**jotm跨数据库 spring + hibernate**

软件包   
ow2-jotm-dist-2.1.9-bin.tar.gz   
spring-framework-3.0.5.RELEASEhibernate-3.6.0.Final   
hibernate-3.6.0.Final   
  
spring和hibernate不再详述，讲讲jotm如何与spring集成   
web开发中，使用hibernate的SessionFactory，而在SessionFactory中注入DataSource   
DataSource需要支持事务，因此采用软件包ow2-jotm-dist-2.1.9-bin.tar.gz中的xapool.jar包来配置DataSource，放弃了org.springframework.jdbc.datasource.DriverManagerDataSource   
  
配置jta事务管理器JtaTransactionManager   
重点<property name="userTransaction" ref="jotm" />

**Java代码  [收藏代码](javascript:void())**

1. <bean id="jotm" **class**="org.springframework.transaction.jta.JotmFactoryBean" />
2. <bean id="transactionManager" **class**="org.springframework.transaction.jta.JtaTransactionManager">
3. <property name="userTransaction" ref="jotm" />
4. <property name="allowCustomIsolationLevels" value="true"></property>
5. </bean>

数据源配置(DataSource)   
重点<property name="transactionManager" ref="jotm" />

**Java代码  [收藏代码](javascript:void())**

1. //1号数据库
2. <bean id="dataSource1" **class**="org.enhydra.jdbc.pool.StandardXAPoolDataSource" destroy-method="shutdown">
3. <property name="dataSource">
4. <bean **class**="org.enhydra.jdbc.standard.StandardXADataSource" destroy-method="shutdown">
5. <property name="transactionManager" ref="jotm" />
6. <property name="driverName" value="com.mysql.jdbc.Driver" />
7. <property name="url" value="jdbc:mysql://127.0.0.1:3306/db1?characterEncoding=utf-8" />
8. </bean>
9. </property>
10. <property name="user" value="root" />
11. <property name="password" value="" />
12. </bean>
14. //2号数据库
15. <bean id="dataSource2" **class**="org.enhydra.jdbc.pool.StandardXAPoolDataSource" destroy-method="shutdown">
16. <property name="dataSource">
17. <bean **class**="org.enhydra.jdbc.standard.StandardXADataSource" destroy-method="shutdown">
18. <property name="transactionManager" ref="jotm" />
19. <property name="driverName" value="com.mysql.jdbc.Driver" />
20. <property name="url" value="jdbc:mysql://127.0.0.1:3306/db2?characterEncoding=utf-8" />
21. </bean>
22. </property>
23. <property name="user" value="root" />
24. <property name="password" value="" />
25. </bean>

SessionFactory配置   
重点<property name="dataSource" ref="dataSource1" />   
重点<property name="dataSource" ref="dataSource2" />

**Java代码  [收藏代码](javascript:void())**

1. //1号SessionFactory
2. <bean id="sessionFactory1" **class**="org.springframework.orm.hibernate3.annotation.AnnotationSessionFactoryBean">
3. <property name="hibernateProperties">
4. <props>
5. <prop key="hibernate.dialect">org.hibernate.dialect.MySQLDialect</prop>
6. </props>
7. </property>
8. <property name="dataSource" ref="dataSource1" />
9. <property name="packagesToScan" value="com.yuce.test.entity"/>
10. </bean>
12. //2号SessionFactory
13. <bean id="sessionFactory2" **class**="org.springframework.orm.hibernate3.annotation.AnnotationSessionFactoryBean">
14. <property name="hibernateProperties">
15. <props>
16. <prop key="hibernate.dialect">org.hibernate.dialect.MySQLDialect</prop>
17. </props>
18. </property>
19. <property name="dataSource" ref="dataSource2" />
20. <property name="packagesToScan" value="com.yuce.test.entity"/>
21. </bean>

附录   
jotm必须的jar包

**Java代码  [收藏代码](javascript:void())**

1. xapool.jar
2. jotm-core.jar
3. carol.jar
4. carol-interceptors.jar

后记：   
在使用spring的    进行跨数据库测试的时候没有问题，而在发布web的时候，启动始终都报异常javax.resource.spi.XATerminator找不到，查看类库的时候发现该类就在javaee.jar中，但发布的时候始终找不到。后来发现，这是由于在MyEclipse发布web的时候，设置项目->属性->Order and Export中没有将javaee.jar等jar包发布到web中，于是勾选该jar包，但测试仍然不能成功。   
最后将javaee.jar包copy到tomcat是lib目录中，启动tomcat进行测试，一切正常。   
暂时好不知道为什么设置了Order and Export也不能行的原因，只是知道在MyEclipse在打包发布的时候也不是完全可信的。切记。

spring-framework-3.0.5.RELEASE 包中没有org.springframework.transaction.jta.JotmFactoryBean类，咋配啊？

官方的回答是：   
JOTM and XAPool support is dropped in Spring 3.0.x (it has been droppen on 28-november-2008 so for quite some time now). There is however no JIRA issue attached to the commit.   
解决办法是：   
1) Copy the code from 2.5.6 (JotmFactoryBean) to your own project and use that implementation   
2) Simply instantiate your orn Jotm instance and use that instead of the factory bean   
3) Use something else instead of Jotm   
Either 3 will solve your problem.   
  
推荐第一种解决办法：   
自己写JotmFactoryBean，以下是spring2.5.6中的源码

**Java代码  [收藏代码](javascript:void())**

1. /\*
2. \* Copyright 2002-2008 the original author or authors.
3. \*
4. \* Licensed under the Apache License, Version 2.0 (the "License");
5. \* you may not use this file except in compliance with the License.
6. \* You may obtain a copy of the License at
7. \*
8. \*      http://www.apache.org/licenses/LICENSE-2.0
9. \*
10. \* Unless required by applicable law or agreed to in writing, software
11. \* distributed under the License is distributed on an "AS IS" BASIS,
12. \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13. \* See the License for the specific language governing permissions and
14. \* limitations under the License.
15. \*/
17. **package** org.springframework.transaction.jta;
19. **import** javax.naming.NamingException;
20. **import** javax.transaction.SystemException;
22. **import** org.objectweb.jotm.Current;
23. **import** org.objectweb.jotm.Jotm;
25. **import** org.springframework.beans.factory.DisposableBean;
26. **import** org.springframework.beans.factory.FactoryBean;
28. /\*\*
29. \* {@link FactoryBean} that retrieves the JTA UserTransaction/TransactionManager
30. \* for ObjectWeb's <a href="http://jotm.objectweb.org">JOTM</a>. Will retrieve
31. \* an already active JOTM instance if found (e.g. if running in JOnAS),
32. \* else create a new local JOTM instance.
33. \*
34. \* <p>With JOTM, the same object implements both the
35. \* {@link javax.transaction.UserTransaction} and the
36. \* {@link javax.transaction.TransactionManager} interface,
37. \* as returned by this FactoryBean.
38. \*
39. \* <p>A local JOTM instance is well-suited for working in conjunction with
40. \* ObjectWeb's <a href="http://xapool.experlog.com">XAPool</a>, e.g. with bean
41. \* definitions like the following:
42. \*
43. \* <pre class="code">
44. \* &lt;bean id="jotm" class="org.springframework.transaction.jta.JotmFactoryBean"/&gt;
45. \*
46. \* &lt;bean id="transactionManager" class="org.springframework.transaction.jta.JtaTransactionManager"&gt;
47. \*   &lt;property name="userTransaction" ref="jotm"/&gt;
48. \* &lt;/bean&gt;
49. \*
50. \* &lt;bean id="innerDataSource" class="org.enhydra.jdbc.standard.StandardXADataSource" destroy-method="shutdown"&gt;
51. \*   &lt;property name="transactionManager" ref="jotm"/&gt;
52. \*   &lt;property name="driverName" value="..."/&gt;
53. \*   &lt;property name="url" value="..."/&gt;
54. \*   &lt;property name="user" value="..."/&gt;
55. \*   &lt;property name="password" value="..."/&gt;
56. \* &lt;/bean&gt;
57. \*
58. \* &lt;bean id="dataSource" class="org.enhydra.jdbc.pool.StandardXAPoolDataSource" destroy-method="shutdown"&gt;
59. \*   &lt;property name="dataSource" ref="innerDataSource"/&gt;
60. \*   &lt;property name="user" value="..."/&gt;
61. \*   &lt;property name="password" value="..."/&gt;
62. \*   &lt;property name="maxSize" value="..."/&gt;
63. \* &lt;/bean&gt;</pre>
64. \*
65. \* Note that Spring's {@link JtaTransactionManager} will automatically detect
66. \* that the passed-in UserTransaction reference also implements the
67. \* TransactionManager interface. Hence, it is not necessary to specify a
68. \* separate reference for JtaTransactionManager's "transactionManager" property.
69. \*
70. \* <p>Implementation note: This FactoryBean uses JOTM's static access method
71. \* to obtain the JOTM {@link org.objectweb.jotm.Current} object, which
72. \* implements both the UserTransaction and the TransactionManager interface,
73. \* as mentioned above.
74. \*
75. \* @author Juergen Hoeller
76. \* @since 21.01.2004
77. \* @see JtaTransactionManager#setUserTransaction
78. \* @see JtaTransactionManager#setTransactionManager
79. \* @see org.objectweb.jotm.Current
80. \*/
81. **public** **class** JotmFactoryBean **implements** FactoryBean, DisposableBean {
83. **private** Current jotmCurrent;
85. **private** Jotm jotm;

88. **public** JotmFactoryBean() **throws** NamingException {
89. // Check for already active JOTM instance.
90. **this**.jotmCurrent = Current.getCurrent();
92. // If none found, create new local JOTM instance.
93. **if** (**this**.jotmCurrent == **null**) {
94. // Only for use within the current Spring context:
95. // local, not bound to registry.
96. **this**.jotm = **new** Jotm(**true**, **false**);
97. **this**.jotmCurrent = Current.getCurrent();
98. }
99. }
101. /\*\*
102. \* Set the default transaction timeout for the JOTM instance.
103. \* <p>Should only be called for a local JOTM instance,
104. \* not when accessing an existing (shared) JOTM instance.
105. \*/
106. **public** **void** setDefaultTimeout(**int** defaultTimeout) {
107. **this**.jotmCurrent.setDefaultTimeout(defaultTimeout);
108. // The following is a JOTM oddity: should be used for demarcation transaction only,
109. // but is required here in order to actually get rid of JOTM's default (60 seconds).
110. **try** {
111. **this**.jotmCurrent.setTransactionTimeout(defaultTimeout);
112. }
113. **catch** (SystemException ex) {
114. // should never happen
115. }
116. }

119. /\*\*
120. \* Return the JOTM instance created by this factory bean, if any.
121. \* Will be <code>null</code> if an already active JOTM instance is used.
122. \* <p>Application code should never need to access this.
123. \*/
124. **public** Jotm getJotm() {
125. **return** **this**.jotm;
126. }
128. **public** Object getObject() {
129. **return** **this**.jotmCurrent;
130. }
132. **public** Class getObjectType() {
133. **return** **this**.jotmCurrent.getClass();
134. }
136. **public** **boolean** isSingleton() {
137. **return** **true**;
138. }

141. /\*\*
142. \* Stop the local JOTM instance, if created by this FactoryBean.
143. \*/
144. **public** **void** destroy() {
145. **if** (**this**.jotm != **null**) {
146. **this**.jotm.stop();
147. }
148. }
150. }

也可以到<http://s3.amazonaws.com/dist.springframework.org/release/SPR/spring-framework-2.5.6-with-dependencies.zip>下载spring2.5.6，在源码里找到对应的类。