LastResourceCommitJms13

Last Resource Commit optimization for JMS

This feature appeared in BTM 1.3.3 and is not available in earlier versions.

> **Be Careful**
>
> There is one caveat with Last Resource Commit. There is a small chance that a transaction ends up with inconsistent results across participating resources if BTM crashes while a transaction is in-flight. The chance is small but it exists so be careful when using that feature. This is in *no way* a limitation of BTM but of the concept itself.
>
> Please note that if you only intend to run transactions against a single JMS server using Last Resource Commit this scenario is 100% safe, otherwise not.

In theory, only JMS servers supporting XA and providing a `javax.jms.XAConnectionFactory` implementation can be used with transaction managers. In practice, there is a way around this limitation.

The Last Resource Commit optimization (sometimes referred to as Last Resource Gambit or Last Agent optimization) allows a single non-XA resource (JMS server or database) to participate in a XA transaction by cleverly ordering the resources.

> **Maximum one non-XA resource**
>
> There can be at most one resource emulating XA with Last Resource Commit participating in a transaction. If it happens that you're trying to use a second emulating resource while one has already been used, BTM will throw an exception. Again, this is not a limitation of BTM but of the concept itself.

To enable it, you just have to create a `PoolingConnectionFactory` using the bitronix.tm.resource.jms.lrc.LrcXAConnectionFactory as the `XAConnectionFactory` implementation.

Here's an example of code configuring a ActiveMQ datasource:

```
PoolingConnectionFactory myConnectionFactory
= new PoolingConnectionFactory();
myConnectionFactory.setClassName("bitronix.t
m.resource.jms.lrc.LrcXAConnectionFactory");
myConnectionFactory.setUniqueName("amq-lrc")
;
myConnectionFactory.setMaxPoolSize(5);
myConnectionFactory.getDriverProperties().se
tProperty("connectionFactoryClassName",
"org.apache.activemq.ActiveMQConnectionFacto
ry");
myConnectionFactory.getDriverProperties().se
tProperty("properties.brokerUrl",
"tcp://localhost:61616");
```

and the same example viewed as a Resource Loader configuration

```
resource.ds.className=bitronix.tm.resource.j
ms.lrc.LrcXAConnectionFactory
resource.ds.uniqueName=amq-lrc
resource.ds.maxPoolSize=5
resource.ds.driverProperties.connectionFacto
ryClassName=org.apache.activemq.ActiveMQConn
ectionFactory
resource.ds.driverProperties.properties.brok
erUrl=tcp://localhost:61616
```