TEMENOS™

TAFJ-AS JBoss install 5.2 EAP

R14/R15

10/3/2015

Temenos

Amendment History:

Revision	Date Amended	Name	Description
1	8/12/2011	TAFJ team	Initial version
2	7st February 2012	H. Aubert	R12GA review
3	15th January 2013	JN. Charpin	R13GA review
4	13th February 2014	JN. Charpin	R14GA review
5	14th March 2014	JN. Charpin	Pooling review
6	19th March 2014	JN. Charpin	Servlet review
7	25th March 2014	JN. Charpin	Pooling review – Servlet review
8	15th May 2014	JN. Charpin	Remote JBoss messaging
9	5th February 2015	JN. Charpin	Memory JVM setting for UAT and production env.
10	6th March 2015	H. Aubert	R15 AMR review

Copyright

Errata and Comments

If you have any comments regarding this manual or wish to report any errors in the documentation, please document them and send them to the address below:
Technology Department

Temenos Headquarters SA
2 Rue de l'Ecole-de-Chimie,
CH - 1205 Geneva,
Switzerland

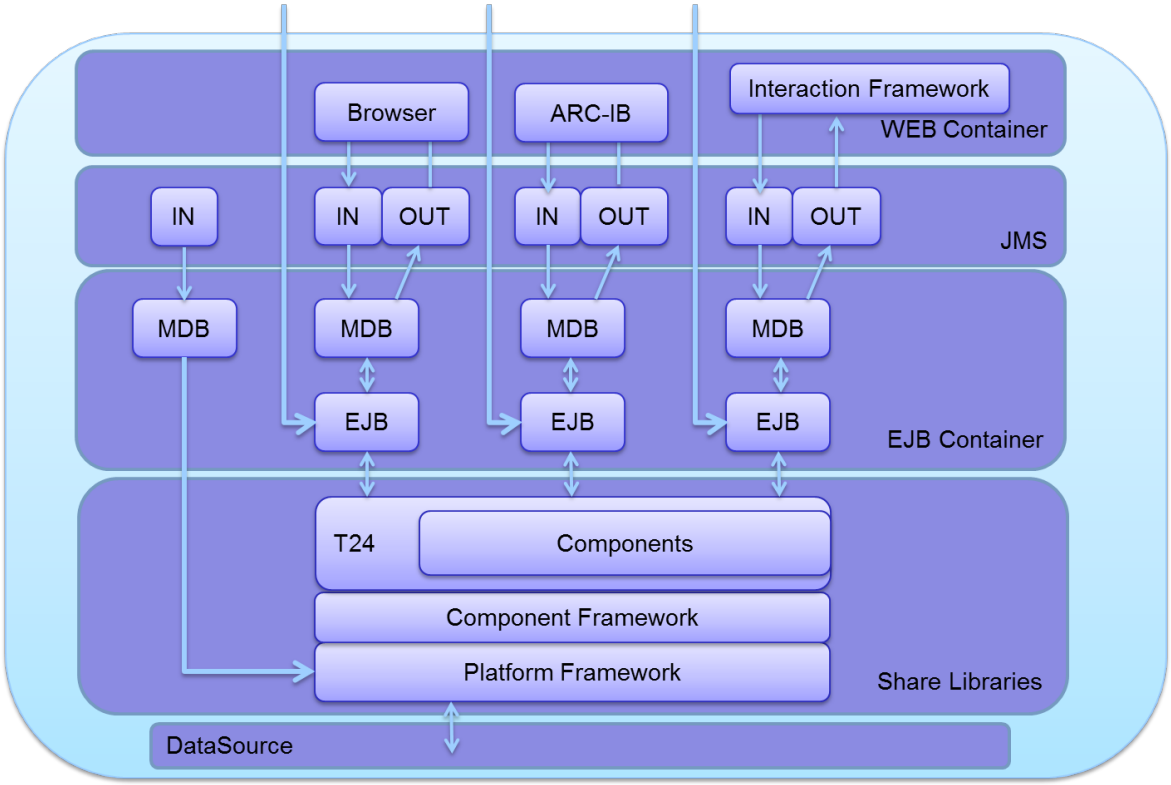
Tel SB: +41 (0) 22 708 1150
Fax: +41 (0) 22 708 1160

Please include your name, company, address, and telephone and fax numbers, and email address if applicable. TAFJdev@temenos.com

Table of Contents

T24 Java deployment in JBoss v5.2

Infrastructure



!! WARNING !!

The version of jBoss 5.2 EAP is depreciated for R15. It will not be supported with R16.

JBoss Installation

This document presents TAFJ deployment for JBoss EAP 5.2 EAP. Please note that JBoss 5.1 doesn't support remote EJB invocation. You will need EAP 5.2 for that purpose.

Get a zip file of jBOSS from TAFJDev@temenos.com or go to Red Hat website and jboss-eap-5.2.zip.

Unzip jBOSS to a directory. This directory **jboss-as** will be called in the document ***JBOSS_HOME***.

Prerequisite

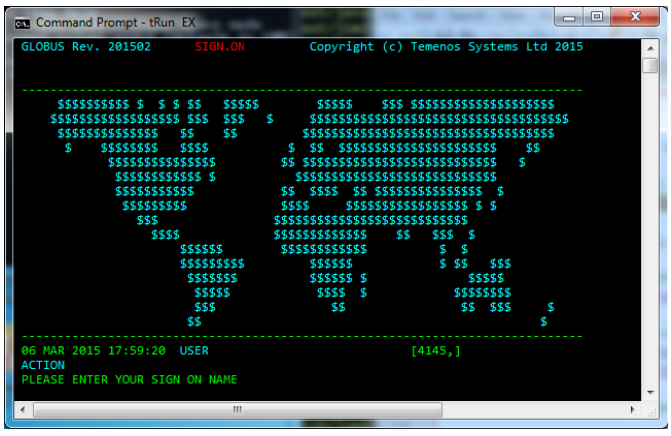
T24 precompiled jar files have to be in a folder that we call in the document ***T24_HOME***.

TAFJ JBOSS Configuration files are preconfigured to find T24 under a path relative to ***JBOSS_HOME*** : ${jboss.home.url}../../../T24/lib

TAFJ has to be installed in a folder that we call in the document ***TAFJ_HOME***.

TAFJ JBOSS Configuration files are preconfigured to find TAFJ under a path relative to ***JBOSS_HOME*** ${jboss.home.url}../../../TAFJ

Before configure JBoss with TAFJ/T24, check your standalone installation is working.



JBoss 5.2 eap configuration for T24 solution with TAFJ

Default memory setting

We need to increase the default memory configuration.

On Windows, in JBOSS_HOME\bin\run.bat change line :

```
set JAVA_OPTS=%JAVA_OPTS% -Xms128m -Xmx512m
```

with

```
set JAVA_OPTS=%JAVA_OPTS% -Xms256m –Xmx1024m -XX:MaxPermSize=256m
```

On Unix, in JBOSS_HOME\bin\run.conf change line 43:

```
JAVA_OPTS="-Xms128m -Xmx512m –Dsun …
```

With

```
JAVA_OPTS="-Xms256m –Xmx1024m -XX:MaxPermSize=256m –Dsun …
```

UAT / Production Memory setting

Please note that above memory setting is the minimal one to be able to deploy the application and run some agents to validate the configuration.

For a UAT and production environment you must refine this setting depending on your expected number of sessions and tSA.

A session/ tSA memory impact vary depending on the job done, it will be at least 20MB and could be up to 60MB.

When planning to run for example 80 interactive sessions and 40 tSA on a server, you could dimension the **max memory for sessions** by applying:

120 sessions * average 50MB = 6G max heap size.

You will have -Xmx6G

A good practice could be to set initial heap size to same value -Xms6G

If you are interested to monitor that more precisely, you could use a monitoring tool like Visual VM.

Take a heapdump when running the expected jobs, isolate the jSession objects and compute the retained size.

This is for one session.

Class Name	Instances [%]	Instances	Size	Retained
com.temenos.tafj.common.**jSession**		1 (0%)	691 (0%)	21,026,378 (19.2%)

This for 16 sessions.

Instances: 16 | Instance size: 691 | Total size: 11,056 | Retained size: 376,874,718	376,874,718 (74%)

You could also simply use TAFJ technical monitor to follow the memory evolution graph over the time and refine your setup.

Memory errors

Java.lang.OutOfMemoryError: **Java heap space**

Increase -Xmx max heap size parameter

Java.lang.OutOfMemoryError: **PermGen space**

Increase -XX:MaxPermSize max perm gen parameter

JVM parameters

Snapshot of the heap to analyse the content when getting a OOM error.

-XX:+HeapDumpOnOutOfMemoryError

-XX:HeapDumpPath=/some/path/

Garbage collection logs, detailed information about Garbage Collection, could be applied as there is a low overhead, display the amount of memory released

-XX:+PrintGC or -verbose:gc

Print messages at garbage collection, simple logging mode, i.e.

[GC 370562K->208870K(964096K), 0.0138438 secs]

[Full GC 174246K->81336K(853504K), 0.7733941 secs]

-XX:+PrintGCDetails

Same as above but print more details at garbage collection, differs depending on GC algorithm.

-Xloggc:<file> equivalent to -XX:+PrintGC -XX:+PrintGCTimeStamps

Log GC verbose output to specified file with time and date information

Environment and JMS configuration

Copy the file ***TAFJ_HOME*\appserver\jboss\jboss5eap\tafj-service.xml** in ***JBOSS_HOME*\server\default\deploy** folder.

This file configures:

1. The classpath. T24 jars, TAFJ jars, database drivers and additional jars you may need.
2. Mandatory system property tafj.home to specify ***TAFJ_HOME*** location.
3. Additional system property like file encoding…

Remark: If you don't install TAFJ, T24 precompiled file in the recommend folder please update the paths in your tafj-service.xml

These paths are relative to the JBOSS_HOME folder

i.e:

TAFJ_HOME = ${jboss.home.url}/../TAFJ_HOME means TAFJ is deployed at same level than JBOSS_HOME.

The library and JMS setup is done. Next step is the configuration of the database connection(s).

Datasource configuration

Copy the file **TAFJ_HOME\appserver\jboss\jboss5eap\t24_ds.xml** in **JBOSS_HOME\server\default\deploy** folder.

When you setup the datasource you have to use the drivers provided with the database. The drivers under %TAFJ_HOME %\DBDrivers are just helper. We cannot guaranty the drivers we provide with TAFJ are working with the specific database version you are using.

This file configures datasource resources. Two datasources need to be configured

1. One transactional datasource for T24 RDBMS.
2. second for TAFJ locking datasource when using locking mode JDBC or ORCL (this datasource is in autocommit mode and is not transactional).

Uncomment and set the database options which match your database server.

i.e.

For oracle:

```
<local-tx-datasource>
        <jndi-name>jdbc/t24DS</jndi-name>
        <connection-url>jdbc:oracle:thin:@[hostname]:[port]:[dbname]</connection-url>
        <driver-class>oracle.jdbc.driver.OracleDriver</driver-class>
        <user-name>[user]</user-name>
        <password>[password]</password>
        <min-pool-size>2</min-pool-size>
        <max-pool-size>10</max-pool-size>
        <idle-timeout-minutes>5</idle-timeout-minutes>
        <exception-sorter-class-
name>org.jboss.resource.adapter.jdbc.vendor.OracleExceptionSorter</exception-sorter-class-
name>
        <metadata>
                <type-mapping>Oracle11g</type-mapping>
        </metadata>
</local-tx-datasource>
```

Replace [hostname], [port], [dbname], [user], [password] respectively by hostname, port, ORACLE_SID, oracle user and oracle password for oracle connection.

Using the JDBC Locking Mechanism

Same as main T24 locking datasource except this one is a non-transactional datasource, see the red tag below.

```
<!-- JDBC Locking mecanism   -->
<no-tx-datasource>
      <jndi-name>jdbc/t24LockingDS</jndi-name>
       <!-- copy the params from t24DS   -->
</no-tx-datasource>
```

Copy the param you did setup for the transactional datasource

i.e

```
<!-- JDBC Locking mecanism   -->
<no-tx-datasource>
       <jndi-name>jdbc/t24LockingDS</jndi-name>
       <connection-url>jdbc:oracle:thin:@[hostname]:[port]:[dbname]</connection-url>
       <driver-class>oracle.jdbc.driver.OracleDriver</driver-class>
       <user-name>[user]</user-name>
       <password>[password]</password>
       <min-pool-size>2</min-pool-size>
       <max-pool-size>10</max-pool-size>
       <idle-timeout-minutes>5</idle-timeout-minutes>
       <exception-sorter-class-
name>org.jboss.resource.adapter.jdbc.vendor.OracleExceptionSorter</exception-sorter-class-
name>
       <metadata>
              <type-mapping>Oracle11g</type-mapping>
       </metadata>
</no-tx-datasource>
```

JMS Configuration

We will configure in this section JMS resources to use default JBoss messaging:

(1) default JMS queues used for Temenos products in reply/request pattern,

1. BROWSER,
2. ARC-IB,
3. SEAT injection,
4. CALL_AT subroutine,

(2) EXEC queue to launch T24 Phantom

(3) TEC topics for T24 Monitor purpose.

To configure JBoss messaging you have two different options:

1- Copy the file ***TAFJ_HOME*\appserver\jboss\jboss5eap\ tafj-destinations-service.xml** to ***JBOSS_HOME*\server\default\deploy\messaging folder.**

This file contains the basic JMS queues and topics you need. If you want to define other communication channels you will have to define new queues in either of the two files depending on the option you choose above.

***JBOSS_HOME*\server\default\deploy\messaging \tafj-destinations-service.xml**.

JBoss Messaging offers a service to persist messages in a database. The default database is HSQLDB which is part of JBoss 5 eap deployment. You can find configuration in ***JBOSS_HOME*\server\default\deploy\messaging \hsqldb-persistence-service.xml.**

Disable messaging persistence, do the following steps as described in the following Jboss thread:

http://community.jboss.org/thread/129340

Step 1: Delete the existing ***JBOSS_HOME*\server\default\deploy\messaging \hsqldb-persistence-service.xml.**

Step 2: Copy the file **TAFJ_HOME\appserver\jboss\jboss5eap\null-persistence-service.xml** in ***JBOSS_HOME*\server\default\deploy\messaging** folder.

Step 3: comment the following **messaging** security policy from the ***JBOSS_HOME*\server\default\deploy\messaging \messaging-jboss-beans.xml** file.

```
<!-- messaging application-policy definition -->
<application-policy xmlns="urn:jboss:security-beans:1.0" name="messaging">
<login-module code="org.jboss.security.auth.spi.DatabaseServerLoginModule" flag="required">
<module-option name="unauthenticatedIdentity">guest</module-option>
<module-option name="dsJndiName">java:/DefaultDS</module-option>
<module-option name="principalsQuery">SELECT PASSWD FROM JBM_USER WHERE USER_ID=?</module-
option>
<module-option name="rolesQuery">SELECT ROLE_ID, 'Roles' FROM JBM_ROLE WHERE USER_ID=?
</module-option>
</login-module>
</application-policy>
```

Step 4: Copy the contents **TAFJ_HOME\appserver\jboss\jboss5eap\login-config.xml** into file ***JBOSS_HOME*\server\default**conf\ login-config.xml

JBOSS settings (optional)

Remove unnecessary logging warnings

Change file ***JBOSS_HOME* /server/default/conf/jboss-log4j.xml**

Add

```
<!--  http://community.jboss.org/thread/149085 -->
<category
name="org.jboss.ejb3.interceptor.InterceptorInfoRepository">
       <priority value="ERROR"/>
</category>

<!-- http://community.jboss.org/message/571318  -->
<category name="org.jboss.injection.EJBRemoteHandler">
       <priority value="ERROR"/>
</category>

<!-- Limit the temenos to ERROR as its verbose -->
<category name="com.temenos">
     <priority value="ERROR"/>
</category>
```

TAFJ Application deployment

Copy ***TAFJ_HOME*\appserver\jboss\jboss5eap\TAFJJEE_EAR.ear** in **JBOSS_HOME\server\default\deploy** folder.

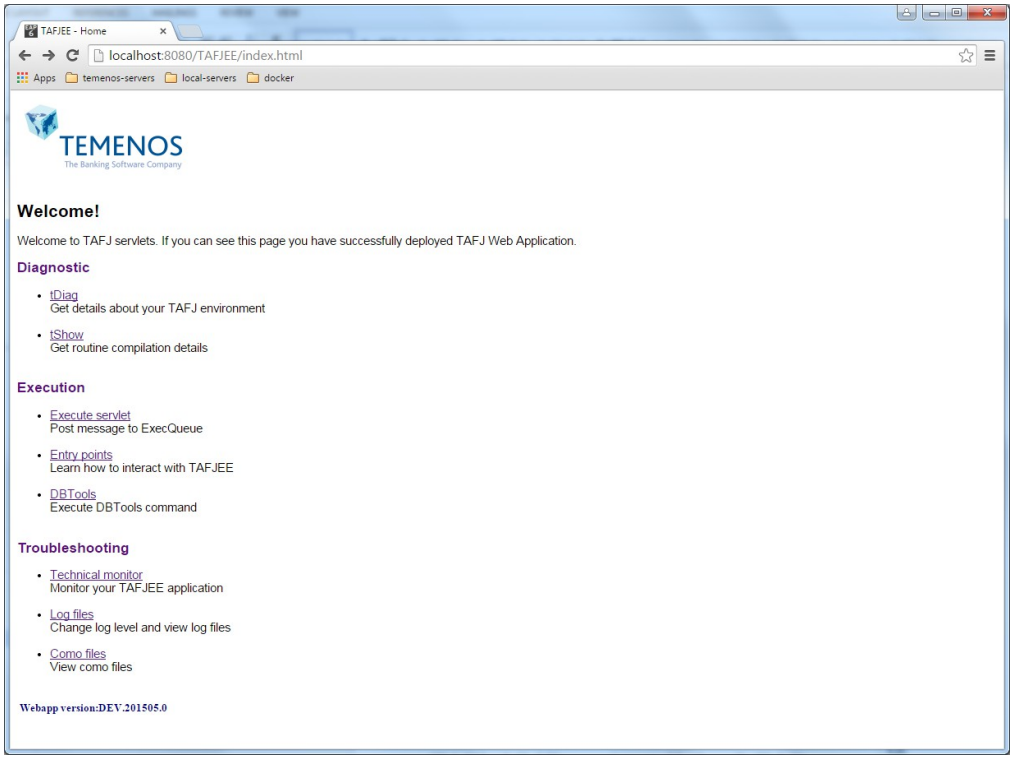
TAFJ Application deployment

Copy files ***TAFJ_HOME*\appserver\jboss\jboss6eap\TAFJJEE_EAR.ear**r under ***JBOSS_HOME*\standalone\deployments**.

TAFJJEE_EAR.ear file will set MDB (TAFJJEE_MDB.jar) and EJB (TAFJJEE_EJB.jar) to read message from JMS Queues, call T24 and publish response in reply queues.

It also contains a war file to deploy helper servlet.

http://localhost:8080/TAFJEE



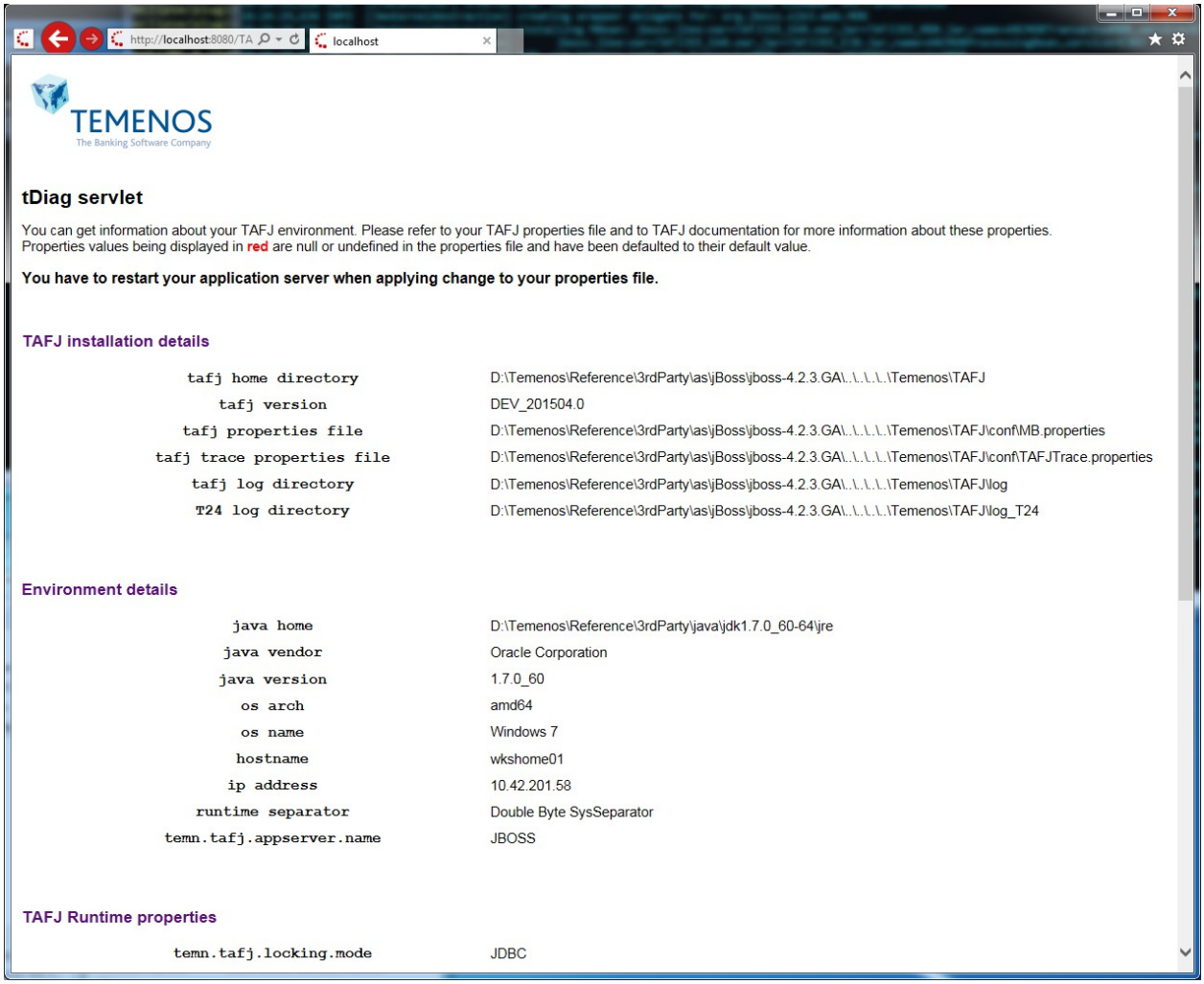
For more details please refer to the **TAFJ-AS TAFJ** documentation.

tDiag Servlet

TAFJJEE_WAR_TAFJ contains a servlet which could be used to get details about TAFJ installation

http://localhost:8080/TAFJEE/tDiag

i.e.



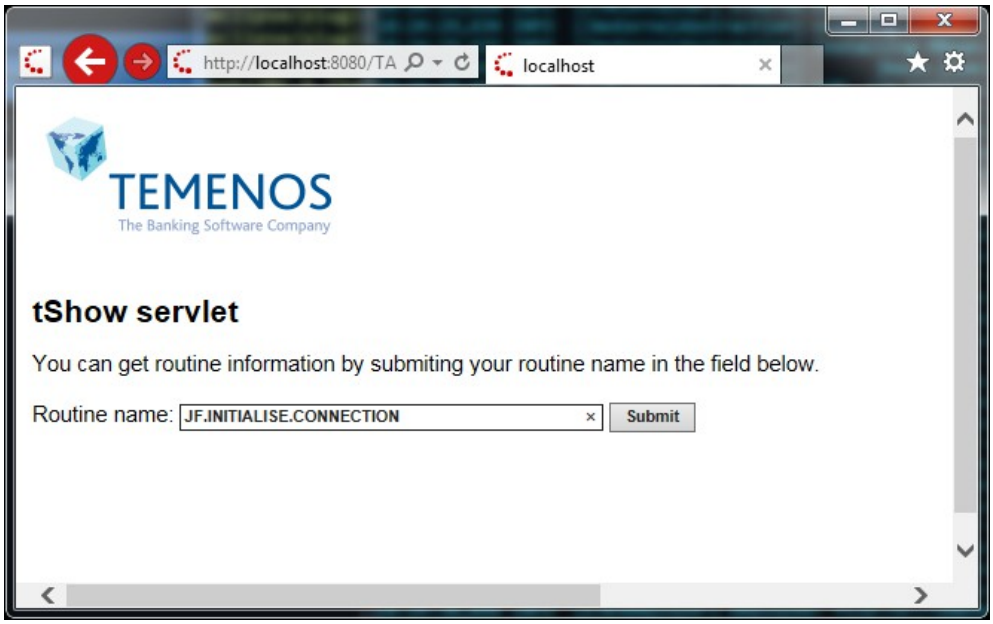
tShow Servlet

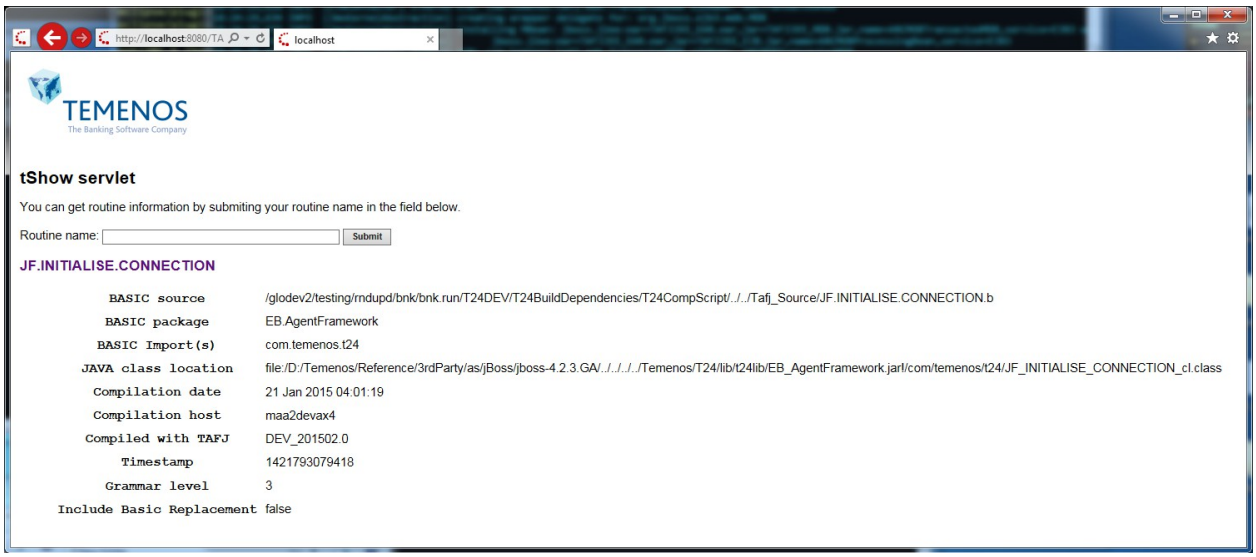
TAFJJEE_WAR_TAFJ contains a servlet which could be used to get compilation details about a specific routine

http://localhost:8080/TAFJEE/tShow

i.e.

to get details about JF.INITIALISE.CONNECTION

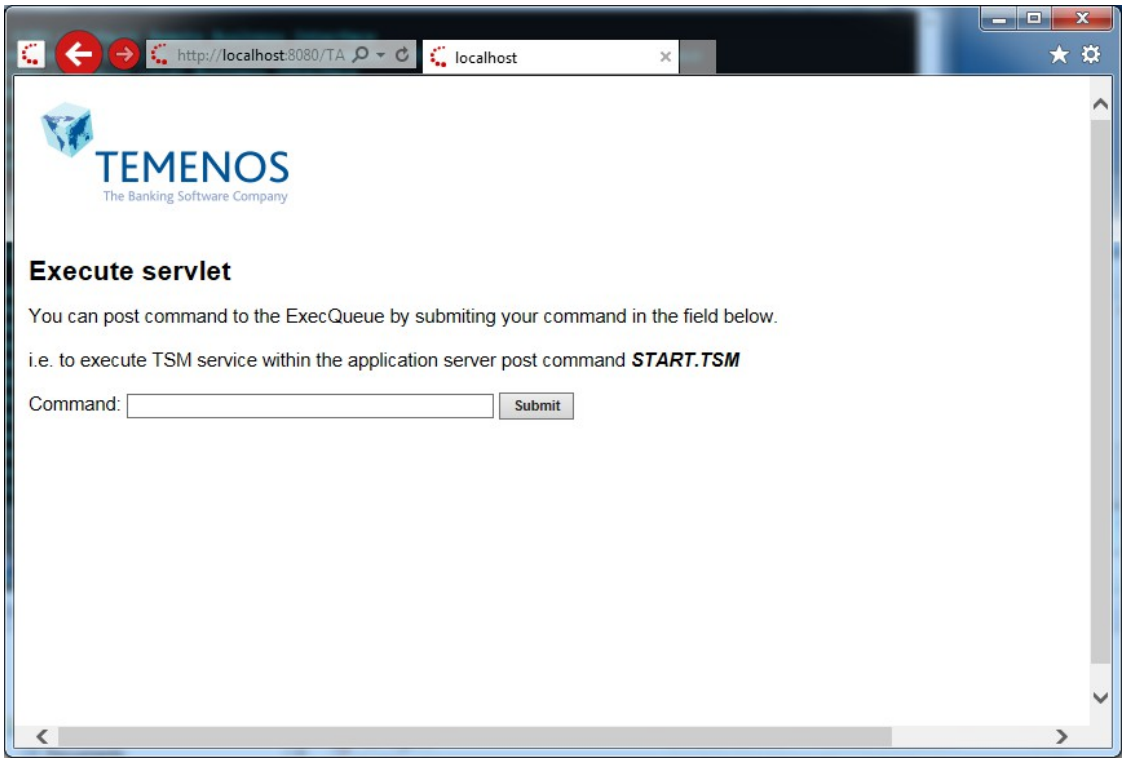




Execute Servlet

TAFJJEE_WAR_TAFJ contains a servlet which could be used to post message to the JMS queue t24ExeqQueue.

http://localhost:8080/TAFJEE/Execute



i.e

to post START.TSM submit START.TSM in the form.

Remark:

The following properties need to be set correctly to run TAFJ within an application server context.

temn.tafj.runtime.phantom.as.process = false

Other TAFJEE functionalities

TAFJEE application offers many other functionalities like monitoring, changing log level, como viewer… please refer to TAFJ AS documentation for detailed information.

T24 Application deployment

BROWSER Application

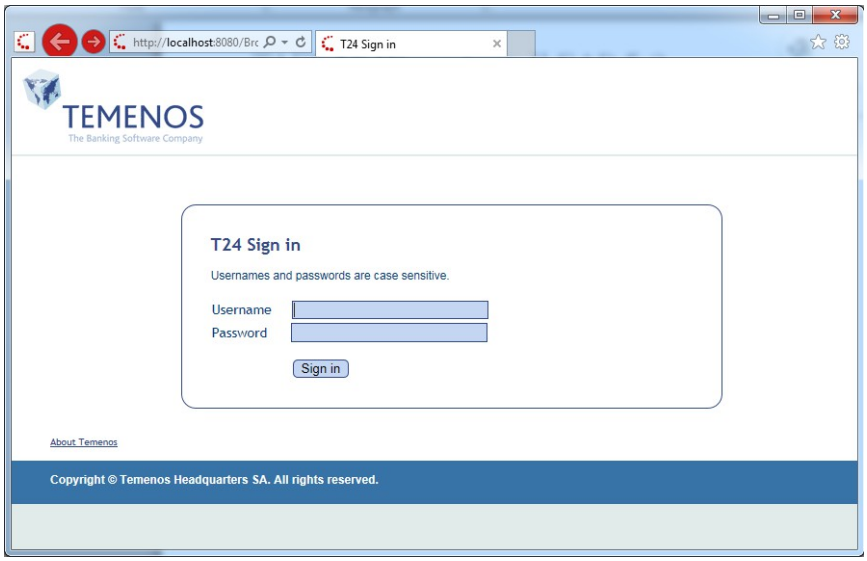
With regards to using Browser, you first need to deploy or unpack it to JBOSS_HOME\server\tafjcontext\deploy. Then you need to get rid of the JCA jar if it is deployed to your jboss (generally it won't be) and ensure that browserParameters.xml is set to JMS like below…

```
<parameter>
      <parameterName>Server Connection Method</parameterName>
      <parameterValue>JMS</parameterValue>
      <!-- Options: INSTANCE / AGENT / JMS -->
</parameter>
```

Add the file **jboss-classloading.xml** (from ***TAFJ_HOME* \appserver\jboss\jboss5eap**) to the **BrowserWeb.war/WEB-INF** directory if it isn't already there. This file is necessary to resolve classpath issues when deploying browser.

Browse servlet Browser

http://localhost:8080/BrowserWeb/servlet/BrowserServlet



Verify deployment

You can verify that you successfully configured all JEE resources by browsing http://localhost:8080/jmx-console/

For MDB/EJB deployment:

- ear=TAFJJEE_EAR.ear,jar=TAFJJEE_MDB.jar,name=ARCIBTransactedMDB,service=EJB3
- ear=TAFJJEE_EAR.ear,jar=TAFJJEE_MDB.jar,name=CallAtTransactedMDB,service=EJB3
- ear=TAFJJEE_EAR.ear,jar=TAFJJEE_MDB.jar,name=SEATTransactedMDB,service=EJB3
- ear=TAFJJEE_EAR.ear,jar=TAFJJEE_MDB.jar,name=TAFJPhantomListenerMDB,service=EJB3
- ear=TAFJJEE_EAR.ear,jar=TAFJJEE_MDB.jar,name=TWSTransactedMDB,service=EJB3
- ear=TAFJJEE_EAR.ear,jar=TAFJJEE_MDB.jar,name=TransactedMDB,service=EJB3
- ear=TAFJJEE_EAR.ear,jar=TAFJJEE_MDB.jar,service=EJB3
- ear=TAFJJEE_EAR.ear,jar=TAFJJEE_T24.jar,name=ARCIBProcessingBean,service=EJB3
- ear=TAFJJEE_EAR.ear,jar=TAFJJEE_T24.jar,name=CallAtProcessingBean,service=EJB3
- ear=TAFJJEE_EAR.ear,jar=TAFJJEE_T24.jar,name=MonitorBean,service=EJB3
- ear=TAFJJEE_EAR.ear,jar=TAFJJEE_T24.jar,name=OFSProcessingBean,service=EJB3
- ear=TAFJJEE_EAR.ear,jar=TAFJJEE_T24.jar,name=SEATProcessingBean,service=EJB3
- ear=TAFJJEE_EAR.ear,jar=TAFJJEE_T24.jar,name=TWSProcessingBean,service=EJB3
- ear=TAFJJEE_EAR.ear,jar=TAFJJEE_T24.jar,service=EJB3
- ear=profileservice-secured.jar,jar=profileservice-secured.jar,service=EJB3
- jar=profileservice-secured.jar,name=SecureDeploymentManager,service=EJB3
- jar=profileservice-secured.jar,name=SecureManagementView,service=EJB3
- jar=profileservice-secured.jar,name=SecureProfileService,service=EJB3
- service=EARClassLoaderDeployer
- service=EARDeployer

For Datasources:

- name='jboss-local-jdbc.rar',service=RARDeployment
- name='jboss-xa-jdbc.rar',service=RARDeployment
- name='jms-ra.rar',service=RARDeployment
- name='mail-ra.rar',service=RARDeployment
- name='quartz-ra.rar',service=RARDeployment
- name=DefaultDS,service=DataSourceBinding
- name=DefaultDS,service=LocalTxCM
- name=DefaultDS,service=ManagedConnectionFactory
- name=DefaultDS,service=ManagedConnectionPool
- name=DefaultJCAMetaDataRepository,service=JCAMetaDataRepository
- name=JmsXA,service=ConnectionFactoryBinding
- name=JmsXA,service=ManagedConnectionFactory
- name=JmsXA,service=ManagedConnectionPool
- name=JmsXA,service=TxCM
- name=jdbc/t24DS,service=DataSourceBinding
- name=jdbc/t24DS,service=LocalTxCM
- name=jdbc/t24DS,service=ManagedConnectionFactory
- name=jdbc/t24DS,service=ManagedConnectionPool
- name=jdbc/t24LockingDS,service=DataSourceBinding
- name=jdbc/t24LockingDS,service=LocalTxCM
- name=jdbc/t24LockingDS,service=ManagedConnectionFactory
- name=jdbc/t24LockingDS,service=ManagedConnectionPool
- service=CachedConnectionManager
- service=WorkManager
- service=WorkManagerThreadPool

For JMS Ressources:

- name=DLQ,service=Queue
- name=ExpiryQueue,service=Queue
- name=dummyQueue,service=Queue
- name=dummyReplyQueue,service=Queue
- name=t24ARCIBQueue,service=Queue
- name=t24ARCIBReplyQueue,service=Queue
- name=t24CALLATQueue,service=Queue
- name=t24CALLATReplyQueue,service=Queue
- name=t24EXECQueue,service=Queue
- name=t24OFSQueue,service=Queue
- name=t24OFSReplyQueue,service=Queue
- name=t24SEATQueue,service=Queue
- name=t24SEATReplyQueue,service=Queue
- name=t24TWSQueue,service=Queue
- name=t24TWSReplyQueue,service=Queue
- name=tecEvents,service=Topic

Trouble Shooting

On which port is jBoss listening?

At the end of the startup console you will see trace:

14:26:42,913 INFO [Http11Protocol] Starting Coyote HTTP/1.1 on http-127.0.0.1-8080

JBOSS is listening on port 8080.

I could browse jboss only on localhost:8080 or 127.0.0.1:8080?

Start jboss with the option -b 0.0.0.0

How to duplicate an environment for testing purpose?

Copy “default” folder where your TAFJ deployment is working in tafj_test_context folder for example under the parent directory of “default”. And start jboss with the option –c tafj_test_context

Where can I find the log files?

JBOSS log file are under JBOSS_HOME\server\tafjcontext\log folder

TAFJ log are under TAFJ_HOME\log folder.

COMO will be generated under TAFJ_HOME\como by default.

Appendixes

Appendix Remote JBoss Messaging server configuration

1- Define an external context mbean pointing to the remote JMS server provider in **server/deploy/messaging/jms-ds.xml** . Replace queuehost:1099 with appropriate host and port. The external context attribute" jndiName", will be used later as a prefix in JMS resources jndi mapping.

```
<mbean code="org.jboss.naming.ExternalContext"
name="jboss.jndi:service=ExternalContext,jndiName=remote">
      <attribute name="JndiName">remote</attribute>
      <attribute name="Properties">
            java.naming.factory.initial=org.jnp.interfaces.NamingContextFac
      tory
            java.naming.factory.url.pkgs=org.jboss.naming:org.jnp.interface
      s
            java.naming.provider.url=jnp://queuehost:1099
      </attribute>
      <attribute
name="InitialContext">javax.naming.InitialContext</attribute>
      <attribute name="RemoteAccess">false</attribute>
</mbean>
```

2- Define a Remote JMS provider in **server/deploy/messaging/jms-ds.xml .** This JMS provider depends on the external context previously defined. It will be used later by MDBs.

Note the "remote/" prefix usage, it refers to the external context jndiName defined above.

<depends> tag refers to the name of the external context mbean.

```
 <!-- Remote JMS Server-->
<mbean code="org.jboss.jms.jndi.JMSProviderLoader"

name="jboss.messaging:service=JMSProviderLoader,name=RemoteJMSProvider,serv
er=remote">
    <attribute name="ProviderName">RemoteJMSProvider</attribute>
    <attribute
name="ProviderAdapterClass">org.jboss.jms.jndi.JNDIProviderAdapter</attribu
te>
    <attribute
name="QueueFactoryRef">remote/XAConnectionFactory</attribute>
```

```
    <attribute
name="TopicFactoryRef">remote/XAConnectionFactory</attribute>
    <depends>jboss.jndi:service=ExternalContext,jndiName=remote</depends>
  </mbean>
```

3- Bind the Remote JMS Provider to MDBs by adding in **TAFJJEE_EAR/TAFJJEE_MDB.jar/META-INF/ejb-jar.xml** the following activation config property "providerAdapterJNDI".

```
<message-driven>

      <display-name>Transacted Listener MDB for OFS</display-name>

      <ejb-name>TransactedMDB</ejb-name>

      <mapped-name>jms/t24OFSQueue</mapped-name>

      <ejb-class>com.temenos.tafj.mdb.TransactedMDB</ejb-class>

      <messaging-type>javax.jms.MessageListener</messaging-type>

      <transaction-type>Container</transaction-type>

      <message-destination-type>javax.jms.Queue</message-destination-type>


<activation-config>

<activation-config-property>

<activation-config-property-name>providerAdapterJNDI</activation-config-
property-name>

<activation-config-property-value>java:/RemoteJMSProvider</activation-
config-property-value>

</activation-config-property>

</activation-config>
```

4- Bind the queue, the connection factory and the reply queue to the remote JMS server by editing **TAFJJEE_EAR/TAFJJEE_MDB.jar/META-INF/jboss.xml** to add the mapping to the external context.

```
<message-driven>

<ejb-name>TransactedMDB</ejb-name>

<destination-jndi-name>remote/queue/t24OFSQueue</destination-jndi-name>
```

```
…
<resource-ref>
<res-ref-name>jms/TAFJQueueConnectionFactory</res-ref-name>
<res-type>javax.jms.ConnectionFactory</res-type>
<jndi-name>remote/ConnectionFactory</jndi-name>
</resource-ref>
<message-destination-ref>
<message-destination-ref-name>jms/ReplyQueue</message-destination-ref-name>
<jndi-name>remote/queue/t24OFSReplyQueue</jndi-name>
</message-destination-ref>
<aop-domain-name>TAFJPooledMDB</aop-domain-name>
</message-driven>
```

Appendix OpenMQ configuration

One way to setup tafj queues is with OpenMQ. OpenMQ is part of the Glassfish distribution from Sun. You don't need to run Glassfish, but you could start the OpenMQ broker in standalone from Glassfish. Under GF_HOME\imq\bin launch imqbrokerd.exe

For more detail about OpenMQ please refer to: https://mq.dev.java.net/

To configure jBoss to use OpenMQ's Queue, do the following step:

1. Under GF_HOME\imq\lib copy file imqjmsra.rar under JBOSS_HOME\server\YOUR_CONTEXT\deploy
2. Copy from TAFJ_HOME\appserver\jboss files openmq-ds.xml and opnemq-queue-ds.xml in JBOSS_HOME\server\YOUR_CONTEXT\deploy
3. You need to update the descriptor files form TAFJJEE_EAR file:
    a. In TAFJJEE_EAR.ear\TAFJJEE_MDB.jar\META-INF\jboss.xml file:
        i. Uncomment all line with "`<resource-adapter-name>imqjmsra.rar</resource-adapter-name>`"
        ii. `Replace all "<jndi-name>java:/ConnectionFactory</jndi-name>" by "<jndi-name>OpenMQConnectionFactory</jndi-name>"`
        iii. `Replace all "queue/" by "phys_" by example <jndi-name>queue/t24OFSReplyQueue</jndi-name> by <jndi-name>phys_t24OFSReplyQueue</jndi-name>`
        iv. `Replace all "<activation-config-property-value>queue/" by "<activation-config-property-value>phys_"`

Restart JBoss.

To configure the openMQ RA to use a XA connection factory, update the ra.xml from imgjmsra.xml by changing:

In resourceadapter\outbound-resourceadapter\connection-definition\connectionfactory-interface change value **javax.jms.ConnectionFactory** by **javax.jms.XAConnectionFactory**

In opnemq-ds.xml by changing :connection-factories**no-tx-connection-factory**\connection-definition\ value **javax.jms. ConnectionFactory**

with

connection-factories**tx-connection-factory**\connection-definition\ **javax.jms.XAConnectionFactory**.

Appendix JBOSS thread pooling configuration v5.1

MDB Global (Strict) Pool Size

This value is set in the ejb3-interceptors-aop.xml under the section “Message Driven

Bean” as shown below :

```
<domain name="MessageDrivenBean">
<bind pointcut="execution(public*@javax.annotation.security.RunAs->*(..))">
<interceptor-ref name="org.jboss.ejb3.security.RunAsSecurityInterceptorFactory"/>
</bind>
<bind pointcut="execution(public**->*(..))">
<interceptor-ref name="org.jboss.ejb3.stateless.StatelessInstanceInterceptor"/>
<interceptor-ref name="org.jboss.ejb3.tx.TxInterceptorFactory"/>
<interceptor-ref name="org.jboss.ejb3.AllowedOperationsInterceptor"/>
<interceptor-ref name="org.jboss.ejb3.entity.TransactionScopedEntityManagerInterceptor"/>
<interceptor-ref name="org.jboss.ejb3.interceptor.EJB3InterceptorsFactory"/>
</bind>
<annotation expr="!class(@org.jboss.annotation.ejb.PoolClass)">
@org.jboss.annotation.ejb.PoolClass(value=org.jboss.ejb3.StrictMaxPool.class,maxSize=50,time
out=120000)
</annotation>
</domain>
```

This setting applies to every MDB instance, if you are looking for a finer configuration of pool per MDB/EJB, which we recommend, refer to next section. The timeout is after how many millisecond the App server will give up trying to get an instance from the pool.

Pool Size per MDB/EJB

In jboss.xml you could reference an entry in tafj-aop.xml file with <aop-domain-name>TAFJSEATPooledMDB</aop-domain-name> per MDB/ EJB.

```
<message-driven>
      <ejb-name>TAFJPhantomListenerMDB</ejb-name>
      <destination-jndi-name>queue/t24EXECQueue</destination-jndi-name>
      …

      <aop-domain-name>TAFJExecPooledMDB</aop-domain-name>
</message-driven>
```

The related pool size configuration is defined in tafj-aop.xml file as below

For MDB: \TAFJJEE_EAR.ear\TAFJJEE_MDB.jar\META-INF\ tafj-aop.xml

i.e.

```
<?xml version="1.0" encoding="UTF-8"?>
<aop xmlns="urn:jboss:aop-beans:1.0">

      <domain name="TAFJPooledMDB" extends="Message Driven Bean" inheritBindings="false">
        <bind pointcut="execution(public * *->*(..))">
        <interceptor-ref name="org.jboss.ejb3.security.AuthenticationInterceptorFactory"/>
        <interceptor-ref name="org.jboss.ejb3.security.RunAsSecurityInterceptorFactory"/>
```

```
      </bind>
      <!-- TODO: Authorization? -->
      <bind pointcut="execution(public * *->*(..))">
         <interceptor-ref name="org.jboss.ejb3.tx.CMTTxInterceptorFactory"/>
         <interceptor-ref name="org.jboss.ejb3.stateless.StatelessInstanceInterceptor"/>
         <interceptor-ref name="org.jboss.ejb3.tx.BMTTxInterceptorFactory"/>
         <interceptor-ref name="org.jboss.ejb3.AllowedOperationsInterceptor"/>
         <interceptor-ref
name="org.jboss.ejb3.entity.TransactionScopedEntityManagerInterceptor"/>
         <!-- interceptor-ref name="org.jboss.ejb3.interceptor.EJB3InterceptorsFactory"/ -->
         <stack-ref name="EJBInterceptors"/>
      </bind>
      <annotation expr="!class(@org.jboss.ejb3.annotation.Pool)">
         @org.jboss.ejb3.annotation.Pool (value="StrictMaxPool", maxSize=4, timeout=10000)
      </annotation>
       </domain>
```

…

```
<domain name="TAFJExecPooledMDB" extends="Message Driven Bean" inheritBindings="false">
         <bind pointcut="execution(public * *->*(..))">
         <interceptor-ref name="org.jboss.ejb3.security.AuthenticationInterceptorFactory"/>
         <interceptor-ref name="org.jboss.ejb3.security.RunAsSecurityInterceptorFactory"/>
      </bind>
      <!-- TODO: Authorization? -->
      <bind pointcut="execution(public * *->*(..))">
         <interceptor-ref name="org.jboss.ejb3.tx.CMTTxInterceptorFactory"/>
         <interceptor-ref name="org.jboss.ejb3.stateless.StatelessInstanceInterceptor"/>
         <interceptor-ref name="org.jboss.ejb3.tx.BMTTxInterceptorFactory"/>
         <interceptor-ref name="org.jboss.ejb3.AllowedOperationsInterceptor"/>
         <interceptor-ref
name="org.jboss.ejb3.entity.TransactionScopedEntityManagerInterceptor"/>
         <!-- interceptor-ref name="org.jboss.ejb3.interceptor.EJB3InterceptorsFactory"/ -->
         <stack-ref name="EJBInterceptors"/>
      </bind>
      <annotation expr="!class(@org.jboss.ejb3.annotation.Pool)">
         @org.jboss.ejb3.annotation.Pool (value="StrictMaxPool", maxSize=30,
timeout=200000)
      </annotation>
       </domain>
</aop>
```

For MDBs in the case of you define a pool size higher than 15 you also need to define an activation config property in: TAFJJEE_EAR.ear\TAFJJEE_MDB.jar\META-INF\ejb-jar.xml

Add or uncomment the following section to the corresponding activation-config section of the MDB, the value must match the maxSize pool value defined in tafj-aop.xml.

```
<activation-config-property>
<activation-config-property-name>maxSession</activation-config-property-
name>
<activation-config-property-value>30</activation-config-property-value>
</activation-config-property>
```

maxSession determines the amount of session/consumers the resource adapter will create to forward messages into the MDB (default is 15).

i.e.

```
<message-driven>
      <display-name>Phantom Executor MDB</display-name>
      <ejb-name>TAFJPhantomListenerMDB</ejb-name>
```

```
      <mapped-name>jms/t24EXECQueue</mapped-name>
      <ejb-class>com.temenos.tafj.mdb.PhantomListener</ejb-class>
      <messaging-type>javax.jms.MessageListener</messaging-type>
      <transaction-type>Bean</transaction-type>
      <message-destination-type>javax.jms.Queue</message-destination-type>
      <activation-config>
            <activation-config-property>
                  <activation-config-property-
                  name>acknowledgeMode</activation-config-property-name>
                  <activation-config-property-value>Auto-
                  acknowledge</activation-config-property-value>
            </activation-config-property>
            <activation-config-property>
                  <activation-config-property-name>maxSession</activation-
                  config-property-name>
                  <activation-config-property-value>30</activation-config-
                  property-value>
            </activation-config-property>
      </activation-config>
```

…

For EJB:

Same mapping between jboss.xml and tafj-aop.xml presented in MDB section could be done.

```
   <?xml version="1.0" encoding="UTF-8"?>
<aop xmlns="urn:jboss:aop-beans:1.0">
      <domain name="TAFJPooledEJB" extends="Stateless Bean"
            inheritBindings="false">
            <annotation expr="!class(org.jboss.ejb3.annotation.Pool)">
                  @org.jboss.ejb3.annotation.Pool (value="StrictMaxPool",
maxSize=4, timeout=10000)
      </annotation>
      </domain>
      <domain name="TAFJSEATPooledEJB" extends="Stateless Bean"
            inheritBindings="false">
            <annotation expr="!class(org.jboss.ejb3.annotation.Pool)">
                  @org.jboss.ejb3.annotation.Pool (value="StrictMaxPool",
maxSize=3, timeout=180000)
      </annotation>
      </domain>
</aop>
```

Appendix OPENMQ thread pooling configuration

Global (Strict) Pool Size

This value is set in the ejb3-interceptors-aop.xml under the section “Message Driven

Bean” like this :

```
<domain name="MessageDrivenBean">
<bind pointcut="execution(public*@javax.annotation.security.RunAs->*(..))">
<interceptor-ref name="org.jboss.ejb3.security.RunAsSecurityInterceptorFactory"/>
</bind>
<bind pointcut="execution(public**->*(..))">
<interceptor-ref name="org.jboss.ejb3.stateless.StatelessInstanceInterceptor"/>
<interceptor-ref name="org.jboss.ejb3.tx.TxInterceptorFactory"/>
<interceptor-ref name="org.jboss.ejb3.AllowedOperationsInterceptor"/>
<interceptor-ref name="org.jboss.ejb3.entity.TransactionScopedEntityManagerInterceptor"/>
<interceptor-ref name="org.jboss.ejb3.interceptor.EJB3InterceptorsFactory"/>
</bind>
<annotation expr="!class(@org.jboss.annotation.ejb.PoolClass)">
@org.jboss.annotation.ejb.PoolClass(value=org.jboss.ejb3.StrictMaxPool.class,maxSize=500,tim
eout=120000)
</annotation>
</domain>
```

This should contain the total amount of MDB instances you will have. The timeout is after how many millis the App server will give up trying to get an instance.

Pool Size of the different DataSource.

Here, we have to take 2 data sources in considerations : The JMS Broker and the Database

For the JMS Broker, the file is openmq-ds.xml

```
<connection-factories>
<no-tx-connection-factory>
<jndi-name>OpenMQConnectionFactory</jndi-name>
<rar-name>imqjmsra.rar</rar-name>
<use-java-context>false</use-java-context>
<connection-definition>javax.jms.ConnectionFactory</connection-definition>
<min-pool-size>20</min-pool-size>
<max-pool-size>500</max-pool-size>
<blocking-timeout-millis>15000</blocking-timeout-millis>
</no-tx-connection-factory>
</connection-factories>
For the Database, the file is t24-ds.xml
<datasources>
<local-tx-datasource>
<jndi-name>jdbc/t24DS</jndi-name>
<connection-url>jdbc:oracle:thin:@localhost:1521:R10GAMB</connection-url>
<driver-class>oracle.jdbc.driver.OracleDriver</driver-class>
<user-name>myuser</user-name>
<password>******</password>
<min-pool-size>20</min-pool-size>
<max-pool-size>500</max-pool-size>
<blocking-timeout-millis>15000</blocking-timeout-millis>
<idle-timeout-minutes>15</idle-timeout-minutes>
<exception-sorter-class-
name>org.jboss.resource.adapter.jdbc.vendor.OracleExceptionSorter</exception-sorter-
classname>
<metadata>
<type-mapping>Oracle11g</type-mapping>
</metadata>
</local-tx-datasource>
</datasources>
```

Pool Size per MDB

Each MDB pools are defined in the TAFJJEE_EAR.ear/TAFJJEE_MDB.jar/META-INF/ejb-jar.xml

Like this :

```
<message-driven>
<display-name>Queue1 Listener MDB</display-name>
<ejb-name>Queue1ListenerMDB</ejb-name>
<ejb-class>com.temenos.tafj.mdb.OFSListener</ejb-class>
<messaging-type>javax.jms.MessageListener</messaging-type>
<transaction-type>Bean</transaction-type>
<message-destination-type>javax.jms.Queue</message-destination-type>
<activation-config>
<activation-config-property>
<activation-config-property-name>EndpointPoolMaxSize</activation-config-property-name>
<activation-config-property-value>10</activation-config-property-value>
</activation-config-property>
<activation-config-property>
<activation-config-property-name>EndpointPoolSteadySize</activation-config-property-name>
<activation-config-property-value>2</activation-config-property-value>
</activation-config-property>
</activation-config>
</message-driven>
```

Appendix: JMX Configuration

If you want to see all jBoss information and because jBoss is embedded in a Mbean server, you need to add this parameter to the JAVA_OPTS variable:

```
JAVA_OPTS="$JAVA_OPTS -Djboss.platform.mbeanserver
-Djavax.management.builder.initial=org.jboss.system.server.jmx.MBean
ServerBuilderImpl"
```

Appendix: Setting up multiple instances of JBoss on the same machine with The ServiceBindingManager

http://community.jboss.org/wiki/ConfigurePorts

Easy Way

Start JBoss with the VM parameter jboss.service.binding.set with either:

-ports-default

- ports-01

- ports-02

Edit the JBOSS configuration file corresponding to your operating system.

jboss-as\bin\ run.conf.bat or jboss-as\bin\ run.conf

and add

:JAVA_OPTS_SET

set "JAVA_OPTS=%JAVA_OPTS% -Djboss.service.binding.set=ports-01

Harder Way
Edit server/default/conf/bindingservice.beans/META-INF/bindings-jboss-beans.xml
Change "ports-default" to the set of port values you want to use.

<!-- The name of the set of bindings to use for this server -->

<parameter>${jboss.service.binding.set:ports-default}</parameter>