ORACLE
CONSULTING

An Oracle White Paper
October 2011

BI Publisher 11g Scheduling & Apache ActiveMQ as JMS Provider

ORACLE

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Introduction

With the introduction of Oracle Business Intelligence Publisher 11g, the Oracle Business Intelligence Publisher scheduler uses the Java Messaging Service (JMS) queue technology. This allows BI Publisher to dedicate one or more BI Publisher servers exclusively for particular functions such as report generation, document generation or high demand delivery channels (FTP, Fax, WebDAV etc.). The default JMS provider for Oracle Business Intelligence Publisher 11g is Weblogic JMS. Alternatively Apache ActiveMQ can be configured as JMS provider.

This white paper describes the following in detail:

- Adding managed BI Publisher Servers
- Configuring cluster instances
- Procedure to use Apache ActiveMQ as JMS provider
- Failover process

BI Publisher 11g Scheduler Architecture

BI Publisher 11g scheduler is a highly scalable, highly performing and robust report scheduling and delivery system. From submitting the job to delivery of the reports it's a multi step process which also supports failover. The scheduler architecture is depicted through the following diagram.

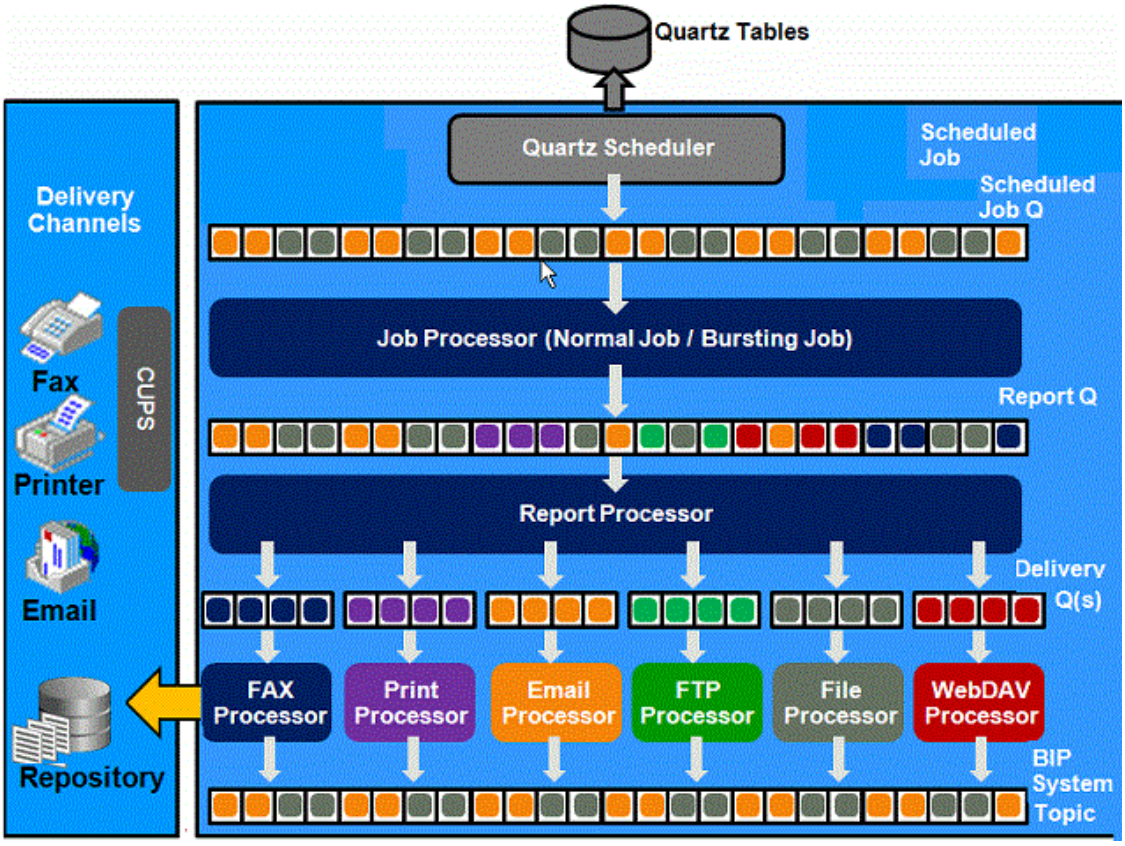

Figure1. Scheduler Architecture

As soon as the user submits a job, the job information is stored in the Quartz tables. Also the necessary triggers are created to run the job, depending on the date, time and frequency at which the job needs to be executed.

A scheduled job is then executed by the Quartz trigger. The job then moves into the scheduler job queue. The job processor then determines whether a particular job needs to be handled by bursting engine or batch job process. This solely depends on whether the job was scheduled for bursting or batch processing. At this stage the job is picked up by the respective engine/process and the business logic is executed. The report metadata is generated and captured into Report Queue.

JMS messages are used for job submission, generation and delivery.

Once the report metadata is captured into Report Queue, the Report Processor comes into action. The job of the Report Processor is to keep listening to the Report Queue and generate the reports based on the metadata available. Once the report is generated it then passes on the delivery related information to Delivery Queue.

The job of the Delivery Processors is to listen to the Delivery Queue and then deliver the report using respective Delivery API(S).

Apache ActiveMQ as JMS Provider

Oracle Business Intelligence Publisher 11g uses Weblogic JMS by default. The scheduler database is installed through Repository Creation Utility (RCU), and the reports are ready for scheduling. Depending on the need, Apache ActiveMQ can be used as JMS provider as well. Oracle recommends to use Apache ActiveMQ 5.2.0 or later. All the examples used in this white paper are for Oracle Business Intelligence Publisher 11g (11.1.1.5) and Apache ActiveMQ 5.5.0.

Download & Install ActiveMQ

Download and install the ActiveMQ software for the respective platform from Apache. Once installed ensure that ActiveMQ is running. The installation and startup steps are available at http://activemq.apache.org/

ActiveMQ queues are accessed using JNDI by default. All the examples in this whitepaper use the default JNDI URL. To alter this default configuration, make modifications to *activemq.xml* found in the <*ActiveMQ_Home*>/conf directory (example: apache-activemq-5.5.0/conf)

If ActiveMQ is chosen as alternative JMS provider, Oracle recommends to use Apache ActiveMQ 5.2.0 or later.

Configure ActiveMQ for BI Publisher

Log into Business Intelligence as administrator (example: weblogic) using <*server_name*>:*port*/xmlpserver (example: http://orabizint:9704/xmlpserver). Click on Administration and Manage BI Publisher. Under System Maintenance click on Scheduler Configuration. The default JMS configuration can be viewed from this page.

Figure2. Default JMS Configuration

To change the JMS provider click on JMS Provider drop down list and choose ActiveMQ. Click on Test JMS button and ensure JMS test completes successfully. Click on Apply button to apply the changes. Log out of Oracle BI and restart.

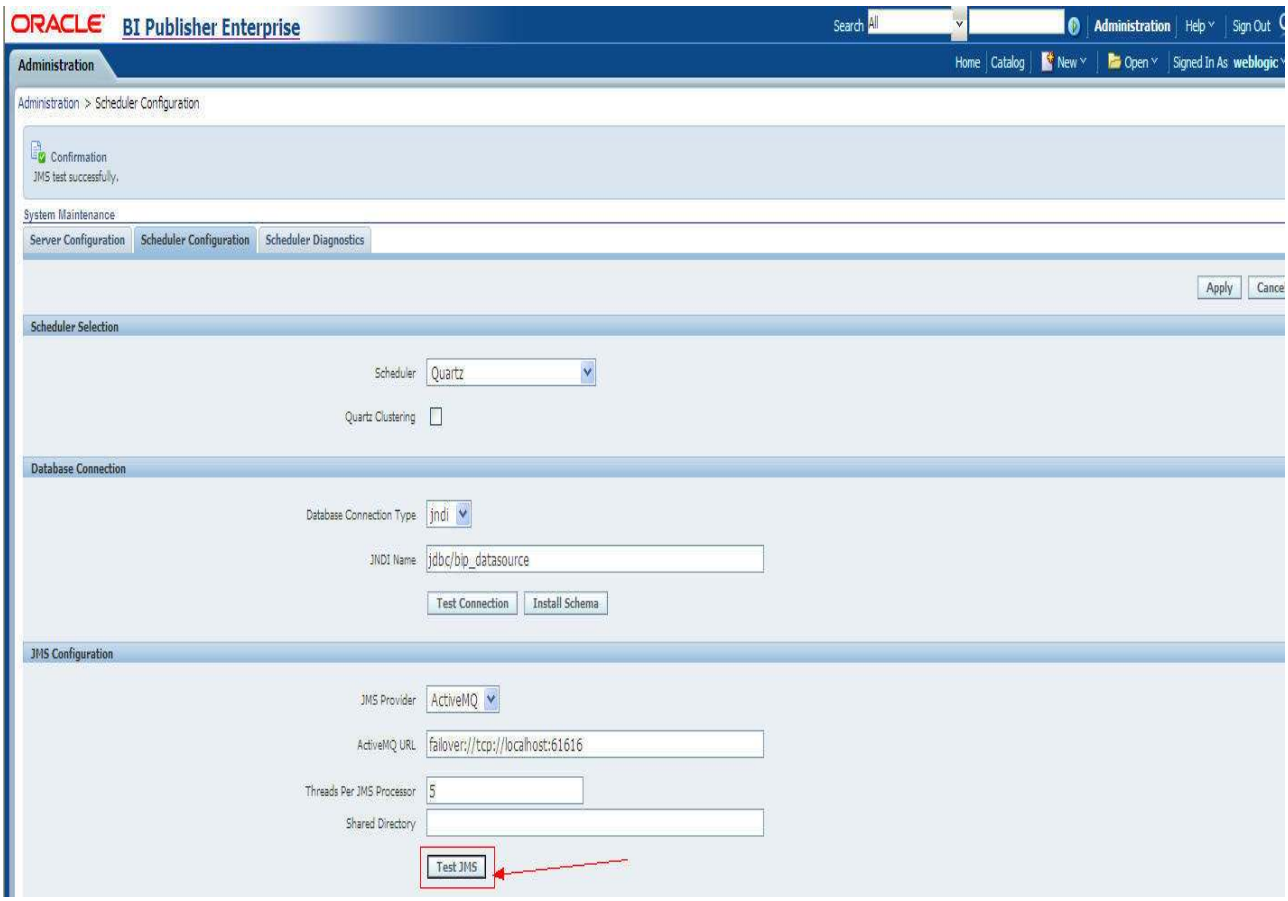

Figure3. Configuring ActiveMQ as JMS

ActiveMQ Console

Start ActiveMQ Console to check whether the configuration done in the BI Publisher is recognized in ActiveMQ. When everything is setup correctly, the ActiveMQ Console lists the BI Server host name and the Broker details here.

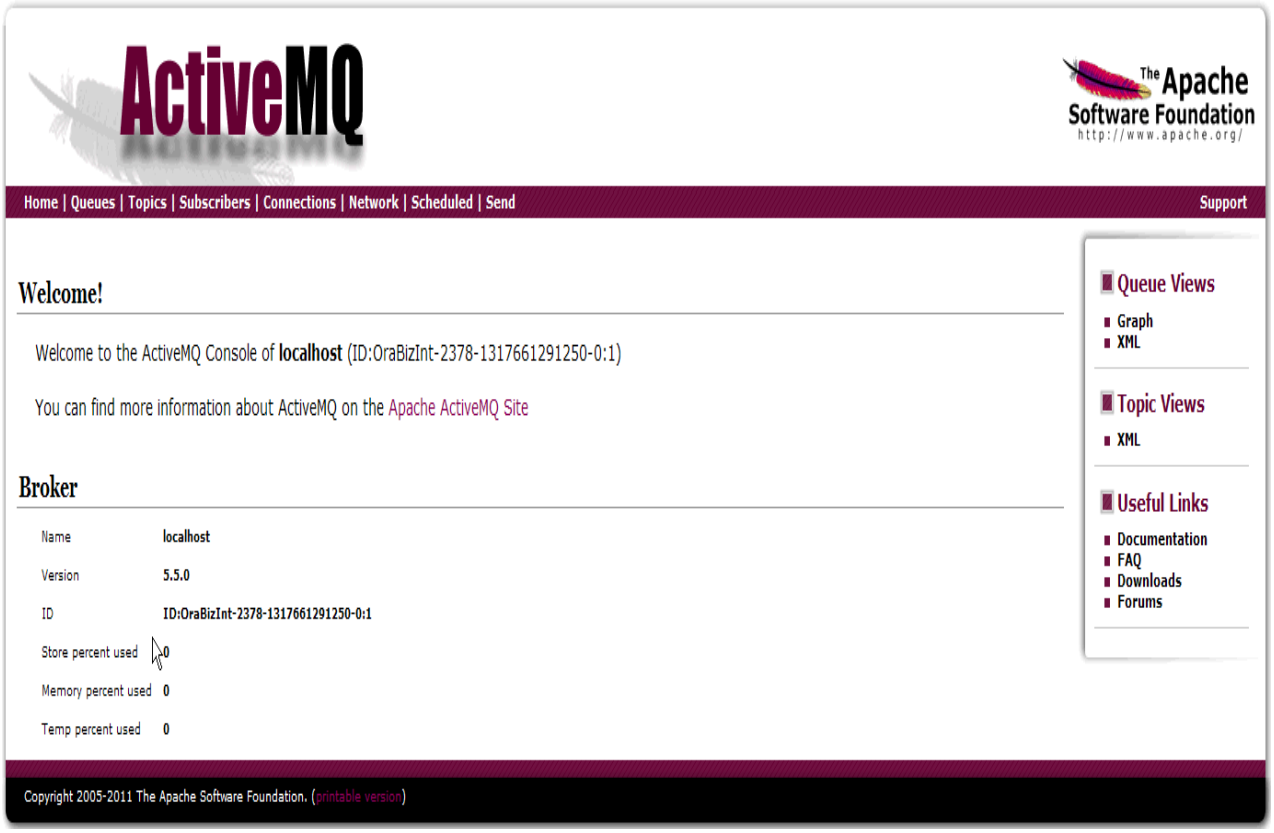

Figure4. ActiveMQ Console

The next step is to check whether all the queues are created properly in the ActiveMQ.

From the ActiveMQ Console home page, click on Queues to view the default queues created for BI Publisher scheduler integration. Notice the queues created for Bursting, Report, Email, Fax etc.

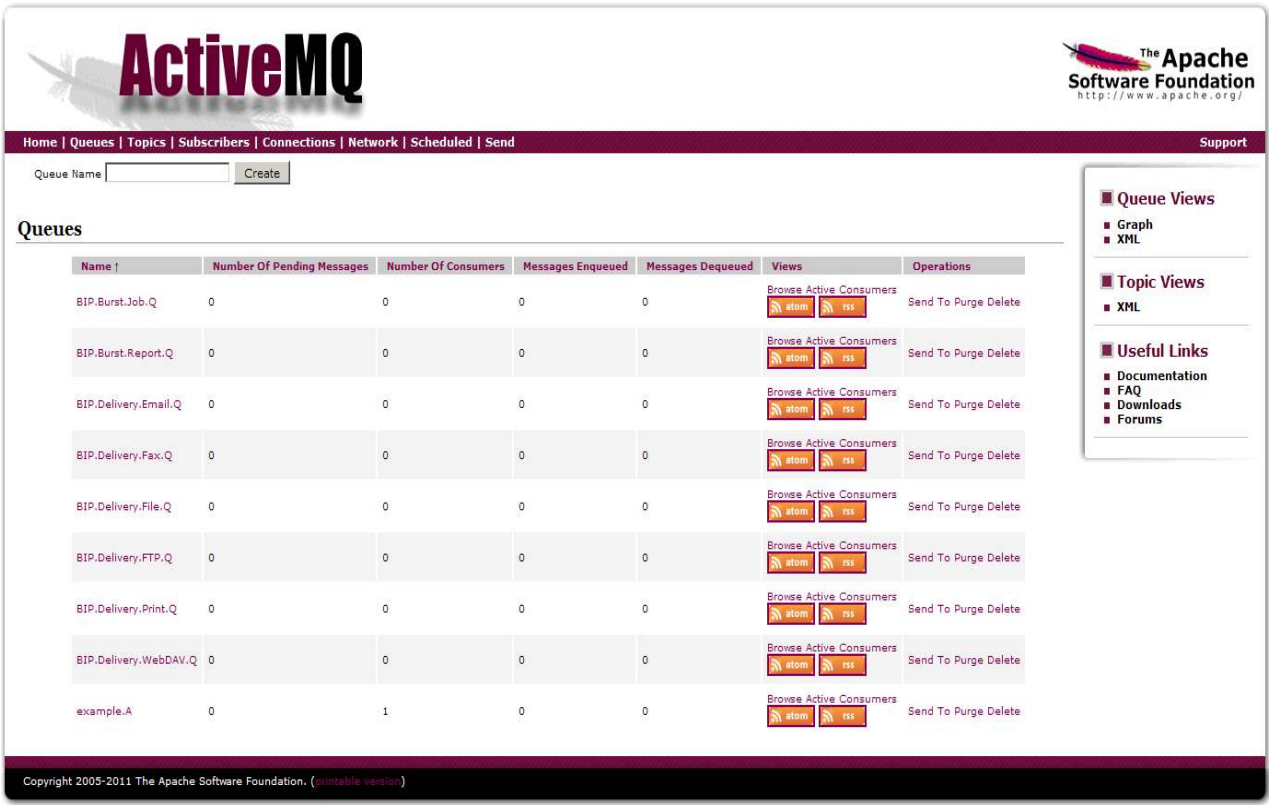

Figure5. ActiveMQ BI Publisher Scheduler Queues

The next section describes the process of adding a managed server to Weblogic.

Add Managed Server

Log into Weblogic Administration Console using <*server_name*>:*port*/console (example: http://orabizint:7001/console) with an admin account (example: weblogic). Under Environment choose Servers. Click on Lock & Edit. Click on New. Enter the details such as Machine Name, Server Name(example: bi_server2), Server Listen Port(example: 9705) etc.

There are 3 options under "Should this server belong to a cluster?" section. To leave the server as a stand-alone server, choose: "No, this is a stand-alone server". To make this server member of an existing cluster choose an existing cluster name by selecting: "Yes, make this server a member of an existing cluster". To create a new cluster and make this server first member of the new cluster choose: "Yes, create a new cluster for this server".

This example uses the existing default cluster "bi_cluster". Click Next button. Review the choices made and click on Finish. The filled in details should look similar to the following screen

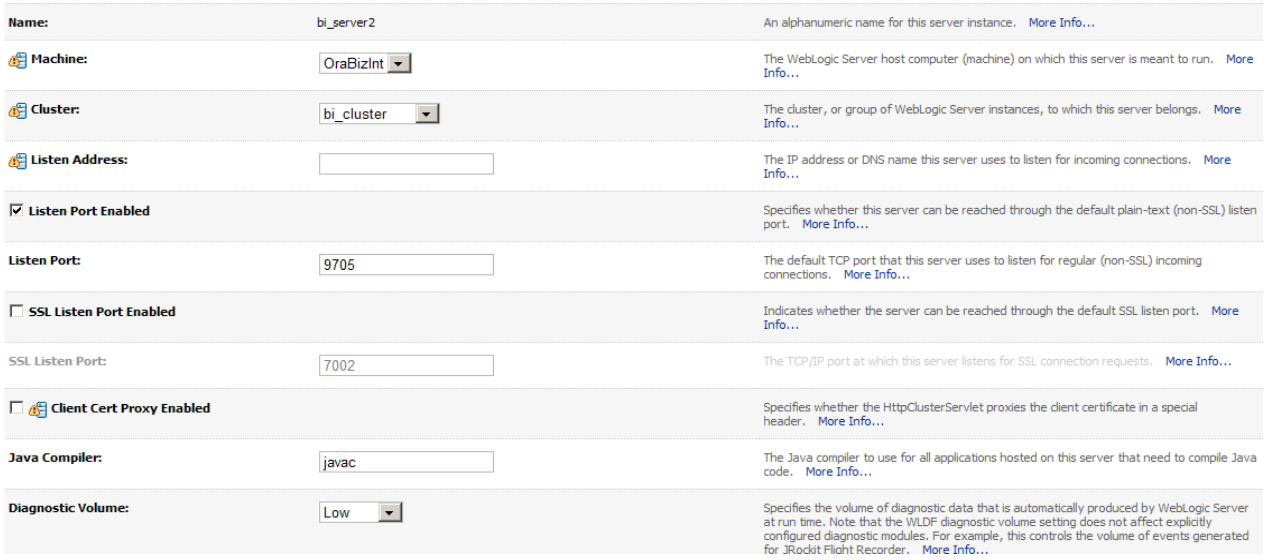

Figure6. Managed Server Details

Start Managed Server

Once the new server is created click on Release Configuration. Click on the Control tab to select the server. Notice that the default state is Shutdown. Click on Start button to start the new server. The state then changes to Starting. Once the server is up and running the state changes to Running.

For further details on how to add a managed server, refer to Adding Managed Server section of Oracle Fusion Middleware Administrator's Guide for Oracle Business Intelligence Publisher.

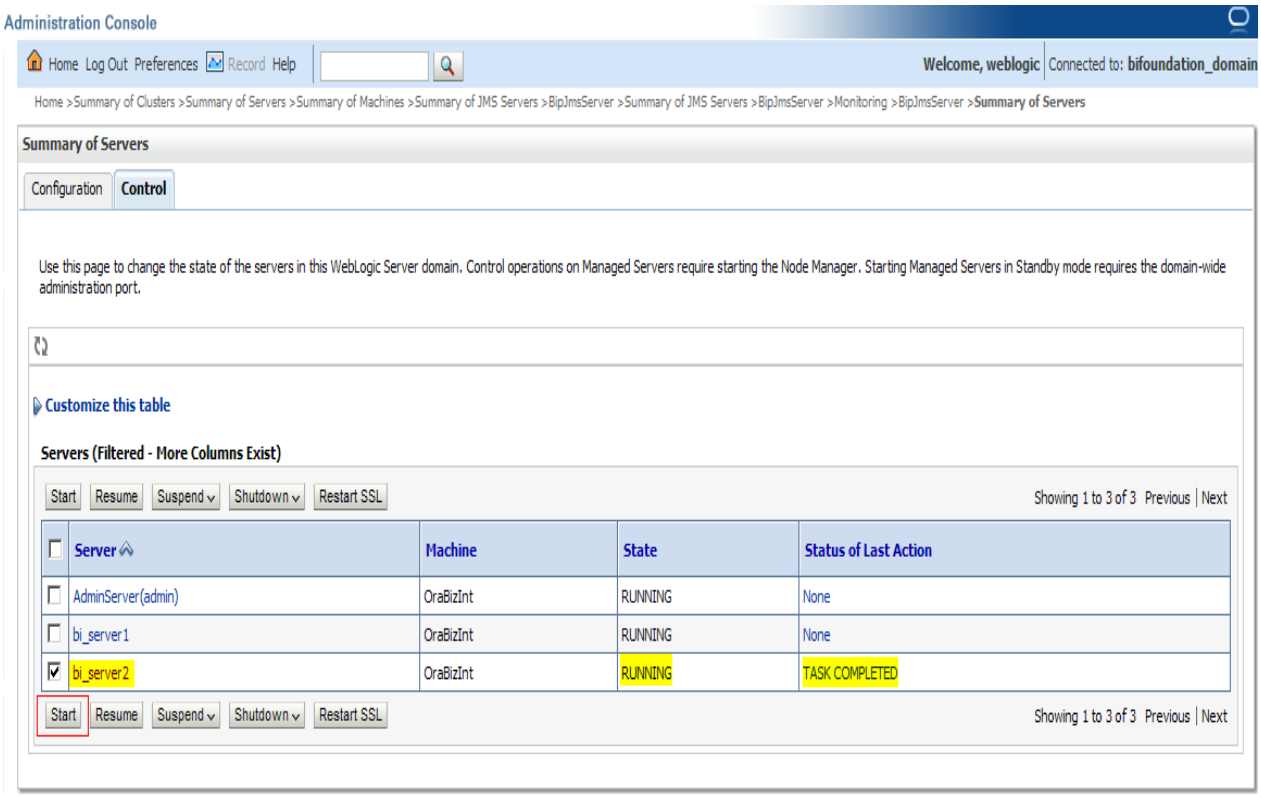

Figure7. Starting Managed Server

Configuring BI Publisher Cluster Instances

Log into Business Intelligence as administrator (example: weblogic). Click on Administration and Manage BI Publisher. Under System Maintenance click on Scheduler Configuration. Notice the second instance added under Cluster Instances. For this example the first instance is used exclusively for JobProcessor and ReportProcessor. The second instance is used exclusively for FTPProcessor. The JobProcessor and ReportProcessor are both allocated with two threads each. The FTPProcessor is allocated with 5 threads.

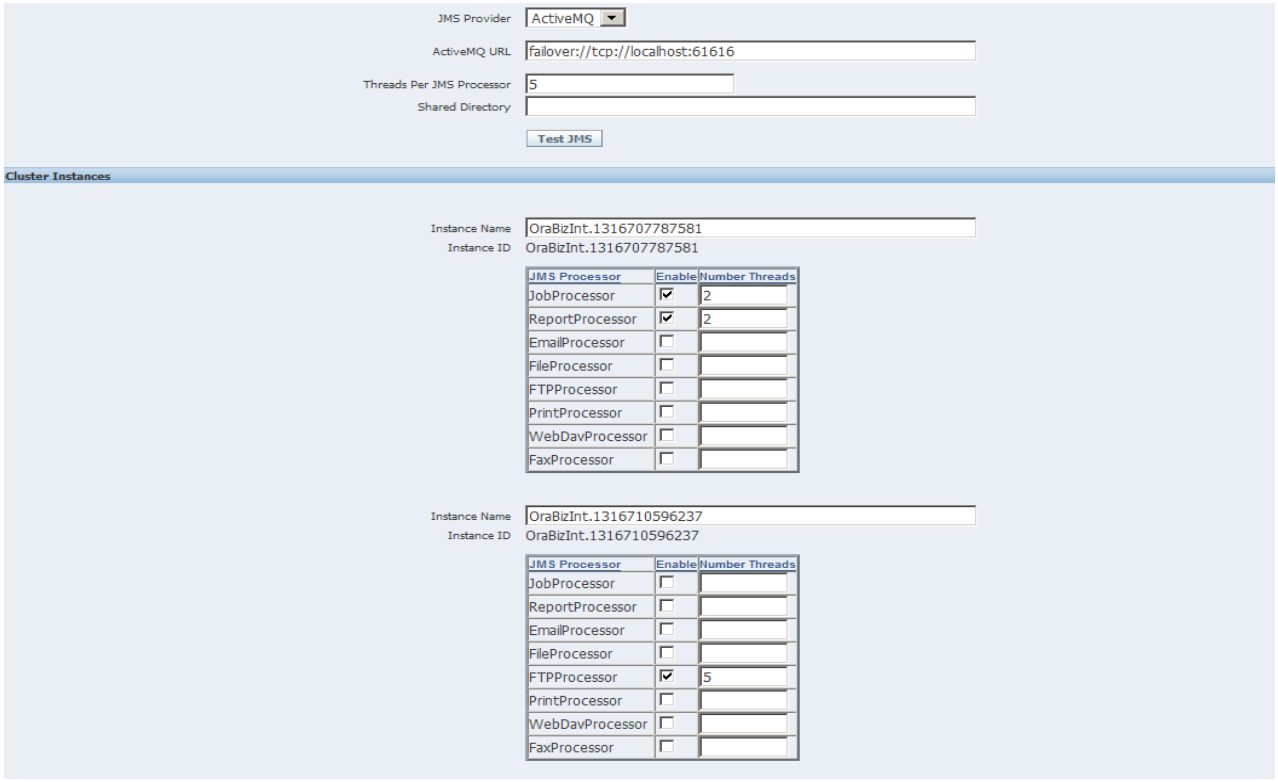

Figure8. Configuring The Cluster Instances

Test Scheduler Configuration

The next step is to test the configuration by running two different jobs one without delivery option and the other with FTP as delivery option. Schedule a report (this example uses Product Listing). Ensure the job is completed successfully. Open ActiveMQ Console and click on Queues tab. Notice that the Job/Report queues have messages Enqueued/Dequeued.

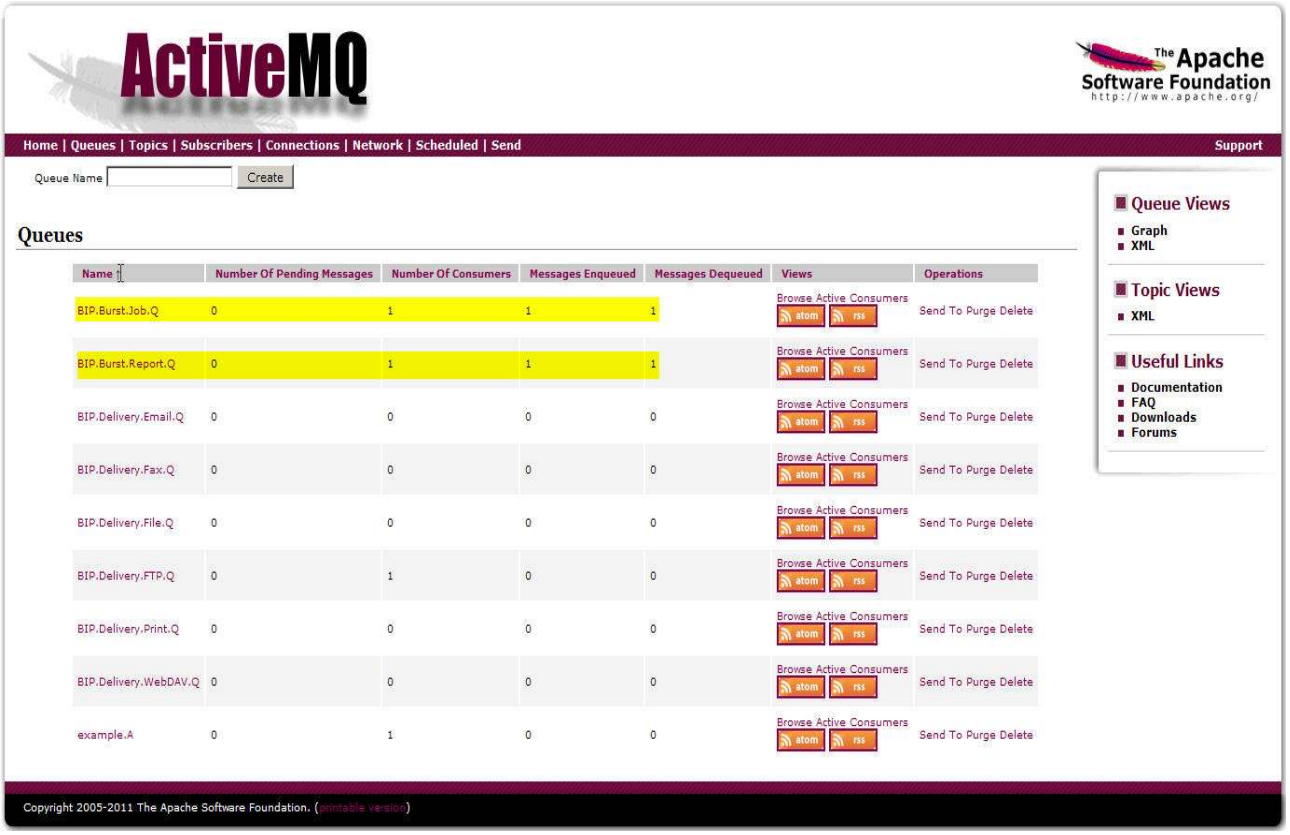

Figure9. ActiveMQ Queues

Click on Connections to verify that two active connections are now available representing the two nodes of bi_cluster. The next step is to schedule the report with FTP as delivery option.

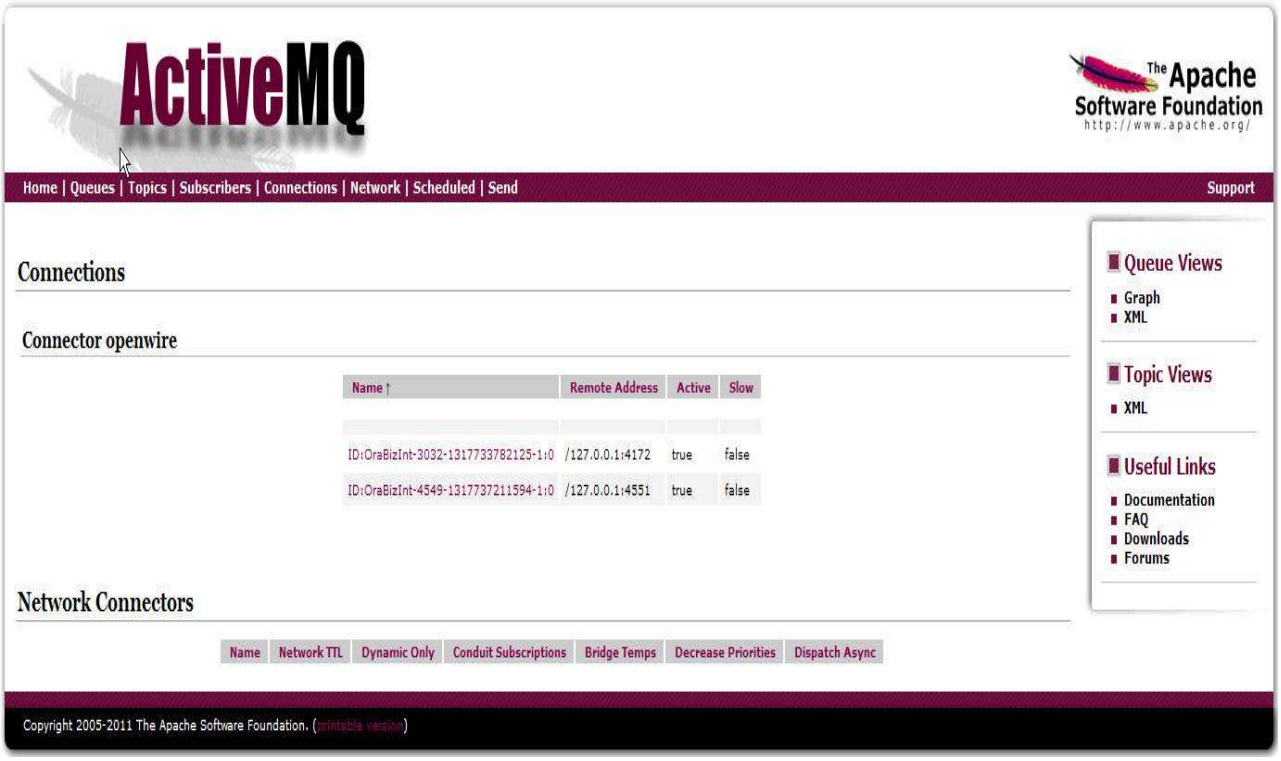

Figure10. ActiveMQ Connections

Schedule the report Product Listing by choosing the delivery option as FTP. Ensure the job is completed successfully. The next step is to check the ActiveMQ Console for the queues. The queues now depict that there are two messages Enqueued/Dequeued for Job/Report queues. One message is Enqueued/Dequeued for FTP.

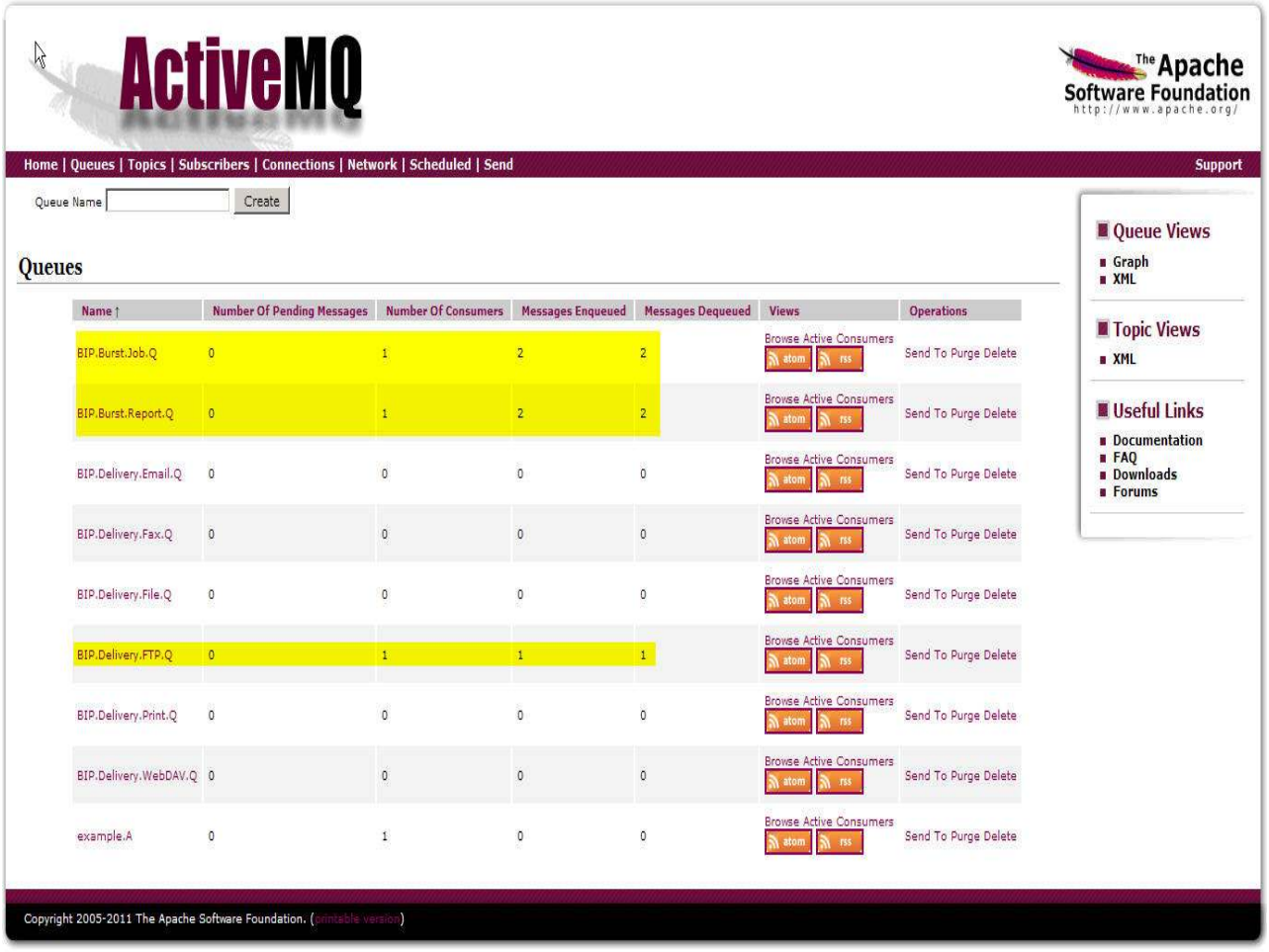

Figure11. ActiveMQ Queues After FTP is Used

Click on Connections and check both the connections to understand that the FTP is handled by the newly added managed server.

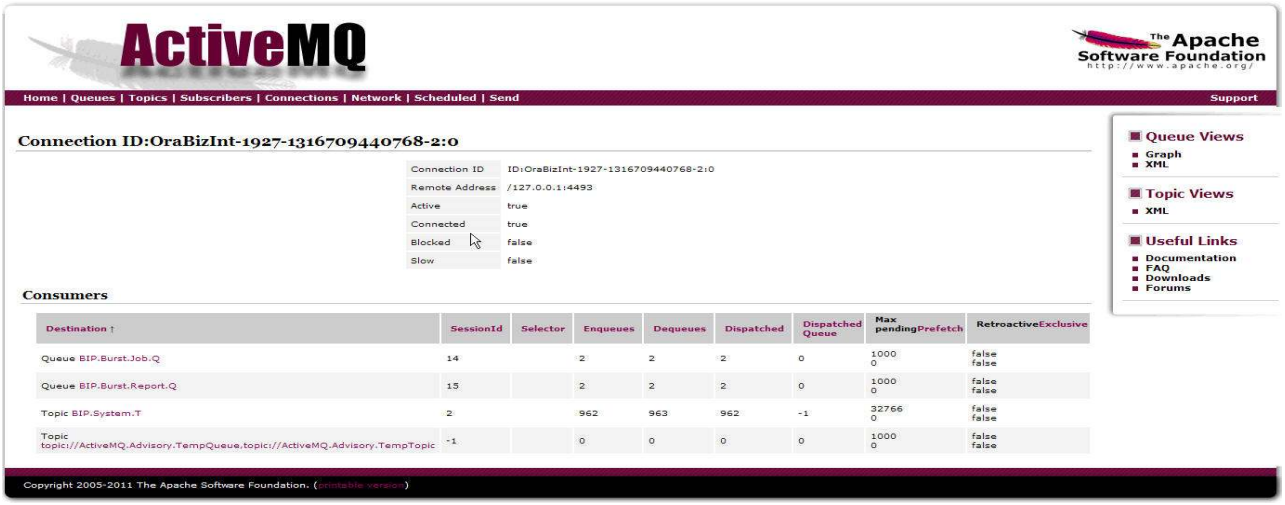

Figure12. ActiveMQ Queues for Report/Job Queues

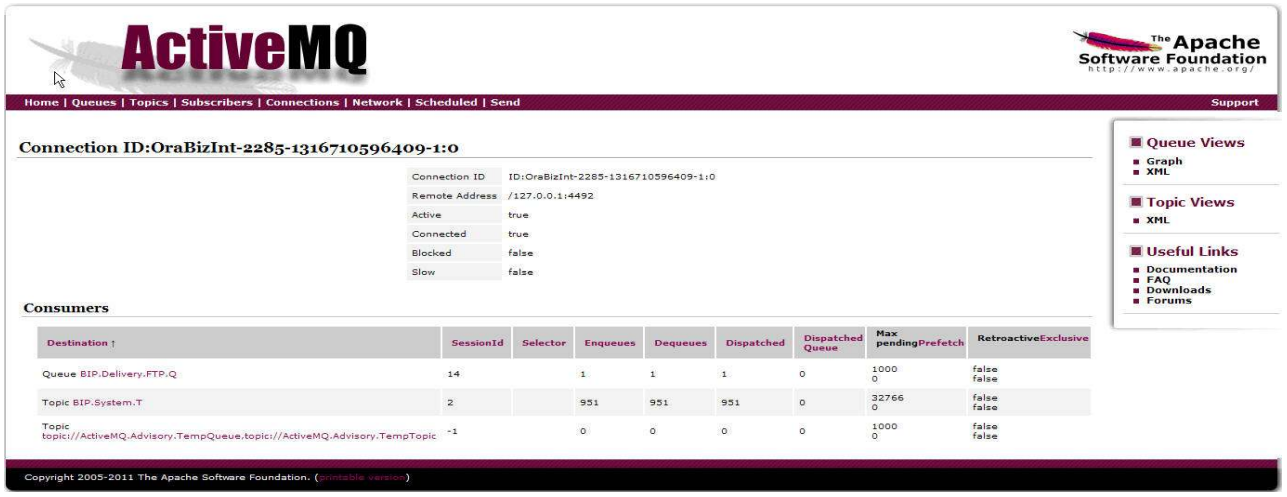

Figure13. ActiveMQ Queues for FTP

Failover Process

The BI Publisher clustering mechanism allows administrators to add more nodes to the required processes on demand. By adding nodes to the cluster, BI Publisher ensures that no report fails to deliver due to server unavailability.

The administrators have to arrive at the number of nodes required for each process for load balancing. The Scheduler Diagnostics page provides good help in arriving at this. In the event of failure of a node in the cluster the other node assigned for the same process will take over. In the above example, if more managed servers are added for FTP, this process will continue even after failure of a particular node.

From the Oracle Business Intelligence home page, click on Administration and Manage BI Publisher. Under System Maintenance click on Scheduler Diagnostics.

--Cluster		Passed
----**Instance - OraBizInt.1316707787581**		Passed
------JMS Instance Config	E:\BI11G\user_projects\domains\bifoundation_domain\servers\bi_server1\tmp_WL_user\bipublisher_11.1.1 \to5gma\war\/WEB-INF/jms_config.xml	Passed
------JMSWrapper	Started (Tue Oct 04 10:06:54 EDT 2011)	Passed
--------JMSClient - system	Started; BIP.System.T: 894 sent, 0 failed	Passed
----------JMSProcessor - ClusterMessageListener	Started; BIP.System.T; 1 threads; 1787 received, 0 failed, 0 running	Passed
--------JMSClient - jmsclient_producer	Started; BIP.Burst.Job.Q: 2 sent, 0 failed; BIP.Burst.Report.Q: 2 sent, 0 failed; BIP.Delivery.FTP.Q: 1 sent, 0 failed	Passed
--------JMSClient - jmsclient_schedule	Started	Passed
----------JMSProcessor - JobProcessor	Started; BIP.Burst.Job.Q; 2 threads; 2 received, 0 failed, 0 running	Passed
----------JMSProcessor - ReportProcessor	Started; BIP.Burst.Report.Q; 2 threads; 2 received, 0 failed, 0 running	Passed
----**Instance - OraBizInt.1316710596237**	Status updated (Tue Oct 04 13:53:48 EDT 2011)	Passed
------JMS Instance Config	E:\BI11G\user_projects\domains\bifoundation_domain\servers\bi_server2\tmp_WL_user\bipublisher_11.1.1 \zh0bkz\war\/WEB-INF/jms_config.xml	Passed
------JMSWrapper	Started (Tue Oct 04 09:48:18 EDT 2011)	Passed
--------JMSClient - system	Started; BIP.System.T: 965 sent, 0 failed	Passed
----------JMSProcessor - ClusterMessageListener	Started; BIP.System.T; 1 threads; 1858 received, 0 failed, 0 running	Passed
--------JMSClient - jmsclient_producer	Started	Passed
--------JMSClient - jmsclient_delivery	Started	Passed
----------JMSProcessor - FTPProcessor	Started; BIP.Delivery.FTP.Q; 5 threads; 1 received, 0 failed, 0 running	Passed
--Database		Passed
----**Database Config**	E:/BI11G/user_projects/domains/bifoundation_domain/config/bipublisher/repository/Admin/Scheduler/database-config.xml	Passed
------Connection Type	jndi	Info
------JNDI Name	jdbc/bip_datasource	Info
----**Toplink Config**	E:/BI11G/user_projects/domains/bifoundation_domain/config/bipublisher/repository/Admin/Scheduler/database-config.xml	Passed

Figure14. Scheduler Diagnostics Page

Conclusion

BI Publisher 11g provides advanced and robust scheduling mechanism. Load balancing and failover processes are supported by adding Weblogic managed servers and managing the clusters. The report repository and the scheduler database are shared across the multiple instances. JMS is used for report job submission, report generation and report delivery to different destinations. In addition to the default Weblogic JMS provider, BI Publisher 11g also supports Apache ActiveMQ.

Oracle consulting has had many experiences in implementing customized scheduling needs for various customers. If you're interested in Oracle consulting to discuss more in detail about the implementation and review of your reporting or scheduling needs, please contact Shankar Duvvuri (shankar.duvvuri@oracle.com), Senior Principal Consultant, Oracle ATS BI Delivery & BI Advanced Reporting group.

BI Publisher 11g Scheduling & Apache ActiveMQ as JMS Provider
October 2011
Author: Shankar Duvvuri

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

Oracle is committed to developing practices and products that help protect the environment

SOFTWARE. HARDWARE. COMPLETE.