Integrating an Oracle Service Bus Clustered Domain with a Remote TIBCO Enterprise Message Service 4.2.0/4.3.0 for Messaging Applications in a Service-Oriented Architecture

An Oracle White Paper
Updated January 2009

Integrating an Oracle Service Bus Clustered Domain with a Remote TIBCO Enterprise Message Service 4.2.0/4.3.0 for Messaging Applications in a Service-Oriented Architecture

Integrating an Oracle Service Bus Clustered Domain and a Remote TIBCO Enterprise Message Service 4.2.0/4.3.0 for Messaging Applications in a Service-Oriented Architecture

INTRODUCTION

Java Message Service (JMS) in Oracle Service Bus is Oracle WebLogic's implementation of the JMS 1.1 specification. It is an enterprise-class messaging system that not only fully supports the JMS 1.1 specification, but also provides numerous extensions to the standard JMS APIs. Because JMS functionality in Oracle Service Bus is tightly integrated into the Oracle WebLogic Server platform, customers can build highly secure Java enterprise applications that can be monitored and administered through the Oracle Service Bus console. In addition to fully supporting extended architecture (XA) transactions, JMS in Oracle Service Bus offers high availability through its clustering and server migration features. It also provides seamless interoperability with third-party messaging vendors.

This white paper describes the messaging communication between a messaging client and the integrated system of an Oracle Service Bus clustered domain and a remote TIBCO EMS 4.2.0/4.3.0.

This white paper describes the communication between a messaging client and an integrated system of Oracle Service Bus clustered domain and remote TIBCO Enterprise Message Service (EMS) 4.2.0/4.3.0 within a service-oriented architecture. It includes a detailed description of the configuration for request/response and one-way design patterns. In the case of request/response, the white paper discusses response correlation by request message ID, as well as by request correlation ID. It distinguishes Oracle Service Bus configurations for point-to-point and publish-subscribe messaging. The configuration implies that Oracle Service Bus provides distributed transactional management. Furthermore, the white paper covers the cases when only proxy services or business services are hosted on TIBCO EMS, as well as the two together.

PROBLEM DEFINITION: HOW TO INTEGRATE ORACLE SERVICE BUS WITH TIBCO ENTERPRISE MESSAGE SERVICE

The goal is to connect the remote TIBCO EMS to Oracle Service Bus using the JMS interface of TIBCO EMS. The user can choose to use the TIBCO Rendezvous native protocol for business processes. TIBCO EMS will then be connected to TIBCO Rendezvous. The integrated system has to have distributed transactional integrity, reliability, and scalability. This white paper discusses only integration between Oracle Service Bus and TIBCO EMS.

DEPLOYMENT ARCHITECTURE

In general, the details of the installation and configuration depend on the operating system, the number of machines, and the desired cluster sizes of Oracle WebLogic Server domains. Typically in a production environment, the client, the Oracle Service Bus, and the TIBCO EMS are hosted on separate machines. In addition, each Oracle WebLogic Server cluster node is hosted on a separate hardware system.

The following sections describe simplified examples of how to set up and configure request/response messaging among a client, an Oracle Service Bus, and a remote TIBCO EMS for testing.

The following sections describe simplified examples of how to set up and configure request/response messaging among a client, an Oracle Service Bus, and a remote TIBCO EMS for testing. The examples use minimum hardware—two systems, each running a Microsoft Windows operating system. The first machine hosts the client and Oracle Service Bus. The second machine hosts TIBCO EMS. In general, the client can be hosted anywhere on the network.

SETUP

We'll discuss the setup variations listed below.

1. Oracle Service Bus configurations are distinguished by proxy and business service hosting.

    - The business service is deployed on the TIBCO EMS JMS provider, and the proxy service is deployed on the Oracle WebLogic JMS provider. That means that TIBCO EMS hosts outbound JMS destinations, and Oracle WebLogic Server hosts inbound JMS destinations. The request message passes from the JMS client, through Oracle WebLogic Server, to TIBCO EMS. The response message, if sent, passes in the opposite direction—from TIBCO EMS, through Oracle WebLogic Server, and back to the client.

    - The proxy service is deployed on the TIBCO EMS JMS provider, and the business service is deployed on the Oracle WebLogic JMS provider. That means that TIBCO EMS hosts inbound JMS destinations, and Oracle WebLogic Server hosts outbound JMS destinations. The request message passes from the JMS client, through TIBCO EMS, to Oracle WebLogic Server. The response message, if sent, passes in the opposite direction—from Oracle WebLogic Server, through TIBCO EMS, and back to the client.

    - Both proxy and business services are deployed on the TIBCO EMS JMS provider. That means that TIBCO EMS hosts both inbound and outbound JMS destinations. The request message passes from the JMS client, through the inbound TIBCO EMS destination, to the outbound TIBCO EMS destination. The response message, if sent, passes in the opposite direction—from the outbound TIBCO EMS destination, through the inbound TIBCO EMS destination, and back to the client.

2. Oracle Service Bus configurations are distinguished by their response message correlation pattern.
   - **Request/response messaging with response correlation by request JMS correlation ID.** The business application has to set Response JMSCorrelationID to request JMSCorrelationID before posting the response message to the outbound response destination.
   - **Request/response messaging with response correlation by request JMS message ID.** The JMS client has to set inbound response uniform resource identifiers as reply-to properties in the inbound request message. The business application has to set Response JMSCorrelationID to request JMSMessageID before posting the response message to the outbound response destination. The Oracle Service Bus reads the outbound response URI specified as a reply-to property in the outbound request message.
3. Oracle Service Bus configurations are distinguished by point-to-point and publish-subscribe messaging: point-to-point request/response messaging and publish-subscribe one-way messaging.

Up to and including Oracle Service Bus 2.1, only response correlation by request JMSCorrelationID was supported. Starting with Oracle Service Bus 2.5, response correlation by JMSMessageID is also supported.

INSTALLATION AND CONFIGURATION

We will now discuss how to install and configure TIBCO EMS. The configuration includes the creation of JMS objects.

We will now discuss how to install and configure TIBCO EMS. The configuration includes the creation of JMS objects. To concentrate the JMS object creation task, we'll create all the objects for all configurations ahead of time. Below are the suggested steps.

1. Install TIBCO EMS on one machine and Oracle Service Bus on a second machine.
2. Start TIBCO EMS from the Windows Start menu:

   ```
   Start->Programs->TIBCO Enterprise Message Service->Start EMS Server
   from the Windows Start menu.
   ```

3. Start the TIBCO EMS administration tool from the Windows Start menu:

   ```
   Start->Programs->TIBCO Enterprise Message Service->Start EMS
   Administration Tool.
   ```

4. Create JMS objects in the TIBCO EMS server. You do that by entering commands in the TIBCO EMS administration tool command window.
5. The set of objects will differ depending on whether you want to configure point-to-point request/response or publish-subscribe messaging.

6. You start by connecting to TIBCO EMS:

```
> connect <external Tibco EMS server host>:7222
```

7. You then create a JMS connection factory. Depending on your scenario, it could be a queue or topic connection factory, supporting global transactions (signified by XA) or without support for global transactions.

```
> create factory XAQueueConnectionFactory xaqueue

> create factory XATopicConnectionFactory xatopic

> create factory QueueConnectionFactory queue

> create factory TopicConnectionFactory topic
```

8. If you plan to have request/response with response correlation by request JMSCorrelationID, to follow the example, you will need the following request/response queues.

    8.1. For a business service deployed on TIBCO EMS:

    ```
    > create queue TibcoCorIDRequestQueue

    > create queue TibcoCorIDResponseQueue
    ```

    8.2. For a proxy service deployed on TIBCO EMS:

    ```
    > create queue TibcoCorIDProxyRequestQueue

    > create queue TibcoCorIDProxyResponseQueue
    ```

9. If you plan to have request/response with response correlation by request JMSMessageID, to follow the example, you will need request/response queues.

    9.1. For a business service deployed on TIBCO EMS:

    ```
    > create queue TibcoRequestQueue

    > create queue TibcoResponseQueue0

    > create queue TibcoResponseQueue1
    ```

    9.2. For a proxy service deployed on TIBCO EMS:

    ```
    > create queue TibcoProxyRequestQueue

    > create queue TibcoProxyResponseQueue
    ```

10. If you plan to have publish-subscribe messaging, to follow the example, you will need to create topics.

    10.1. For a business service deployed on TIBCO EMS:

    ```
    > create topic TibcoRequestTopic
    ```

10.2. For a proxy service deployed on TIBCO EMS:

```
> create topic TibcoProxyRequestTopic
```

11. NOTE: The following TIBCO EMS configuration modification is required for any scenario that uses a remote (external) TIBCO EMS server.

    During the process of integrating Oracle WebLogic Server and TIBCO EMS, it was discovered that the application was not connecting properly to the JMS queues. Error messages were being logged in the Oracle WebLogic Server application stating that no connection to 'localhost' was available:

    ```
    "NestedException Message is: Failed to connect to the server at
    tcp://localhost:7222>"
    ```

    The problem was that the default URL property for a queue connection factory in TIBCO EMS was invalid for scenarios using an external server. This can be fixed by modifying the configuration on the TIBCO EMS server in the following manner:

    - Log in to the TIBCO EMS administration console.
    - Add the appropriate URL property to the newly created factory.

    ```
    addprop factory XAQueueConnectionFactory url=tcp://<ems-
    server>:7222

    addprop factory XATopicConnectionFactory url=tcp://<ems-
    server>:7222
    ```

    This configuration change is reflected in the factories.conf file located in one of the following directories: *C:\tibco\ems\bin* or */opt/tibco/ems/bin*. When this modification is complete, the proper connection factory URL is returned when Oracle Service Bus connects to TIBCO EMS.

12. Configure the Oracle Service Bus clustered domain, optionally, in production mode with Oracle Database and with two colocated managed servers.

13. Edit the setDomainEnv.cmd file to include at least the tibjms.jar file, and, if needed, other JARs in the classpath for Oracle WebLogic Server. For example:

    ```
    set TIBCO_EMS_DIR=C:\tibco\ems\clients\java

    set

    PRE_CLASSPATH=%PRE_CLASSPATH%;%TIBCO_EMS_DIR%\ti
    bcrypt.jar;%TIBCO_EMS_DIR%\tibjms.jar;

    %TIBCO_EMS_DIR%\tibjmsadmin.jar;%TIBCO_EMS_DIR%\tibjmsa
    pps.jar;%TIBCO_EMS_DIR%\tibrvjms.jar
    ```

14. Make sure that the TIBCO EMS server (tibemsd) is running.

15. Create a new JMS system module on the Oracle WebLogic Server cluster.

16. In this JMS system module, configure a foreign JMS server with a foreign connection factory and destinations, an XA connection factory if you need a global transaction, and uniform distributed destinations (UDDs).

17. Below is the example of JMS module configuration settings.

17.1. Foreign JMS server

Name: “ForeignServerTibco”

JNDI Initial Context Factory:

"com.tibco.tibjms.naming.TibjmsInitialContextFactory"

JNDI Connection URL:
"tibjmsnaming://<hostname>:7222"sdafsdf

17.2. Foreign connection factories

17.2.a. Queue connection factory:

Name: “TIBCO XAQueueConnectionFactory”

Local JNDI Name: “TIBCO.xaqcf”

Remote JNDI Name: “XAQueueConnectionFactory”

17.2.b. Topic connection factory:

Name: “TIBCO XATopicConnectionFactory”

Local JNDI Name: “TIBCO.xatcf”

Remote JNDI Name: “XATopicConnectionFactory”

17.3. Foreign destinations

17.3.a. Response correlation by request JMSCorrelationID

Outbound request queue:

Name: “TIBCO JMSQueue TibcoCorIDRequestQueue:

Local JNDI Name: “TIBCO.TibcoCorIDRequestQueue”

Remote JNDI Name: “TibcoCorIDRequestQueue”

Outbound response queue:

Name: “TIBCO JMSQueue TibcoCorIDResponseQueue”

Local JNDI Name: “TIBCO.TibcoCorIDResponseQueue”

Remote JNDI Name: “TibcoCorIDResponseQueue”

Inbound request queue:

Name: “TIBCO JMSQueue TibcoCorIDProxyRequestQueue”

Local JNDI Name: “TIBCO.TibcoCorIDProxyRequestQueue”

Remote JNDI Name: “TibcoCorIDProxyRequestQueue”

Inbound response queue:

Name: “TIBCO JMSQueue TibcoCorIDProxyResponseQueue”

Local JNDI Name: “TIBCO.TibcoCorIDProxyResponseQueue”

Remote JNDI Name: “TibcoCorIDProxyResponseQueue”

17.3.b. Response correlation by request JMSMessageID

Outbound request queue:

Name: “TIBCO JMSQueue TibcoRequestQueue:

Local JNDI Name: “TIBCO.TibcoRequestQueue”

Remote JNDI Name: “TibcoRequestQueue”

Two outbound response queues—one for each of the two cluster servers:

Name: “TIBCO JMSQueue TibcoResponseQueue0”

Local JNDI Name: “TIBCO.TibcoResponseQueue0”

Remote JNDI Name: “TibcoResponseQueue0”

Name: “TIBCO JMSQueue TibcoResponseQueue1”

Local JNDI Name: “TIBCO.TibcoResponseQueue1”

Remote JNDI Name: “TibcoResponseQueue1”

Inbound request queue:

Name: “TIBCO JMSQueue TibcoProxyRequestQueue:

Local JNDI Name: “TIBCO.TibcoProxyRequestQueue”

Remote JNDI Name: “TibcoProxyRequestQueue”

Inbound response queue:

Name: “TIBCO JMSQueue TibcoProxyResponseQueue”

Local JNDI Name: “TIBCO.TibcoProxyResponseQueue”

Remote JNDI Name: “TibcoProxyResponseQueue”

17.4. Outbound request topic for publish-subscribe

Name: “TIBCO JMSTopic TibcoRequestTopic”

Local JNDI Name: “TIBCO.TibcoRequestTopic”

Remote JNDI Name: “TibcoRequestTopic”

17.5. Inbound request topic for publish-subscribe

Name: “TIBCO JMSTopic TibcoProxyRequestTopic”

Local JNDI Name: “TIBCO.TibcoProxyRequestTopic”

Remote JNDI Name: “TibcoProxyRequestTopic”

18. Create a new XA connection factory, optionally, with default delivery mode set to persistent. For example:

“weblogic.alsb.jms.tibco.ConnectionFactoryXA”

19. For proxy and business services deployed on a Oracle WebLogic Server clustered domain, create UDDs, uniform distributed queues (UDQs), and uniform distributed topics (UDTs) deployed on cluster servers. Below are examples of Java Naming and Directory Interface (JNDI) names.

19.1. Inbound request UDQs

19.1.a. For response correlation by message ID:

“TibcoGatewayUDQReq”

19.1.b. For response correlation by correlation ID:

“TibcoCorIDGatewayUDQReq”

19.2. Inbound response UDQ

“TibcoGatewayUDQRes”

Optionally:

“InvalidMsgTibcoQ”

19.3. Inbound publish UDTs

“TibcoGatewayUDTopicReq”

Optionally

“InvalidMsgTibcoTopic”

20. Verify that the Oracle WebLogic Server JNDI tree shows all foreign JMS server objects (connection factories and queues).

DEPLOYMENT SCHEMES

Once we configure Oracle WebLogic Server by creating all participating JMS objects, we will configure Oracle Service Bus for various deployment combinations of JMS proxy and business services.

We'll create three projects, and each project will include

- Request/response with correlation by request message ID
- Request/response with correlation by request correlation ID
- Publish-subscribe

Project #1: Deploying a Java Message Service Proxy Service on Oracle WebLogic Server and Java Message Service Business Services on TIBCO Enterprise Message Service

Our first project will include three applications: request/response with response correlation by request JMS ID; request/response with response correlation by request JMS correlation ID; and publish-subscribe.

Suppose we need to deploy a JMS proxy service on Oracle WebLogic Server and JMS business services on TIBCO EMS. This will be our first project and it will include three applications: request/response with response correlation by request JMS ID; request/response with response correlation by request JMS correlation ID; and publish-subscribe.

For the request/response application with response correlation by request JMS ID, we'll indicate that the JMS proxy service requires a response, and we'll specify an inbound request destination in the Oracle Service Bus console. The inbound response destination will be read by Oracle Service Bus code from the reply-to property that you have to set on your inbound request message.

We'll also specify outbound request/response destinations in the Oracle Service Bus console. The outbound request destination will be the TIBCO EMS request queue, and outbound response destinations will be multiple TIBCO EMS response queues—as many as there are Oracle WebLogic cluster servers on which a foreign JMS server is deployed.

A business application should read the JMS ID of the outbound request message and set the JMS correlation ID of the outbound response message to that value.

Oracle Service Bus will set reply-to properties on the outbound request messages corresponding to the TIBCO EMS outbound response destinations. A business application should send the outbound response messages to the outbound response destinations extracted as reply-to properties from outbound request messages. A business application should also read the JMS ID of the outbound request message and set the JMS correlation ID of the outbound response message to that value.

For the request/response application with response correlation by request JMS correlation ID, we'll indicate that the JMS proxy service requires a response, and we'll specify inbound request and response destinations in the Oracle Service Bus console.

We'll also specify outbound request/response destinations in the Oracle Service Bus console. The outbound request/response destinations will be TIBCO EMS request/response queues.

Oracle Service Bus will set reply-to properties on the outbound request messages corresponding to the TIBCO EMS outbound response destinations specified in the Oracle Service Bus console at design time. A business application should read the JMS correlation ID of the outbound request message and set the JMS correlation ID of the outbound response message to that value.

The JMS proxy service does not require a response for the publish-subscribe application. We'll specify only an inbound request destination in the Oracle Service Bus console. We'll also specify an outbound request destination in the Oracle Service Bus console. The outbound request destination will be the TIBCO EMS request topic.

When proxy and business services are subscribed to a topic, to ensure that the subscriber receives only one copy of the published message, a proxy service must be deployed on only one of the cluster servers. Use the advanced transport setting to target the desired server in a cluster.

Project #2: Deploying a Java Message Service Proxy Service on TIBCO Enterprise Message Service and Java Message Service Business Services on Oracle WebLogic Server

Project #3: Deploying a Java Message Service Proxy Service and Java Message Service Business Services on TIBCO Enterprise Message Service

These two configuration groups—deploying a JMS proxy service on TIBCO EMS and JMS business services on Oracle WebLogic Server, and deploying a JMS proxy service and JMS business services on TIBCO EMS—differ by whether Oracle WebLogic Server or TIBCO EMS hosts the destinations and provides the connection factories.

For each pair of proxy and business services, we'll configure a pipeline between the JMS proxy service and the JMS business service, as well as routing to the business service.

For each pair of proxy and business services, we'll configure a pipeline between the JMS proxy service and the JMS business service, as well as routing to the business service. Optionally, we'll configure message header logging as the request/response actions for request/response applications, or only request actions for publish-subscribe applications. The logging will show how many request/response messages would be passing the request/response pipelines of each managed server.

You can write and use a JMS business service as a Java client based on JMS 1.1 specification APIs. This business service gets the request message from the request queue and puts the response message in the response queue. Depending on the chosen pattern, set the correlation ID on the response message to the request correlation ID or the request message ID.

TESTING

Let's first look at the request/response scenario with a proxy service deployed on the Oracle WebLogic Server JMS and a business service deployed on TIBCO EMS.

- When the TIBCO EMS server and all Oracle WebLogic cluster servers are alive and connected, the number of responses is equal to the number of requests.
- When the TIBCO EMS server is alive but one of the Oracle WebLogic cluster servers in the Oracle Service Bus cluster is stopped and restarted, you should see how the Oracle Service Bus cluster server that stayed alive receives response messages for each request message that it sent. And when you start the Oracle Service Bus cluster server that was stopped, you should see how the Oracle Service Bus cluster server that was shut down receives the response messages and discards them. Response messages received by the restarted server are invalid and discarded because the in-memory message mapping is gone when the server is shut down.
- When the TIBCO EMS server is shut down and restarted, the outcome for version 2.1 depends on the timing of the restart. If the message sending retry comes when the TIBCO EMS server is restarted, the client will receive both error messages due to the TIBCO EMS being down, and a valid response when the TIBCO EMS becomes functional. This behavior is changed for version 2.5 with 'quality of service' set to 'exactly once' to ensure only one response, whether of failure or success.

Now, let's look at the request/response scenario with a proxy service deployed on TIBCO EMS and a business service deployed on the Oracle WebLogic Server JMS.

- When the TIBCO EMS server is alive but one of the Oracle WebLogic cluster servers in the Oracle Service Bus cluster is stopped, all outbound request messages should be received and all outbound responses should be sent.
- When the TIBCO EMS server is shut down and restarted, the outcome depends on the timing of the restart. If the inbound destination is configured with retries, and the retry comes when the TIBCO EMS server becomes functional, the inbound request will be routed to the business service and the inbound response will come to the inbound TIBCO EMS response queue.

CONCLUSION

This white paper describes integrated messaging system comprised of the clustered Oracle Service Bus and remote TIBCO EMS. It discusses a variety of server configurations for the messaging applications within a service-oriented architecture.

APPENDIX: REFERENCES AND RELATED DOCUMENTS

- **Integrating Remote JMS Providers**
  e-docs.bea.com/wls/docs81/faq/interop.html
- **Configure Foreign Servers**
  e-docs.bea.com/wls/docs100/ConsoleHelp/taskhelp/
  jms_modules/foreign_servers/ConfigureForeignServers.html and
  e-docs.bea.com/wls/docs103/ConsoleHelp/taskhelp/
  jms_modules/foreign_servers/ConfigureForeignServers.html
- **TIBCO Enterprise Message Service APIs**
  If you installed TIBCO EMS on a Windows system at the default location, the path to the APIs is C:\tibco\ems\doc\html\api\Java\index.html.

Integrating an Oracle Service Bus Clustered Domain with a Remote TIBCO Enterprise
Message Service 4.2.0/4.3.0 for Messaging Applications in a Service-Oriented Architecture
Updated January 2009

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com