Oracle SOA Suite High-Availability FAQ and Troubleshooting Guide

An Oracle White Paper
March 2010
Contents

1 Preface .......................................................................................................................... 1
2 General .......................................................................................................................... 2
   2.1 What are the Basic Steps for a Clustered Installation of BPEL, ESB and Web Services Manager Components? ................................................................. 2
   2.2 Can I Use the Application Server "J2EE, Web Server and SOA Suite" Advanced Installation Type for a Clustered Setup? ................................................................. 2
   2.3 Do I Need to Install Oracle HTTP Servers in Separate Oracle Homes? .................. 2
   2.4 Can BPEL, OWSM and ESB-Runtime Components be Deployed to the Same OC4J Container? .......................................................... 3
   2.5 What is an ONS Topology? .................................................................................... 3
   2.6 What are the Differences between Application Server Clustering and BPEL or ESB Clustering? .................................................. 3
   2.7 Is Time Synchronization Important between Nodes in a SOA Cluster? ................. 3
   2.8 What Load Balancer Configurations are Required? ........................................... 3
3 Oracle BPEL Process Manager .................................................................................... 4
   3.1 Does BPEL Clustering Require an Application Server Cluster? ......................... 4
   3.2 Are there Other Advantages to Running an Application Server Cluster with a BPEL Cluster? ............................................................ 4
   3.3 Why is it Necessary to Configure jGroups for BPEL Clustering Since the BPEL Engine is Stateless? .......................................................... 4
   3.4 How is a BPEL Cluster Defined? ........................................................................ 4
   3.5 What Happens When a New BPEL Process is Deployed While Another Node is Down? ............................................................... 5
   3.6 Can I Cluster a BPEL Process by Deploying it into Two Domains of the Same BPEL Server? ............................................................... 5
   3.7 What Artifacts are Included in the Dehydration Store Database? ....................... 5
   3.8 Can I Add New Nodes to the BPEL Cluster without Shutting Down the Dehydration Database or Existing BPEL Nodes? .............................................. 5
   3.9 When a BPEL Node Goes Down, How do Other Nodes Get the Instance and Continue Execution? ...................................................... 6
   3.11 Does TCP have Advantages Over UDP as the jGroups Protocol? ..................... 6
   3.12 How is BPEL Configured to Work with a Real Application Cluster (RAC) Database? ............................................................... 7
   3.13 Are Modifications to the Process Deployment Descriptor Propagated to all Nodes of the BPEL Cluster? ...................................................... 7
   3.14 Are Modifications to collaxa-config.xml or the BPELAdmin Console Propagated to all Nodes in the BPEL Cluster? .............................. 7
3.15 Replacement of JGroups for BPEL Clustering and BPEL Singleton Adapters Communication in 10.1.3.5.1........................................................................................................... 7

4 Enterprise Service Bus........................................................................................................ 9

4.1 What is the ESB Multi-Tier Deployment Architecture?.................................................. 9
4.2 What are the Characteristics of an ESB Cluster?.......................................................... 9
4.3 The ESB Standalone Installer Prompts for 3 Different Installation Options. Which Option
    Should I Choose?............................................................................................................ 9
4.4 Why Does ESB Clustering Require AQ JMS Messaging?............................................ 10
4.5 Why is it Necessary to Add "service-failover="1"" and Remove the "numprocs="1"" in
    opmn.xml to Configure the ESBDT Cluster?............................................................... 10
4.6 Within the orion-application.xml File, What Does the primary_oc4j Setting Indicate?...... 10
4.7 Where do I Find the create_esb_topics.sql File? How do I Verify if the ESB Topics Have been
    Created?......................................................................................................................... 11
4.8 What is the Advantage to Deploying a Load-Balancer in Front of the ESBDT Nodes?..... 11
4.9 How Can I Manually Configure the ESBRT Nodes to Point to a New ESBDT?.......... 11
4.10 What is the Runtime Interaction between ESBRT and ESBDT?................................. 11
4.11 Is it Necessary to Install ESBRT and BPEL in the Same Oracle Home?..................... 12
4.12 Is it Necessary to Install ESBRT and BPEL in the Same OC4J Container?................... 12
4.13 What are the Options for Updating ESB Metadata, i.e. the oraesb.esb_parameter Table? 12
4.14 What is the Relationship between the ESB Configuration Files esb_config.ini and orion-
    application.xml?......................................................................................................... 13
4.15 Where Should I Specify the cluster_name Parameter for ESBRT Instances? Does ESBDT
    Require a cluster_name Setting?................................................................................... 13
4.16 What are the Correct ESB Console Settings for an ESB System in a Cluster?............ 13
4.17 What is the Relationship Between ESB Systems and ESB Runtime Clusters?............. 14
4.18 What are the Advantages to Using Multiple ESB Systems in a Clustered Runtime
    Installation?.................................................................................................................. 14
4.19 How do you Apply a Patch When ESBDT and ESBRT are in Separate Containers?..... 14
4.20 Are There Other Recommendations for ESBRT Active-Passive Configuration?........... 15
4.21 Why Doesn't ESBDT Support an Active-Active Topology?......................................... 15

5 Adapters............................................................................................................................ 16

5.1 Which Technology Adapters Use Active-Passive Clustering?...................................... 16
5.2 How Do I Setup a Singleton Adapter for BPEL?......................................................... 16
5.3 How Do I Setup a Singleton Adapter for ESB?......................................................... 17
5.4 How Does the Adapter Cluster Work?.......................................................................... 17
5.5 What if ESB and BPEL are Not Installed Together? Does that Effect my Adapter Cluster
    Configuration?.............................................................................................................. 18
5.6 Can I Have More Than One Primary Activation Agent to Process High Volume Requests? 18
5.7 What is the Relationship between the Adapter Cluster and the BPEL or ESB Cluster?.... 18
5.8 Should I Use TCP or UDP Communication in an Adapter Cluster?.............................. 18
5.9 What Loggers are Useful for Adapters?......................................................................... 18
5.10 How Can I Improve Throughput with Active-Passive Adapters in BPEL?................... 19
5.11 How Do I Check if an Activation Agent is Setup Correctly for BPEL?....................... 19
5.12 How Do I Monitor Activation Agent Failover in BPEL?............................................ 20
5.13 How Do I Monitor Adapter Activation in ESB?......................................................... 20
5.14 What Additional Configuration is Necessary for File and FTP Adapters in a Cluster?.... 22
5.15 How Do I Enable Distributed Polling for an Inbound Database Adapter?................... 22
5.16 Can MQ and JMS Adapters Operate in an Active-Active Cluster?.............................. 23

6 Troubleshooting................................................................................................................. 24
6.1 How Can I Verify my ESB Cluster is Setup Correctly? ................................................................. 24
6.2 Invoking ESB Services Registered In ESB Cluster Fails With Error "Cannot find service on current cluster" ............................................................................................................. 24
6.3 Error in log.xml File During ESB Startup: "Invalid Destination Topics/ESB_MONITOR" and "No resource named 'ojmsRP/Topics/ESB_MONITOR' found" ......................................................... 24
6.4 After Cluster Node Fails, ESB Returns Connection Refused Error .................................................. 24
6.5 When Deploying a BPEL Process to a Node in the Cluster, it Does Not Propagate Across the Cluster ........................................................................................................................................... 25
6.6 Verify jGroups Initializes Correctly .................................................................................................. 25
6.7 Verify jGroups is Binding to the Correct Network Interface for UDP or TCP .............................. 27
6.8 Test Connectivity between BPEL Nodes ......................................................................................... 27
6.9 Verify UDP jGroups Settings for mcast_addr and mcast_port ..................................................... 27
6.10 Verify TCP jGroups Settings for start_port, initial_hosts and port_range ............................ 28
6.11 jGroup Nodes May Drop Out of the BPELCluster ..................................................................... 28
6.12 Enable Detailed jGroups Logging .............................................................................................. 28
6.13 A New BPEL Domain is Created When a Cluster Node is Down, That Node Fails to Pickup New Domain on Restart ......................................................................................... 29
6.14 BPEL Processes Missing from Console When BPEL Cluster Uses a BigIP Loadbalancer with Persistence Set to On ........................................................................................................ 29

7 References ......................................................................................................................................... 30
1 Preface

This document is a supplement to the complete documentation for Oracle SOA Suite high-availability; please review the list in the references section. Specifically, the reader should be familiar with the latest Oracle Application Server Enterprise Deployment Guide.

The current scope of this document is focused on clustering of Oracle Application Server as it relates to SOA Suite components, BPEL Process Manager (BPEL), Enterprise Service Bus (ESB) and Adapters. For topics not addressed here, please see to the reference section for pointers to the relevant documentation.
2 General

2.1 What are the Basic Steps for a Clustered Installation of BPEL, ESB and Web Services Manager Components?

The BPEL, ESB and Web Services Manager components have different clustering characteristics and each requires specific installation and configuration steps to setup their HA cluster. The Enterprise Deployment Guide 10.1.3.3 describes these steps in great detail, see figure 1-1 for a representative architecture diagram.

The basic installation steps for each node in a 10.1.3.3 cluster are as follows:

1. Install Application Server 10.1.3.1
2. [Optional] Install Oracle HTTP Server in separate Oracle Home
3. Install BPEL Process Manager 10.1.3.1
4. Install Enterprise Service Bus runtime 10.1.3.1
5. Install Web Services Manager 10.1.3.1
6. Install Application Server 10.1.3.3 / 10.1.3.4 patchset
7. Install Enterprise Service Bus 10.1.3.3 / 10.1.3.4 designtime
8. [Optional] Apply the latest 10.1.3.3 / 10.1.3.4 MLR patchset

Note the Enterprise Deployment Guide 10.1.3.3 is currently the latest version. The steps are identical for a new install of 10.1.3.4.

2.2 Can I Use the Application Server "J2EE, Web Server and SOA Suite" Advanced Installation Type for a Clustered Setup?

No, the "J2EE, Web Server and SOA Suite" installation type should never be chosen for SOA Suite clustered installations. Instead it's important to start with the advanced installation option of "J2EE and Web Server" or "J2EE" for each SOA Suite node in the cluster, then separately install BPEL, ESB and Web Services Manager into the appropriate application server Oracle Homes. See the Enterprise Deployment Guide for details.

2.3 Do I Need to Install Oracle HTTP Servers in Separate Oracle Homes?

If you intend to deploy Oracle HTTP Server (OHS) outside a firewall, then you must install another Oracle Home with OHS in that network zone. If not, it's generally easier to install OHS and the SOA Suite components into the same Oracle Homes. For that scenario, you can start with the application server advanced installation for "J2EE and Web Server", and then install the SOA components into the same Oracle Home.
2.4 Can BPEL, OWSM and ESB-Runtime Components be Deployed to the Same OC4J Container?

Yes, although your performance goals might require separating these components into different OC4J containers to expand available memory resources but they can be deployed together.

Only the ESB design-time component must be deployed into a separate OC4J container to operate in an active-passive failover configuration.

2.5 What is an ONS Topology?

An ONS topology or OPMN/ONS Cluster is a group of Oracle Application Server instances that communicate to each other through ONS. ONS topologies share availability notifications across the participants in the topology, this enables dynamic routing from OHS to OC4J (i.e. routing behavior that changes dynamically without configuration changes) and dynamic discovery of application server and OHS instances.

2.6 What are the Differences between Application Server Clustering and BPEL or ESB Clustering?

Application server clustering is independent of BPEL clustering but in practice both are almost always used together. Setting up an application server cluster however does not enable nodes to share BPEL specific data.

Similarly, application server clustering is independent of ESB runtime clustering. ESB runtime (ESBRT) nodes and ESB design-time (ESBDT) nodes are clustered by configuring each ESBRT / ESBDT to share the same Metadata repository database. This establishes the common persistence layer for all ESB servers to share the same AQ messaging system used for group communication.

Application server clustering is used for active-passive failover of the ESBDT servers. The use of OC4J groups and Application Server clustering allows OPMN to ensure only one ESBDT instance is active at any time and also controls the startup of the standby ESBDT when the active ESBDT goes down.

2.7 Is Time Synchronization Important between Nodes in a SOA Cluster?

Yes, each node in the SOA cluster must have accurate time synchronization. BPEL process activities can have time-sensitive behavior, such as wait activities, and time-dependent process scheduling. Two nodes without time synchronization may not properly execute BPEL processes.

2.8 What Load Balancer Configurations are Required?

The Enterprise Deployment Guide 10.1.3.3 covers load balancer configurations in sections 2.4, 3.1, 3.9, 3.10, 3.12. 4.1, 4.3.1, and 4.3.3.
3 Oracle BPEL Process Manager

3.1 Does BPEL Clustering Require an Application Server Cluster?

BPEL clustering operates whether or not it runs on top of an application server cluster. The BPEL cluster nodes communicate via jGroups technology and share the same dehydration store.

Note is BPEL 10.1.3.5.1, jGroups was replaced with an internal database-based mechanism to provide cluster communication between BPEL nodes. See the section “Replacement of JGroups for BPEL Clustering…” in this paper for more details.

In practice, an application server is almost always used since this handles OHS routing requests to the available BPEL instances. Without an application server cluster, manual routing rules must be configured from OHS to the OC4J instances running BPEL.

Thus with application server clustering configured, when an OC4J container running BPEL goes down, SOAP/HTTP requests coming into an OHS instance in the cluster will be automatically routed to another available BPEL node.

3.2 Are there Other Advantages to Running an Application Server Cluster with a BPEL Cluster?

When using the BPEL Java API to invoke BPEL processes, you can make use of the oracle.j2ee.rmi.loadBalance JNDI property and the OPMN provider URL to achieve load balancing.

3.3 Why is it Necessary to Configure jGroups for BPEL Clustering Since the BPEL Engine is Stateless?

The BPEL jGroups cluster allows one node to notify other cluster nodes of specific process and domain changes:

- Process deployment or undeployment
- Creation or deletion of BPEL domains
- BPEL domain property settings
- bpel.xml property changes for a BPEL project

These process and domain notifications are passed using jGroups communication between nodes.

Note in BPEL 10.1.3.5.1, jGroups was replaced with an internal database-based mechanism to provide cluster communication between BPEL nodes. See the section “Replacement of JGroups for BPEL Clustering…” in this paper for more details.

3.4 How is a BPEL Cluster Defined?

BPEL clustering is achieved with the appropriate configuration in the jgroups-protocol.xml and collaxa-config.xml files for each node, and by pointing all nodes to a common dehydration store database. These clustering configuration steps are covered in the Enterprise Deployment Guide.
Note in BPEL 10.1.3.5.1, jGroups was replaced with an internal database-based mechanism to provide cluster communication between BPEL nodes. See the section “Replacement of JGroups for BPEL Clustering…” in this paper for more details.

3.5 What Happens When a New BPEL Process is Deployed While Another Node is Down?
When the down node is started, it will synchronize with the dehydration store database and deploy the new process onto itself.

3.6 Can I Cluster a BPEL Process by Deploying it into Two Domains of the Same BPEL Server?
No. Domains represent a grouping of BPEL processes; creation of a new domain does not add another physical node to a cluster for high-availability.

In addition, the full path of a BPEL process includes the domain name, process name, and version extension. Processes are considered different with variation on any one of these three elements and therefore cannot be clustered across different domains.

3.7 What Artifacts are Included in the Dehydration Store Database?
The following artifacts are stored in the BPEL dehydration database:

- BPEL deployment suitcases, including bpel.xml
- Domain definitions (domains table)
- BPEL instances meta data (cube_instance table)
- BPEL instances audit trail (audit_trail table)
- Asynchronous invoke messages (invoke_message table)

3.8 Can I Add New Nodes to the BPEL Cluster without Shutting Down the Dehydration Database or Existing BPEL Nodes?
In general, yes. You need to configure the new node to point to the existing dehydration database and properly configure the jgroups-protocol.xml and collaxa-config.xml files.

jGroups supports two different network protocols for cluster communication, TCP and UDP. If TCP is used, then all existing nodes must have the list of nodes within jgroups-protocol.xml file updated to include the new node. In this case, the existing BPEL nodes must be restarted to reload the jGroups configuration. Since UDP uses multicast, no such restart is necessary.

In BPEL 10.1.2, you must copy the BPEL suitcase file (the jar file under orabpel/domains/domain_name/deploy) from the existing nodes to the new node. In all versions, you must ensure that the process design does not include hard coded values such as host name that are only suitable for a specific node.
3.9 When a BPEL Node Goes Down, How do Other Nodes Get the Instance and Continue Execution?

If the BPEL process instance is waiting at a mid-process receive activity when the BPEL node goes down, the process instance is not in an active JTA transaction. The process instance remains at the dehydrated state until the message for which it is waiting arrives. Upon arrival of the message, an available BPEL node retrieves the instance from the dehydration store and continues processing.

If the BPEL process is in the middle of an active JTA transaction when the server goes down, the transaction is rolled back. If this is an asynchronous invocation, the instance is listed in the manual recovery page of the BPEL console. From this page an administrator can resubmit the process and an available BPEL server in the cluster can continue execution. Recovery begins from the latest dehydration point or, if no dehydration point was reached before failure, the process starts at the beginning. The recovery process can be automated through custom programs utilizing the BPEL API.

If this is a synchronous invocation, the client receives an error and is responsible for resubmitting the message.

If the process instance is waiting at the wait activity, then you must update the timer table of one of the nodes so that the wait activity awakens on that node. Otherwise, even after the timer expires, none of the nodes will complete the wait activity until a manual recovery is performed.

3.10 How Does an Administrator Perform Manual Recovery of a BPEL Process?

Go to the BPEL Console, navigate to the Process tab and then click the Perform Manual Recovery link in the lower left. The pending instances on this page are categorized by invoke messages, callback, and activity.

Exercise caution when performing manual recovery. The BPEL console recovery page shows all messages for which the corresponding instance is not completed. This includes instances that are still in-flight and have not yet completed a transaction. Use some criteria to recover the right messages (e.g. look at the message age or payload via the BPEL Java API), to understand if the instance has truly rolled back and is in need of recovery.

3.11 Does TCP have Advantages Over UDP as the jGroups Protocol?

UDP multicast does not work when clustered nodes are located on different network subnets (although special network configuration can make this possible).

Adapter clustering also uses jGroups to enable active-passive (singleton) adapter endpoints. If there are more than 2 (two) singleton adapters defined in an adapter cluster, it is recommended to use UDP instead of TCP. See the Adapters section in this paper for more information on adapter clustering.
3.12 How is BPEL Configured to Work with a Real Application Cluster (RAC) Database?

This configuration is covered in the Enterprise Deployment Guide 10.1.3.3 section 3.29, and the Oracle SOA Suite XA and RAC Guide.

3.13 Are Modifications to the Process Deployment Descriptor Propagated to all Nodes of the BPEL Cluster?

Yes. Changes to the bpel.xml deployment descriptor file, such as number of activation agents, is synchronized across all nodes in BPEL cluster automatically.

3.14 Are Modifications to collaxa-config.xml or the BPELAdmin Console Propagated to all Nodes in the BPEL Cluster?

No. Changes to the collaxa-config.xml or BPELAdmin console properties must be applied to each node manually.

3.15 Replacement of JGroups for BPEL Clustering and BPEL Singleton Adapters Communication in 10.1.3.5.1

Starting in version 10.1.3.5.1 BPEL clustering and BPEL singleton adapters no longer use jGroups for cluster communication and singleton adapter coordination between the nodes of a BPEL cluster.

Note that ESB singleton adapters still use jGroups; this remains unchanged from previous versions.

To define version 10.1.3.5.1: it is the application of 10.1.3 Oracle Application Server patch set 10.1.3.5.0 (patch ID 8626084) and Integration patch MLR #1 (patch ID 9034573) or it is the 10.1.3.5.1 WebLogic SOA installation. This change is realized by the fix to bug 8608385 being included into 10.1.3.5.1.

Implementation of this feature does not require any configuration on the user’s part. Once a system has version 10.1.3.5.1 applied the functionality is fully implemented.

jGroups is replaced as the means of propagating cluster messages across BPEL nodes with a database-based solution where cluster messages are inserted into a database table (a BPEL cluster is by definition nodes that share the same dehydration store database). Each BPEL node has a specific thread polling the database table for new cluster messages. The same database-based infrastructure is also used by cluster nodes to nominate a master node for the cluster. The master node will be the node where adapter active-passive (singleton) activations will be enabled.

Three new tables are introduced into the orabpel schema to provide this functionality:

CLUSTER_MASTER – Contains the NODE_ID of the BPEL cluster node that has the active polling endpoint for inbound singleton adapters. Adapter clusters are still defined as before using a clusterGroupIId for the activation agent of the adapter in bpel.xml. The master node will change if the current master node is taken out of the cluster and will be replaced by one of the other active cluster nodes.

CLUSTER_MESSAGE – Keeps track of changes made to the BPEL cluster at the BPEL system and domain level that need to be distributed to all cluster nodes via the posting
of the cluster messages. Columns include DOMAIN_ID, NODE_ID, MSG_TYPE, MSG_TEXT, and MSG_DATE.

CLUSTER_NODE – Maintains a list of active nodes in the cluster by NODE_ID and IP_ADDRESS. Nodes are added and removed from this table as nodes become active or inactive. One of the NODE_IDs in this table will also be designated as the master node in the CLUSTER_MASTER table.

Cluster messages added to the CLUSTER_MESSAGE table include the following:

- BPEL domain creation/deletion from /BPELAdmin
- BPEL system logger level changes from /BPELAdmin
- Process deployment/undeployment via ANT, JDeveloper, or /BPELConsole
- Process state on/off and process Lifecycle Active/Retired from /BPELConsole
- Clearing of WSDL cache from /BPELConsole
- Process Descriptor changes from /BPELConsole
- BPEL domain logger level changes from /BPELConsole
- BPEL domain configuration (domain.xml) changes from /BPELConsole

Rows from the CLUSTER_MESSAGE table are removed automatically at regular intervals once their distribution requirements have been satisfied.

Formerly when debugging a JGroups problem you would set logger “org.collaxa.thirdparty.jgroups” to Debug level in the BPELAdmin console. For this new database based solution set the logger “collaxa.cube.cluster” to Debug in BPELAdmin. These messages log into the BPEL container logfile, e.g. $ORACLE_HOME/opmn/logs/default_group-oc4j_soa-default_group-1.log.
4 Enterprise Service Bus

4.1 What is the ESB Multi-Tier Deployment Architecture?

Oracle ESB has a multi-tiered architecture which makes it flexible for managing service metadata separate from runtime instances. The ESB designtime server (ESBDT) provides interfaces for all metadata changes from JDeveloper, ant-based import/export or browser-based ESB Control changes. The ESBDT communicates with the ESB runtime server (ESBRT) using JMS and provides to the ESB management console for instance tracking, error management and failed message resubmission.

The ESBRT loads an in-memory cache at startup that contains all service metadata and artifacts such that services run straight from memory. Subsequently all interactions with the Metadata server goes through JMS, which offloads instance tracking and error handling from the runtime server. Additionally, the runtime server is highly available and scalable and guarantees message delivery across a distributed cluster topology.

In addition, there is a data-tier that consists of a database and WebDav server. The database stores the ESB metadata and hosts the AQ messaging system used in a clustered configuration. The WebDav server publishes the ESB service related artifacts such as WSDLs and XSDs.

4.2 What are the Characteristics of an ESB Cluster?

An ESB deployment has two types of components:

- ESB Runtime Server that supports active-active and active-passive configurations.
- ESB Design-time Server that supports active-passive configuration. This component is sometimes referred to as the ESB Repository Server.

ESB Design-time Server (ESBDT) and ESB Runtime Server (ESBRT) must be run on different OC4J containers. In version 10.1.3.1, the ESBDT and ESBRT must then be installed to different Oracle Homes.

Starting in 10.1.3.3, the ESBRT and ESBDT components can run in the same Oracle Home, but in different OC4J containers.

All ESBDTs and ESBRTs in a cluster share the same Metadata repository and the same AQ-JMS as the underlying messaging system.

4.3 The ESB Standalone Installer Prompts for 3 Different Installation Options. Which Option Should I Choose?

The graphic Oracle Universal Installer for ESB standalone prompts for 3 different installation options: Repository, Runtime, and Repository and Runtime.

If you are following the Enterprise Deployment Guide, the Runtime option is used to install ESBRT into the same OC4J container as BPEL PM within an existing Oracle Home. After upgrading to the 10.1.3.3 patchset, the ESBRT can be installed via ant by following section 3.19 of the Enterprise Deployment Guide 10.1.3.3.
The Repository option should only be chosen when installing the ESBDT into its own Oracle Home. Starting in 10.1.3.3, that is no longer required. Note the 10.1.3.1 graphical installer will not allow an ESBDT installation into an existing Oracle Home with ESBRT already deployed. For this reason you must install ESBDT via ant deployment after the 10.1.3.3 patchset is applied, as described in the Enterprise Deployment Guide 10.1.3.3 section 3.19.

The Repository and Runtime option should never be selected for high-availability installations. This installs ESBRT and ESBDT into the same OC4J Container, which is not a valid HA configuration.

4.4 Why Does ESB Clustering Require AQ JMS Messaging?
Before AQ JMS (i.e. database persistence) is configured the individual ESB nodes use their own local JMS artifacts that use file-based persistence. This is not a shared repository and does not offer messaging consistency or shared data. In order to configure a highly available and shared system, database persistence must be setup for all ESBRTs and ESBDTs. This ensures ESB JMS artifacts point to the same place in the database for use in the cluster. When the database itself is setup in a highly available fashion using RAC this provides all ESB components a common place to access JMS artifacts and a redundant database service with continuity after a RAC node failure.

4.5 Why is it Necessary to Add “service-failover="1"” and Remove the “numprocs="1"” in opmn.xml to Configure the ESBDT Cluster?
Adding service-failover="1" configures OPMN to allow only one active ESBDT at any time in the cluster. The numprocs property defines the number of JVMs for that OC4J container. Active-passive topologies cannot be enforced if a container is running multiple JVMs. Removing the numprocs="1" is required to ensure the value is not changed in the future.

If the numprocs property is not removed in conjunction with adding the service-failover attribute, OPMN will not start the container. Use the following command to verify if you have properly configured your opmn.xml file.

> opmnctl validate

Note the active-passive setup of ESBDT is not achieved until the nodes are clustered via OPMN by either multicast or static discovery.

4.6 Within the orion-application.xml File, What Does the primary_oc4j Setting Indicate?
This only applies to version 10.1.3.1 where the primary_oc4j flag setting of “true” means the application should function as an ESBDT instance, and a setting of “false” means the application should function as an ESBRT instance. Starting in 10.1.3.3, this distinction is no longer manually configured and the primary_oc4j parameter is not relevant and can be ignored.
4.7 Where do I Find the *create_esb_topics.sql* File? How do I Verify if the ESB Topics Have been Created?

The Enterprise Deployment Guide section 3.22 references the *create_esb_topics.sql* file. This file is located at `$ORACLE_HOME/integration/esb/sql/oracle`. To verify if the ESB topics exist, login to the database as user oraesb and execute the following SQL statement:

```sql
select object_name, object_type from dba_objects where object_type like 'QUEUE' and object_name like '%ESB_%';
```

In the subsequent display verify that the following queues exist:
ESB_JAVA_DEFERRED, ESB_CONTROL, ESB_ERROR, ESB_ERROR_RETRY, ESB_MONITOR

4.8 What is the Advantage to Deploying a Load-Balancer in Front of the ESB DT Nodes?

The ESB RT nodes poll the ESB DT node for certain metadata information during runtime (e.g. service WSDLs and XSDs for BPEL processes). By incorporating a load-balancer in front of the ESB DT active and passive nodes you can virtualize the IP address of the ESB DT node and this enables a more seamless ESB DT failover. Without the virtual IP address, when a failover occurs an administrator will need to manually reconfigure the ESB RT nodes to poll the new ESB DT active node.

4.9 How Can I Manually Configure the ESB RT Nodes to Point to a New ESB DT?

Run the following SQL script within the oraesb schema, replacing the param_value parameters with your appropriate values.

```sql
update esb_parameter set param_value='<ESBDT hostname >' where param_name = 'DT_OC4J_HOST';
update esb_parameter set param_value='<ESBDT port>' where param_name = 'DT_OC4J_HTTP_PORT';
```

Restart all ESB instances after updated these parameters.

4.10 What is the Runtime Interaction between ESB RT and ESB DT?

The primary mechanism of interaction between ESB DT and the ESB RT nodes is through JMS topics.

The ESB DT subscribes to both a monitor and error topic to which all ESB RT instances publish their monitoring and error messages. ESB DT then persists that information to the database.

The ESB RT subscribes to both an administration and retry topic to which the ESB DT publishes control and retry messages.

As mentioned above, the ESB RT nodes also poll the ESB DT node for certain metadata information during runtime (e.g. service WSDLs and XSDs for BPEL processes).
4.11 Is it Necessary to Install ESBRT and BPEL in the Same Oracle Home?

No. However, depending on the version, ESBRT clustered adapters will attempt to use the jgroups-protocol.xml file in the normal location of a BPEL installation ($OH/bpel/system/config/jgroups-protocol.xml). If not present, the adapters will default to an internal configuration that uses UDP.

4.12 Is it Necessary to Install ESBRT and BPEL in the Same OC4J Container?

No, but when ESBRT and BPEL are installed into the same container, the following benefits apply:

- Native Java calls between ESB and BPEL
- Transaction propagation between ESB and BPEL
- XREF and DVM functions are available in BPEL
- For interactions between BPEL and ESB instances, drill-through from the ESB console flow trace to BPEL console flow trace and vice-versa

4.13 What are the Options for Updating ESB Metadata, i.e. the oraesb.esb_parameter Table?

Option 1: use the provided ant utility as describe in the Enterprise Deployment Guide section 3.24.

Option 2: use the following SQL script. Note you must replace the hostname and port parameter values with the hostname and port to access your ESBDT instance.

```sql
delete esb_parameter where param_name = 'PROP_NAME_DEFERRED_TOPIC_JNDI';
delete esb_parameter where param_name = 'PROP_NAME_INITIAL_CONTEXT_FACTORY';
delete esb_parameter where param_name = 'ACT_ID_RANGE';
insert into esb_parameter values('PROP_NAME_DEFERRED_TOPIC_JNDI','ESBTopics/Topics/ESB_JAVA_DEFERRED');
insert into esb_parameter values('PROP_NAME_INITIAL_CONTEXT_FACTORY','com.evermind.server.rmi.RMIInitialContextFactory');
insert into esb_parameter values('ACT_ID_RANGE', '400');
update esb_parameter set param_value='<hostname of the load balancer serving the ESBDT>' where param_name = 'DT_OC4J_HOST';
update esb_parameter set param_value='<load balancer port serving ESBDT>' where param_name = 'DT_OC4J_HTTP_PORT';
update esb_parameter set param_value = 'OracleOJMS/TCF' where param_name = 'PROP_NAME_DEFERRED_TCF_JNDI';
update esb_parameter set param_value = 'OracleOJMS/XATCF' where param_name = 'PROP_NAME_DEFERRED_XATCF_JNDI';
update esb_parameter set param_value = 'ESBTopics/Topics/ESB_CONTROL' where param_name = 'PROP_NAME_CONTROL_TOPIC_JNDI';
update esb_parameter set param_value = 'OracleOJMS/XATCF' where param_name = 'PROP_NAME_CONTROL_TCF_JNDI';
```
update esb_parameter set param_value = 'ESBTopics/Topics/ESB_ERROR' where param_name = 'PROP_NAME_ERROR_TOPIC_JNDI';
update esb_parameter set param_value = 'OracleOJMS/TCF' where param_name = 'PROP_NAME_ERROR_TCF_JNDI';
update esb_parameter set param_value = 'OracleOJMS/XATCF' where param_name = 'PROP_NAME_ERROR_XATCF_JNDI';
update esb_parameter set param_value = 'ESBTopics/Topics/ESB_ERROR_RETRY' where param_name = 'PROP_NAME_ERROR_RETRY_JNDI';
update esb_parameter set param_value = 'OracleOJMS/XATCF' where param_name = 'PROP_NAME_ERROR_RETRY_TCF_JNDI';
update esb_parameter set param_value = 'ESBTopics/Topics/ESB_MONITOR' where param_name = 'PROP_NAME_MONITOR_TOPIC_JNDI';
update esb_parameter set param_value = 'OracleOJMS/TCF' where param_name = 'PROP_NAME_MONITOR_TCF_JNDI';
update wf_agents set tcf_jndi = 'OracleOJMS/XATCF' where queue_type = 'DEFERRED';
update wf_agents set name = 'ESBTopics/Topics/ESB_JAVA_DEFERRED' where queue_type = 'DEFERRED';
update wf_agents set queue_name = 'ESBTopics/Topics/ESB_JAVA_DEFERRED' where queue_type = 'DEFERRED';
commit;

4.14 What is the Relationship between the ESB Configuration Files esb_config.ini and orion-application.xml?

The settings in $ORACLE_HOME/integration/esb/config/esb_config.ini file apply to all ESB instances in the Oracle Home.

If specified, the $ORACLE_HOME/j2ee/<OC4J-container>/applications/<esb-rt or esb-dt>/META-INF/orion-application.xml overrides the esb_config.ini settings for that ESB instance.

4.15 Where Should I Specify the cluster_name Parameter for ESBRT Instances? Does ESBDT Require a cluster_name Setting?

In practice, it’s best to set the cluster_name for all ESBRTs in a single Oracle Home through the cluster_name property in esb_config.ini. If you have requirements for separate ESB clusters in a single Oracle Home then override the esb_config.ini property in the appropriate orion-application.xml files.

Starting in 10.1.3.3, cluster_name for an ESBDT instance is not relevant; you do not need to set this property in the ESBDT orion-application.xml.

4.16 What are the Correct ESB Console Settings for an ESB System in a Cluster?

Cluster Name: This should match the cluster_name provided in the ESBRT’s esb_config.ini or orion-application.xml file for the ESBRT(s) where you want the deployed services to run.
**Virtual Host**: Hostname of the ESBRT load balancer

**Port**: HTTP port of the ESBRT load balancer

**Topic Location**: ESBTopics/Topics/ESB_JAVA_DEFERRED (as defined in the PROP_NAME_DEFERRED_TOPIC_JNDI of the ESB Metadata)

**Connection Factory Location**: OracleOJMS/XATCF (as defined in the PROP_NAME_CONTROL_TCF_JNDI, PROP_NAME_DEFERRED_XATCF_JNDI, PROP_NAME_ERROR_RETRY_TCF_JNDI, PROP_NAME_ERROR_XATCF_JNDI)

### 4.17 What is the Relationship Between ESB Systems and ESB Runtime Clusters?

ESB systems are groupings of ESB services defined in either JDeveloper or through the ESB console, and visible in the ESB console. Each system is configured for deployment to an ESB runtime cluster on the system definition page (multiple systems can be assigned to the same runtime cluster). The services of that ESB system are then available on nodes running that runtime ESB cluster, i.e. onto nodes where the esb_config.ini or orion-application.xml cluster_name property is a match.

### 4.18 What are the Advantages to Using Multiple ESB Systems in a Clustered Runtime Installation?

The services in an ESB system can be scaled as needed across many runtime servers. Each ESBRT server is assigned to a single cluster that configures it to load services into its memory cache only from that cluster (group of ESB systems).

If all ESB Systems are configured to the same ESB cluster, this is referred to as a symmetric cluster. This provides high availability such that if one ESBRT server goes down other ESBRT servers can process requests.

If ESB systems are configured to different ESB clusters that is called an asymmetric cluster. In this configuration, different ESBRT servers will provide a different set of ESB services.

A hybrid approach can utilize both symmetric and asymmetric ESB runtime clusters to both scale up services in an ESB system as well as make them highly-available by running more than one ESBRT server against each cluster.

### 4.19 How do you Apply a Patch When ESBDT and ESBRT are in Separate Containers?

**Note**: This only applies to 10.1.3.3 installs up to MLR#8. Starting with 10.1.3.3 MLR#9, the patch is automatically applied to all containers in the same Oracle Home.

The scenario is you have installed ESBRT and BPEL into one OC4J container, e.g. OC4J_SOA. ESBDT is installed in the same application server but in a different container, e.g. OC4J_ESBDT. You now want to apply patch on top of this installation, in both containers.

See MetaLink Note 549995.1: "How To Patch ESB RT And ESB DT Running In Separate Container But In Single Oracle Home".
4.20 Are There Other Recommendations for ESBDT Active-Passive Configuration?

OPMN manages the active-passive failover for each ESBDT instance by pinging the active instance to check its state. In order to reduce risk of a false positive (i.e. declaring the active ESBDT down when actually still up), you should increase the ping timeout setting in the ESBDT <process-type> element. The default is timeout value is 20 seconds and by adding the <pingtimeout> element you can increase it. See the OPMN documentation for more details.

<pingtimeout="timeout" retry="num" interval="interval"/>

4.21 Why Doesn’t ESBDT Support an Active-Active Topology?

ESBDT subscribes to both a monitor and error topic to which all ESBRT instances publish their monitoring and error messages. ESBDT then persists that information to the database. Two ESBDT instances active at the same would duplicate instance and error data in the database as each instance inserts the same data. ESBDT also maintains an in-memory cache of metadata; that information could get out of sync with two ESBDTs.
5 Adapters

5.1 Which Technology Adapters Use Active-Passive Clustering?
Oracle SOA Suite includes many technology adapters, of which File and FTP are commonly configured in an active-passive configuration (aka “singleton” setup). This is due to file system restrictions that make it impossible to avoid race conditions when two nodes attempt to retrieve the same file. Note circumstances sometimes require active-passive configuration of AQ or JMS adapters that subscribe to the same messaging topic since each adapter instance will consume the same message.

5.2 How Do I Setup a Singleton Adapter for BPEL?
Within the bpel.xml of the BPEL process, add the “clusterGroupId” property in the JCA Activation Agent for the adapter.

```xml
<activationAgents>
  <activationAgent className="..." partnerLink="MyInboundAdapterPL">
    <property name="clusterGroupId">myBpelAdapterCluster</property>
    <property name="clusterAcrossSubnet">true</property>
  </activationAgent>
</activationAgents>
```

If the BPEL PM servers in the cluster are located across TCP/IP subnet boundaries, then it is necessary to add the attribute “clusterAcrossSubnet=true”.

```xml
<property name="clusterAcrossSubnet">true</property>
```

This configuration uses the default jGroups properties. To override those properties, use the relevant customizations below. Notice the convention of prefixing each customized property name with the name of the cluster group id.

```xml
<activationAgents>
  <activationAgent className="..." partnerLink="MyInboundAdapterPL">
    <property name="clusterGroupId">myAdapterCluster</property>
    <property name="myAdapterCluster_mcast_addr">224.0.0.35</property>
    <property name="myAdapterCluster_mcast_port">45566</property>
    <property name="myAdapterCluster_ip_ttl">32</property>
    <property name="myAdapterCluster_verify_suspect_timeout">1500</property>
    <property name="myAdapterCluster_join_timeout">5000</property>
    <property name="myAdapterCluster_join_retry_timeout">2000</property>
  </activationAgent>
</activationAgents>
```
For more information, see MetaLink Note 730515.1 How to Enable a Singleton Adapter in BPEL Cluster Environment.

5.3 How Do I Setup a Singleton Adapter for ESB?

In an ESB cluster the adapters are deployed into all nodes of the cluster and by default are active in all of the nodes. To avoid race conditions adapters such as File or FTP, it's necessary to configure the inbound adapter to be a singleton.

To configure an ESB adapter as a singleton, open the ESB project in JDeveloper and edit the *.esbsvc file for the adapter and add the following property:

```xml
<endpointProperties>
   <property name="clusterGroupId" value="ESB-Adapter-Cluster"/>
</endpointProperties>
```

Alternatively, add the same endpoint property via the Properties tab of the inbound adapter service in the ESB console.

For more information, see [E10294-02] – Configuring Oracle Enterprise Service Bus for Singleton Adapters, and MetaLink Note 746108.1 How to Enable a Singleton Adapter in an ESB Cluster Environment.

5.4 How Does the Adapter Cluster Work?

The Adapter cluster uses jGroups technology to communicate between adapter instances and maintain an active-passive configuration. There are differences in the specific jGroups configuration used depending on the BPEL or ESB version.

a) In SOA 10.1.3.3 both BPEL and ESB use the jgroups-protocol.xml for the configuration properties.

b) Starting in SOA 10.1.3.3 MLR#11, BPEL adapters use their own internal jGroups configuration (separate from jgroups-protocol.xml and not exposed) with UDP. To force adapters clustering to use the jgroups-protocol.xml, specify the following parameter in the bpel.xml of the BPEL process:

```xml
useJgroupConfigFile=true
```

c) Starting in 10.1.3.4 MLR#5, ESB also uses its own internal jGroups configuration, based on UDP. Similarly, to configure an ESB adapter to use the jgroups-protocol.xml configuration, add the endpoint property as in (b) above.

In a clustered configuration, only one adapter instance will be allowed to start reading or publishing messages. The adapter framework instance initially chooses one at random to assume the primary activation. If a primary activation becomes unresponsive, then any of the remaining members of the cluster group will immediately
detect this, and reassign the primary activation responsibility to one of activation agents standing by. [B31017_01]

5.5 What if ESB and BPEL are Not Installed Together? Does that Effect my Adapter Cluster Configuration?

If ESB-RT and BPEL are installed into different containers and ESB-RT needs the jgroups-protocol.xml file for its jGroups properties (see above for when this is true), edit the file ORACLE_HOME/opmn/conf/opmn.xml and look for the ESB-RT container definition. There, add “-Dorabpel.home” to the JVM startup parameter. This will tell ESB-RT where to find the jgroups-protocol.xml file.

```xml
<process-type id="oc4j_esbrt" module-id="OC4J" status="enabled">
    <module-data>
      <category id="start-parameters">
        <data id="java-options" value="-server ... -Dorabpel.home=/home/oracle/product/10.1.3.1/OracleAS_1/bpel />
      </category>
    </module-data>
</process-type>
```

5.6 Can I Have More Than One Primary Activation Agent to Process High Volume Requests?

No, within one logical cluster group, only one activation agent can be the primary. If you need multiple primary activations, you need as many separate cluster groups. If there is only one cluster group defined, only the (currently) primary activation will process messages from the endpoint.

5.7 What is the Relationship between the Adapter Cluster and the BPEL or ESB Cluster?

The “clusterGroupId” property creates an adapter cluster that is independent of the BPEL or ESB cluster. Depending on the version and your configuration, they can share the same jGroups properties.

Always set an adapter’s “clusterGroupId” to a different value than the names of your BPEL and ESB clusters.

5.8 Should I Use TCP or UDP Communication in an Adapter Cluster?

If there are more than 2 (two) singleton adapters defined in an adapter cluster, use UDP instead of TCP. If the cluster spans across multiple subnets you can still use UDP with the “clusterAcrossSubnet=true” property on each adapter endpoint.

5.9 What Loggers are Useful for Adapters?

The essential BPEL loggers for adapter troubleshooting are:

<domain_name>.collaxa.cube.activation
<domain_name>.collaxa.cube.ws

To change logger log levels, go to the BPEL Console under domain configuration in the Logging tab.
5.10 How Can I Improve Throughput with Active-Passive Adapters in BPEL?

Within BPEL the adapter framework supports a server distribution fan-out pattern for inbound messages. This allows a single active endpoint to evenly spread (round robin) the inbound message it receives to create process instances across the BPEL servers in the cluster.

This load distribution feature helps improve throughput in an active-passive cluster and can be configured in the bpel.xml JCA Activation Agent:

```xml
<activationAgent
    className="oracle.tip.adapter.fw.agent.jca.JCAActivationAgent"
    partnerLink="AdapterInboundPL">
    <property name="bpelServers">
        bpel.host1.net:23791,
        bpel.host2.net:23791,
        bpel.host3.net:23791
    </property>
    <property name="clusterGroupId">myBpelAdapterCluster</property>
</activationAgent>
```

The port numbers are the ORMI (request) port of the OC4J instance where BPEL is deployed. All BPEL servers mentioned in the "bpelServers" property must have the same server credentials.

5.11 How Do I Check if an Activation Agent is Setup Correctly for BPEL?

When starting the container you should see in the OPMN log the following lines that show the JCA Activation Agent for each PartnerLink in every process loaded:

```xml
<2009-02-19 11:25:47,643> <INFO>
<default.collaxa.cube.activation> <AdapterFramework::Inbound>
Loading JCAActivationAgent for {blockingStartStop=false, clusterGroupId=kclugage-adapter-cluster, portType=Get_ptt}
</2009-02-19 11:25:47,643> <INFO>
<default.collaxa.cube.activation> <AdapterFramework::Inbound>
JCAActivationAgent::load - Locating Adapter Framework instance: OraBPEL
<default.collaxa.cube.activation> <AdapterFramework::Inbound>
Instantiating inbound part of Adapter Framework instance: OraBPEL
</2009-02-19 11:25:47,689> <INFO>
<default.collaxa.cube.activation> <AdapterFramework::Inbound>
JCAActivationAgent::load - Done loading JCAActivationAgent for processId='bpel://localhost/default/pool_FTP~1.0/
```
Next the JCA Activation Agent is initialized and will join the adapter cluster, using the UDP protocol in this example:

```
<2009-02-19 11:25:54,533> <INFO>
<default.collaxa.cube.activation> <AdapterFramework::Inbound>
JCAActivationAgent::init - Initializing the JCA activation agent, processId='bpel://localhost/default/pool_FTP~1.0/
<2009-02-19 11:25:54,533> <INFO>
<default.collaxa.cube.activation> <AdapterFramework::Inbound>
JCAActivationAgent joining cluster group kclugage-adapter-cluster1.0
Feb 19, 2009 11:25:55 AM
org.collaxa.thirdparty.jgroups.protocols.UDP createSockets
INFO: sockets will use interface 144.25.146.161
```

### 5.12 How Do I Monitor Activation Agent Failover in BPEL?

Check the OPMN logs of a BPEL container that is still running. You should see a new node is now the Primary Activation in the cluster group.

```
<2009-02-19 17:07:15,590> <INFO>
<default.collaxa.cube.activation> <AdapterFramework::Inbound>
JCAActivationAgent appointed primary activation in cluster group kclugage-adapter-cluster1.0
<2009-02-19 17:07:15,590> <INFO>
<default.collaxa.cube.activation> <AdapterFramework::Inbound>
Adapter Framework instance: OraBPEL - endpointActivation for portType=Get_ptt, operation=Get
<2009-02-19 17:07:15,606> <INFO>
<default.collaxa.cube.activation> <FTP Adapter::Inbound> ENDPOINT ACTIVATION CALLED IN FTP ADAPTER
<2009-02-19 17:07:15,606> <INFO>
<default.collaxa.cube.activation> <AdapterFramework::Inbound>
Adapter Framework instance: OraBPEL - successfully completed endpointActivation for portType=Get_ptt, operation=Get
```

### 5.13 How Do I Monitor Adapter Activation in ESB?

If clustering is setup correctly, a properly configured singleton adapter in ESB will only be active in one ESBRT instance. The container log.xml file of that instance (at $ORACLE_HOME/j2ee/<container_name_ESBRT>:/log/default_group/oc4j/log.xml) will have the following output:
Starting initialization:

```xml
<Message>
  <Header>
    <TSTZ_ORIGINATING>2009-02-20T09:35:41.408-08:00</TSTZ_ORIGINATING>
    <Component_ID>tip</Component_ID>
    <MSG_TYPE TYPE="NOTIFICATION"></MSG_TYPE>
    <MSG_LEVEL>1</MSG_LEVEL>
    <HOST_ID>kclugage-pc2</HOST_ID>
    <HOST_NWADDR>144.25.146.161</HOST_NWADDR>
    <MODULE_ID>esb.server.service.impl.inadapter</MODULE_ID>
    <THREAD_ID>25</THREAD_ID>
    <USER_ID>SYSTEM</USER_ID>
  </Header>
  <Correlation_Data>
    <EXEC_CONTEXT_ID><UNIQUE_ID>144.25.146.161:66483:1235151341408:6</UNIQUE_ID><SEQ>0</SEQ></EXEC_CONTEXT_ID>
  </Correlation_Data>
  <Payload>
    <MSG_TEXT>Creating and initializing inbound JCA endpoint for service: "Get" WSDL location:
    "esb://ESB_Projects/SOACloneTests_Pool_FTP_ESB/pool_ftp_in.wsdl"
    portType: "Get_ptt" activation properties:"Get"</MSG_TEXT>
  </Payload>
</Message>
```

Initialization complete:

```xml
<Message>
  <Header>
    <TSTZ_ORIGINATING>2009-02-20T09:35:41.455-08:00</TSTZ_ORIGINATING>
    <Component_ID>tip</Component_ID>
    <MSG_TYPE TYPE="NOTIFICATION"></MSG_TYPE>
    <MSG_LEVEL>1</MSG_LEVEL>
    <HOST_ID>kclugage-pc2</HOST_ID>
    <HOST_NWADDR>144.25.146.161</HOST_NWADDR>
    <MODULE_ID>esb.server.service.impl.inadapter</MODULE_ID>
    <THREAD_ID>25</THREAD_ID>
    <USER_ID>SYSTEM</USER_ID>
  </Header>
  <Correlation_Data>
    <EXEC_CONTEXT_ID><UNIQUE_ID>144.25.146.161:66483:1235151341408:6</UNIQUE_ID><SEQ>14</SEQ></EXEC_CONTEXT_ID>
  </Correlation_Data>
  <Payload>
    <MSG_TEXT>Successfully finished endpoint activation for operation "DefaultSystem.pool_ftp_in_RS.Get".</MSG_TEXT>
  </Payload>
</Message>
```
5.14 What Additional Configuration is Necessary for File and FTP Adapters in a Cluster?

The File and FTP adapters use a control file to coordinate information between instances, such as the sequence numbering of output files. In a clustered environment, this control file must be shared between all instances.

1. Create a folder on a shared file system. This folder should have write permission and be accessible from all the systems that are running File and FTP adapters. This is the shared folder that will store the control files.

2. Edit the pc.properties file available in the ORACLE_HOME\bpel\system\service\config directory on each of the nodes, and set oracle.tip.adapter.file.controldirpath to the shared folder name.

3. Restart the servers.

Please note that in the case of ESB, you will need to rename ORACLE_HOME\integration\esb\config\pc.properties.esb to pc.properties and make the relevant changes there.

5.15 How Do I Enable Distributed Polling for an Inbound Database Adapter?

Inbound database adapters in an active-active cluster configuration must be coordinated to ensure two instances polling a database table do not retrieve the same rows (thereby creating duplicate instances). Reservation Distributed Polling is the two-step polling strategy used to achieve this coordination. The first step is to reserve the rows; the second step is for processing. There are two types available:

MarkReservedValue (Not Recommended for Clusters)

Each adapter instance is configured with a unique MarkReservedValue that is documented in the BPEL tutorial:

OH/bpel/samples/tutorials/122.DBAdapter/advanced/polling/DistributedPolling

Note: The MarkReservedValue of an adapter is unique only to that node (each adapter thread is assigned its own id), not across the cluster. Thus this type of Reservation Distributed Polling can only be used in non-clustered environments; otherwise duplicate row processing will occur.

Select for Update (Recommended)

With the “select for update no wait”, the rows are reserved and processed in one transaction, so no processing on the same rows can occur in parallel.

To use this feature, within in the database adapter do not input a reserved value and check the “Distributed Pooling” checkbox in the database adapter wizard; Select For Update No Wait will be used.

Note: If you are on 10.1.3.3.x version you need to apply Patch 6690511. This patch is already included in 10.1.3.4.

For more information, see MetaLink Note 556375.1 How to Configure Distributed Polling To Accomplish Logical Delete Strategy for DB Adapter In HA Environment.
5.16 Can MQ and JMS Adapters Operate in an Active-Active Cluster?

JMS adapters can operate as active-active only with DB persisted queues (AQ).

MQ adapters in 10.1.3.1 do not support an active-active cluster, duplicate message reads will occur. The 10.1.3.3 MQ adapter does support an active-active cluster.

For more information, see MetaLink Note 452176.1 - Q: MQ-Adapter and Jms-Adapter: Can they Operate In An Active-Active Cluster?
6 Troubleshooting

6.1 How Can I Verify my ESB Cluster is Setup Correctly?

Reference MetaLink Note 470267.1 How To Verify ESB Cluster Configuration? This provides instruction to verify the following ESB components:

- Register ESB demo project using Ant
- Verify ESB cluster configuration
- Verify Asynchronous routing rule
- Verify Error Handling and Resubmit function
- Delete ESB services

6.2 Invoking ESB Services Registered In ESB Cluster Fails With Error "Cannot find service on current cluster"

Full error message:
oracle.tip.esb.server.common.exceptions.BusinessEventRetriableException: Cannot find service "" on current cluster

Reference MetaLink Note 430348.1: Invoking ESB Services Registered In ESB Cluster Fails With Error "Cannot find service on current cluster".

This is caused by incorrect configuration of the Slide Repository and JNDIs for the Topic and Topic Connection Factory in both the ESBDT and ESBRT servers. If they are not configured correctly, ESB web service providers cannot be generated after registering ESB services. The solution is provided by following the steps in the Enterprise Deployment Guide as outlined in the Note.

6.3 Error in log.xml File During ESB Startup: "Invalid Destination Topics/ESB_MONITOR" and "No resource named 'ojmsRP/Topics/ESB_MONITOR' found"

The following functional symptoms are also present:

- SOAP Endpoint URI shown in a successfully registered ESB service is not accessible in a browser.
- ESB service does not run successfully.

This is caused by incorrect configuration of JNDIs for the Topic and Topic Connection Factories for one or more ESBRTs and ESBDTs.

This can be fixed by following section 3.23 Configuring JNDIs for the Topic and Topic Connection Factory of the Enterprise Deployment Guide 10.1.3.3, making sure that these steps are done on all of the ESBRTs and ESBDTs.

6.4 After Cluster Node Fails, ESB Returns Connection Refused Error

When an ESBRT cluster node fails, the remaining node throws the following exception:
ORABPEL-12600
Generic error.
oracle.tip.esb.server.common.exceptions.BusinessEventRetriableException: An unhandled exception has been thrown in the ESB system. The exception reported is: "org.collaxa.thirdparty.apache.wsif.WSIFException: exception on JaxRpc invoke: HTTP transport error: javax.xml.soap.SOAPException: java.security.PrivilegedActionException: javax.xml.soap.SOAPException: Message send failed: Connection refused

This is caused by connections being cached during failover and can be solved by editing the opmn.xml file to add `-DHTTPClient.disableKeepAlives=true` to the ESBRT JVM startup parameters. Reference MetaLink Note 560228.1: “Upon Cluster Failover, ESB returns Connection refused error.”

6.5 When Deploying a BPEL Process to a Node in the Cluster, it Does Not Propagate Across the Cluster

Prior to 10.1.3.3 MLR#11, in order to propagate across the cluster, the BPEL process must be deployed with an incremental version number. In 10.1.3.3 MLR#11 or greater, process re-deployments with the same version number are automatically propagated.

If the process does not propagate across the cluster, there is most likely an issue with the BPEL jGroups configuration.

A simple test for verifying the clustering configuration is to create a new BPEL domain, then verify the domain folder is created on the file system for each node under $OH/bpel/domains. If the new domain sub-folder is created only on one node the clustering configuration is not correct. See the jGroups related troubleshooting tips in this section.

Note in BPEL 10.1.3.5.1, jGroups was replaced with an internal database-based mechanism to provide cluster communication between BPEL nodes. See the section “Replacement of JGroups for BPEL Clustering...” in this paper for more details.

6.6 Verify jGroups Initializes Correctly

Check the opmn log for the BPEL container for all nodes at $ORACLE_HOME/opmn/logs/<group name><container name><group name>-1.log. This logfile will contain jGroups related information during startup and steady-state operation. Soon after startup you should find log entries for UDP or TCP.

Example jGroups Log Entries for UDP

Apr 3, 2008 6:30:37 PM
org.collaxa.thirdparty.jgroups.protocols.UDP createSockets
INFO: sockets will use interface 144.25.142.172
Apr 3, 2008 6:30:37 PM
org.collaxa.thirdparty.jgroups.protocols.UDP createSockets
INFO: socket information:
local_addr=144.25.142.172:1127, mcast_addr=228.8.15.75:45788, bind_addr=/144.25.142.172, ttl=32
sock: bound to 144.25.142.172:1127, receive buffer size=64000, send buffer size=32000
mcast_recv_sock: bound to 144.25.142.172:45788, send buffer size=64000, receive buffer size=32000
mcast_send_sock: bound to 144.25.142.172:1128, send buffer size=32000, receive buffer size=64000
Apr 3, 2008 6:30:37 PM
org.collaxa.thirdparty.jgroups.protocols.TP$DiagnosticsHandler
bindToInterfaces
\------------------------------------------------------------------\-
GMS: address is 144.25.142.172:1127
\------------------------------------------------------------------\-

Example jGroups Log Entries for TCP
Apr 3, 2008 6:23:39 PM
org.collaxa.thirdparty.jgroups.blocks.ConnectionTable start
INFO: server socket created on 144.25.142.172:7900
Apr 3, 2008 6:23:39 PM
org.collaxa.thirdparty.jgroups.protocols.TP$DiagnosticsHandler
bindToInterfaces
\------------------------------------------------------------------\-
GMS: address is 144.25.142.172:7900
\------------------------------------------------------------------\-

In the log below the "socket created on" indicates that the TCP socket is established on the own node at that IP address and port the "created socket to" shows that the second node has connected to the first node, matching the logfile above with the IP address and port.

Example jGroups Log Entries for TCP, with Connection to 2nd Node
Apr 3, 2008 6:25:40 PM
org.collaxa.thirdparty.jgroups.blocks.ConnectionTable start
INFO: server socket created on 144.25.142.173:7901
Apr 3, 2008 6:25:40 PM
org.collaxa.thirdparty.jgroups.protocols.TP$DiagnosticsHandler
bindToInterfaces
\------------------------------------------------------------------\-
GMS: address is 144.25.142.173:7901
\------------------------------------------------------------------\-
Apr 3, 2008 6:25:41 PM
org.collaxa.thirdparty.jgroups.blocks.ConnectionTable
getConnection
INFO: created socket to 144.25.142.172:7900

Verify that the IP address in the line "INFO: sockets will use interface" (for UDP) or "INFO: server socket created on" and "INFO: created socket to" (for TCP) are correct. If not, refer to the next troubleshooting section with instructions for setting the bind_addr property.
6.7 Verify jGroups is Binding to the Correct Network Interface for UDP or TCP

Machines often have multiple network interfaces, virtual or physical, that can confuse jGroups communication if jGroups binds to the wrong one.

In the UDP or TCP section of jgroups-protocol.xml, specify the machine IP address that jGroups should use for binding to that node’s network interface by adding the bind_addr property. This will prevent jGroups from binding to an incorrect NIC interface/IP address on its own node.

**UDP Settings with bind_addr**

```xml
<config>
  <UDP mcast_send_buf_size="32000"
    ...,
    ucast_recv_buf_size="64000"
    mcast_addr="228.8.15.75"
    bind_addr="<xxx.xxx.xxx.xxx>"
    receive_on_all_interfaces="false"
    loopback="true"
    ...
  </config>
```

**TCP Settings with bind_addr and receive_on_all_interfaces**

```xml
<config>
  <TCP bind_addr="<xxx.xxx.xxx.xxx>"
    receive_on_all_interfaces="false"
    ...
  </config>
```

Alternatively, add the following startup parameter to the JVM settings in opmn.xml. This overrides any value set in the jgroup-protocol.xml file.

```
-Dbind.address=<xxx.xxx.xxx.xxx>
```

Save the opmn.xml file, the execute `opmnctl reload` and `opmnctl restartproc` `process-type=<container>` from the command line.

Note the value of bind_addr will vary from node to node depending on the IP address that the particular node will use to communicate with other members of the cluster.

6.8 Test Connectivity between BPEL Nodes

Test connections between different cluster nodes using ping, telnet, and traceroute. The presence of firewalls and number of hops between cluster nodes can affect performance as they have a tendency to take down connections after some time or simply block them.

Also reference MetaLink Note 413783.1: “How to Test Whether Multicast is Enabled on the Network.”

6.9 Verify UDP jGroups Settings for mcast_addr and mcast_port

**mcast_addr:** set this to a value in the multicast address range of 224.0.0.0 to 239.255.255.255, this value must be the same on all nodes of the cluster. The default is 228.8.15.75, it is not necessary to use this value. Ping the IP address and choose a
different address if you receive any responses as this means that address is already in use. It’s a good idea to check with your network admin as well.

**mcast_port:** set this to a free port on all nodes of the cluster, this value must be the same on each. The default in the file is 45788, it is not necessary to use this value.

### 6.10 Verify TCP jGroups Settings for start_port, initial_hosts and port_range

**start_port:** when BPEL initializes, this is the first port number where jGroups will attempt to establish a TCP socket. If the listed port is not available jGroups will continue to increment by one until a free port is found and bound to successfully. This value can be set to any value and can vary from node to node.

**initial_hosts** and **port_range:** **initial_hosts** is a list of all hosts in the cluster that this node will attempt to connect via jGroups using TCP. The port number enclosed in [] is the first port that the own node will attempt to reach on the listed host. A TCP binding will be established with each port starting from the first port (enclosed by [ ]) and each successive port incrementing by one until the number of ports tried equals the value of **port_range**.

### 6.11 jGroup Nodes May Drop Out of the BPELCluster

Check the jGroup logging within the OPMN log file to ensure the intended nodes have not been dropped from the jGroup cluster. This can happen due to network outages or high CPU utilization on the affected nodes that disrupt the heartbeat messages required to maintain active cluster membership.

To reduce risk of nodes dropping from the cluster you can increase the fault detection **FD timeout** setting. FD timeout specifies the maximum time in milliseconds for a cluster node to respond to a message. A node will be dropped from a cluster after the max number of tries is reached without a response. Both the UDP and TCP sections use FD timeout and max_tries.

Patch 6354719 changed jGroups behavior to automatically rejoin its own node to the cluster, if it drops out for some reason. This patch is available as either a standalone patch in releases 10.1.3.1 and 10.1.3.3 or in 10.1.3.3.1 MLR #2 and above.

Alternatively, restart the BPEL server.

### 6.12 Enable Detailed jGroups Logging

1. Stop the BPEL OC4J container
2. Open the $OH/j2ee/BPEL-OC4J-container/config/j2ee-logging.xml and add the following log_handler for jGroups. This will create a log.xml file in the path directory with jGroups debug messages.

#### jGroups Log Handler Configuration, Part I

```xml
<log_handler name="jgroups-handler"
class="oracle.core.ojdl.logging.ODLHandlerFactory">
<property name="path" value="%ORACLE_HOME%/bpel/system/logs/jgroups"/>
<property name="maxFileSize" value="10485760"/>
<property name="maxLogSize" value="104857600"/>
```
3. In the same file add the following logger, just below the org.quartz logger

```xml
<logger name="org.collaxa.thirdparty.jgroups" level="FINEST"
useParentHandlers="false">
<handler name="jgroups-handler"/>
</logger>
```

4. Start the BPEL OC4J container and you should start to see jGroups trace level messages in the log.xml output file

The above would set the jGroups logger to FINEST which is log4j TRACE level. If you want to see only ERROR message, you can use the em-console to change the level for this "org.collaxa.thirdparty.jgroups" to SEVERE or you can directly edit this file change the level to "SEVERE".

6.13 A New BPEL Domain is Created When a Cluster Node is Down, That Node Fails to Pickup New Domain on Restart

This is addressed by patch 6086453, available in the 10.1.3.1 initial patchset and above.

6.14 BPEL Processes Missing from Console When BPEL Cluster Uses a BigIP Loadbalancer with Persistence Set to On

When BPEL PM restarts, it first checks the connectivity of the soap server url defined in collaxa-config.xml. When the soap server URL is set to a BigIP load balancer virtual hostname, BPEL PM doesn’t receive an HTTP response 200 because a GUI window is opened to confirm user cookie settings. Since this is a BigIP product feature, the only way to get around this is set the persistence settings of BigIP to "simple" where there will be no GUI popup windows.
7 References

These are references for more SOA Suite high-availability installation, configuration, architecture and best practice documents.

[E10294-01] Oracle Application Server Enterprise Deployment Guide 10g Release 3 (10.1.3.3.0)

[SOA HA] Oracle SOA Suite XA and RAC Database Configuration Guide


[E10971-01] Oracle SOA Suite Best Practices Guide 10g Release 3 (10.1.3.3.0)

[B28994-02] Oracle Application Server Adapters for Files, FTP, Databases, and Enterprise Messaging User’s Guide 10g Release 3 (10.1.3.1.0)

[B28941-03] Oracle Application Server High Availability Guide 10g Release 3 (10.1.3.1.0)

[OBE] Setting up the Oracle Web Services Manager High-Availability (HA) Topology

[OBE] Configuring Oracle Web Services Manager for High Availability (HA)

[B28988-02] Oracle Business Activity Monitoring Installation Guide 10g (10.1.3.1.0)


[E10294-02] Oracle Application Server Enterprise Deployment Guide