultWorkspace="default"/>
<!--
  workspace configuration template:
  used to create the initial workspace if there's no workspace yet
-->
<Workspace name="\${wsp.name}">
  <!--
    virtual file system of the workspace:
    class: FQN of class implementing the FileSystem interface
  -->

  <FileSystem class="org.apache.jackrabbit.core.fs.db.MSSqlFileSystem">
    <param name="driver" value="com.microsoft.sqlserver.jdbc.SQLServerDriver"/>
    <param name="url" value="jdbc:sqlserver://<IP address>:1433"/>
    <param name="user" value="drools"/>
    <param name="password" value="drools"/>
    <param name="schema" value="mssql"/>
    <param name="schemaObjectPrefix" value="FS_WS_\${wsp.name}_"/>
  </FileSystem>
  <!--
    persistence manager of the workspace:
    class: FQN of class implementing the PersistenceManager interface
  -->
  <PersistenceManager class="org.apache.jackrabbit.core.persistence.bundle.MSSqlPersistenceManager">
    <param name="driver" value="com.microsoft.sqlserver.jdbc.SQLServerDriver"/>
    <param name="url" value="jdbc:sqlserver://<IP address>:1433"/>
    <param name="user" value="drools"/>
    <param name="password" value="drools"/>
    <param name="schema" value="mssql"/>
    <param name="schemaObjectPrefix" value="PM_WS_\${wsp.name}_"/>
  </PersistenceManager>
  <!--
    Search index and the file system it uses.
    class: FQN of class implementing the QueryHandler interface
  -->
  <SearchIndex class="org.apache.jackrabbit.core.query.lucene.SearchIndex">
    <param name="path" value="\${wsp.home}/index"/>
    <param name="textFilterClasses" value="org.apache.jackrabbit.extractor.MsWordTextExtractor,org.apache.jackrabbit.extractor.MsExcelText

    <param name="extractorPoolSize" value="2"/>
    <param name="supportHighlighting" value="true"/>
  </SearchIndex>
</Workspace>
<!--
  Configures the versioning
-->
<Versioning rootPath="\${rep.home}/version">
  <!--
    Configures the filesystem to use for versioning for the respective
    persistence manager
  -->
  <FileSystem class="org.apache.jackrabbit.core.fs.db.MSSqlFileSystem">
    <param name="driver" value="com.microsoft.sqlserver.jdbc.SQLServerDriver"/>
    <param name="url" value="jdbc:sqlserver://<IP address>:1433"/>
    <param name="user" value="drools"/>
    <param name="password" value="drools"/>

```

```

    <param name="schema" value="mssql"/>
    <param name="schemaObjectPrefix" value="Versioning_FS_"/>
</FileSystem>
    <!--
        Configures the persistence manager to be used for persisting version state.
        Please note that the current versioning implementation is based on
        a 'normal' persistence manager, but this could change in future
        implementations.
    -->
<PersistenceManager class="org.apache.jackrabbit.core.persistence.bundle.MSSqlPersistenceManager">
    <param name="driver" value="com.microsoft.sqlserver.jdbc.SQLServerDriver"/>
    <param name="url" value="jdbc:sqlserver://<IP address>:1433"/>
    <param name="user" value="drools"/>
    <param name="password" value="drools"/>
    <param name="schema" value="mssql"/>
    <param name="schemaObjectPrefix" value="Versioning_PM_" />
</PersistenceManager>

</Versioning>
<!--
    Search index for content that is shared repository wide
    (/jcr:system tree, contains mainly versions)
-->
<SearchIndex class="org.apache.jackrabbit.core.query.lucene.SearchIndex">
    <param name="path" value="{rep.home}/repository/index"/>
    <param name="textFilterClasses" value="org.apache.jackrabbit.extractor.MsWordTextExtractor,org.apache.jackrabbit.extractor.MsExcelTextExt

    <param name="extractorPoolSize" value="2"/>
    <param name="supportHighlighting" value="true"/>
</SearchIndex>
</Repository>

```

workspace.xml:

```

<?xml version="1.0" encoding="UTF-8"?><Workspace name="default">
    <!--
        virtual file system of the workspace:
        class: FQN of class implementing the FileSystem interface
    -->

    <FileSystem class="org.apache.jackrabbit.core.fs.db.MSSqlFileSystem">
    <param name="driver" value="com.microsoft.sqlserver.jdbc.SQLServerDriver"/>
    <param name="url" value="jdbc:sqlserver://<IP address>:1433"/>
    <param name="user" value="drools"/>
    <param name="password" value="drools"/>
    <param name="schema" value="mssql"/>
    <param name="schemaObjectPrefix" value="FS_WS_{wsp.name}_"/>
</FileSystem>
    <!--
        persistence manager of the workspace:
        class: FQN of class implementing the PersistenceManager interface
    -->
<PersistenceManager class="org.apache.jackrabbit.core.persistence.bundle.MSSqlPersistenceManager">
    <param name="driver" value="com.microsoft.sqlserver.jdbc.SQLServerDriver"/>
    <param name="url" value="jdbc:sqlserver://<IP address>:1433"/>

```

```
<param name="user" value="drools"/>
<param name="password" value="drools"/>
<param name="schema" value="mssql"/>
<param name="schemaObjectPrefix" value="PM_WS_${wsp.name}_"/>
</PersistenceManager>
<!--
  Search index and the file system it uses.
  class: FQN of class implementing the QueryHandler interface
-->
<SearchIndex class="org.apache.jackrabbit.core.query.lucene.SearchIndex">
  <param name="path" value="${wsp.home}/index"/>
  <param name="textFilterClasses" value="org.apache.jackrabbit.extractor.MsWordTextExtractor,org.apache.jackrabbit.extractor.MsExcelText

  <param name="extractorPoolSize" value="2"/>
  <param name="supportHighlighting" value="true"/>
</SearchIndex>
</Workspace>
```