

XML static code analysis: "SingleConnectionFactory" instances should be set to "reconnectOnException"

2 minutes

Use of a Spring `SingleConnectionFactory` without enabling the `reconnectOnException` setting will prevent automatic connection recovery when the connection goes bad.

That's because the `reconnectOnException` property defaults to `false`. As a result, even if the code that uses this connection factory (Spring's `DefaultMessageListenerContainer` or your own code) has reconnect logic, that code won't work because the `SingleConnectionFactory` will act like a single-connection pool by preventing connection `close` calls from actually closing anything. As a result, subsequent factory `create` operations will just hand back the original broken `Connection`.

Noncompliant Code Example

```
<bean id="singleCF"
class="org.springframework.jms.connection.SingleConnectionFactory">
<!-- Noncompliant -->
    <constructor-arg ref="dummyConnectionFactory" />
</bean>
```

Compliant Solution

```
<bean id="singleCF"  
class="org.springframework.jms.connection.SingleConnectionFactory"  
p:reconnectOnException="true">  
  <constructor-arg ref="dummyConnectionFactory" />  
</bean>
```

or

```
<bean id="singleCF"  
class="org.springframework.jms.connection.SingleConnectionFactory">  
  <constructor-arg ref="dummyConnectionFactory" />  
  <property name="reconnectOnException">  
<value>true</value></property>  
</bean>
```