JD Edwards EnterpriseOne

—

Transaction Server Reference Guide for Microsoft Windows using WebSphere

9.2

JD Edwards EnterpriseOne
Transaction Server Reference Guide for Microsoft Windows using WebSphere

9.2

Part Number: E55832-06

Contents

ORACLE

ORACLE

Preface

Welcome to the JD Edwards EnterpriseOne documentation.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at *http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc* .

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit *http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info* or visit *http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs* if you are hearing impaired.

Related Information

For additional information about JD Edwards EnterpriseOne applications, features, content, and training, visit the JD Edwards EnterpriseOne pages on the JD Edwards Resource Library located at:

http://learnjde.com

Conventions

The following text conventions are used in this document:

Convention	Meaning
Bold	Boldface type indicates graphical user interface elements associated with an action or terms defined in text or the glossary.
Italics	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`Monospace`	Monospace type indicates commands within a paragraph, URLs, code examples, text that appears on a screen, or text that you enter.
> Oracle by Example	Indicates a link to an Oracle by Example (OBE). OBEs provide hands-on, step- by-step instructions, including screen captures that guide you through a process using your own environment. Access to OBEs requires a valid Oracle account.

1 Understanding the Transaction Server Components

Overview

The *JD Edwards EnterpriseOne* Transaction Server Components installer performs the installation, upgrade, and configuration of the Transaction Server Components. These components provide the Real Time Events functionality on the Transaction Server.

This document is not a substitute for the administration manuals provided by your Oracle WebLogic 11g IBM WebSphere Application Server vendor, the network administration manuals provided by your network vendor, or the installation and configuration manuals for third-party products used with *JD Edwards EnterpriseOne* .

The procedures in this guide support the Oracle WebLogic 11g IBM WebSphere Application Server. This document is designed for management information system (MIS) managers and installers. To successfully install the Transaction Server Components on an Application Server, you must have a working knowledge of these topics:

- Hardware and software requirements
- Oracle WebLogic 11g IBM WebSphere Application Server setup and administration
- *JD Edwards EnterpriseOne* platforms and operating systems

The recommended method of obtaining this information is to attend the relevant training courses. Information about course offerings, dates, and locations is available on My Oracle Support. At a minimum, read these guides before beginning:

- *JD Edwards EnterpriseOne Tools Foundation Guide*
- *JD Edwards EnterpriseOne Tools Runtime Administration Guide*
- *JD Edwards EnterpriseOne Tools System Administration Guide*
- *JD Edwards EnterpriseOne Tools Package Management Guide*
- *JD Edwards EnterpriseOne Administration Guide*

Certifications

Customers must conform to the supported platforms for the release as detailed in the JD Edwards EnterpriseOne Certifications (formerly known as Minimum Technical Requirements). In addition, JD Edwards EnterpriseOne may integrate, interface, or work in conjunction with other Oracle products. Refer to the following link for cross-reference material in the Program Documentation for Program prerequisites and version cross-reference documents to assure compatibility of various Oracle products.

http://www.oracle.com/corporate/contracts/index.html

Refer to the Certifications tab on My Oracle Support and search for this product:

- **JD Edwards EnterpriseOne Transaction Server**.

ORACLE

For additional information on using Certifications, refer to this document on My Oracle Support (*https:// support.oracle.com*):

- **Certifications FAQ for JD Edwards EnterpriseOne [Article ID 1525328.1]**

Obtaining the Transaction Server Managed Software Component

You can obtain the Transaction Server Managed Software Component from:

- Update Center
- Change Assistant

The name of the component to download is called *92xTy*, where *x* is the major tools release (for example, 9.2) and *y* is the maintenance release. For example, the Managed Software Component for the Transaction Server for 9.2.3.0 is called 92T30.

> **Note:** Once obtained, the Transaction Server Managed Software Component is installed using the JD Edwards EnterpriseOne tool called Server Manager. Refer to the *JD Edwards EnterpriseOne Tools Server Manager Guide* for additional details.

Prerequisites

These prerequisites are required prior to installing the Transaction Server Components on an Application Server.

- Verify that your installation meets the prerequisites for installing the IBM WebSphere Application Server at this link: Oracle WebLogic Server.

  http://www-01.ibm.com/software/webservers/appserv/was/library/index.html

- IBM WebSphere Oracle WebLogic Application Server must be installed, configured and running. For *JD Edwards EnterpriseOne* -specific installation instructions refer to the *JD Edwards EnterpriseOne HTML Server Reference Guide for IBM WebSphere on Microsoft Windows JD Edwards EnterpriseOne HTML Server Reference Guide for IBM WebSphere on UNIX JD Edwards EnterpriseOne HTML Server Reference Guide for IBM WebSphere on IBM i JD Edwards EnterpriseOne HTML Server Reference Guide for Oracle WebLogic Server on Microsoft Windows JD Edwards EnterpriseOne HTML Server Reference Guide for Oracle WebLogic Server on UNIX* , which is located on OTN (Oracle Technology Network).

  > **CAUTION:** You must use an Oracle administration user to install and administer an application server on Unix. Do not run any Oracle administration commands under root user. However, you must install and configure the application server as the root user.

ORACLE

Understanding Transaction Server Communications

The JD Edwards EnterpriseOne Server Manager communicates with the Transaction Server using JMX (Java Management Extensions). The Embedded Agent in the Transaction Server does not connect to HTTP nor does it connect using Plugins. This is the mechanism by which Server Manager manages the Transaction Server.

The Embedded Agent generally allocates ports in an incremental manner. For example, if a Server Manager Agent installed on the machine is running on port number 14502 (default) then the Embedded Agents running inside managed instances on that machine would allocate ports starting from 14502+1=14503, 14504, and so on depending on the number of managed instances on that machine.

> **Tip:** In order to ascertain the Port Number on which the Transaction Server Embedded Agent is running, open the `SystemOut.log` file of the WebSphere J2EE Container and search for the string `*Management Agent*` which should take you to a line containing this string:
>
> ```
> 00000017 Server I Starting the management agent listener on port '14503'
> ```

The Transaction Server communicates with the JD Edwards EnterpriseOne Enterprise Servers using the EnterpriseOne Proprietary Communication Protocol; that is, JDENET messages.

The Transaction Server connects to the JD Edwards EnterpriseOne database using proprietary database middleware (JDBj). The default ports are dependent on the database type:

- 1521 for Oracle database
- 1433 for SQL Server database
- 50000 for *DB2 for i* or *DB2 for Linux, Windows, and UNIX* database

Although the above are default ports, it is possible to run these databases on different ports. Further, these ports are only defined for the Bootstrap database. The JDBj code retrieves the port numbers for the other databases from the Data Source Master Table based on the port numbers for each of the Data Source records as configured during the Planner and Workbench and the Work with Data Sources Application.

ORACLE

2 Configuring the Transaction Server Components

You must use Server Manager Console to install and configure Transaction Server instances. These Server Manager settings for the Transaction Server are found under Configuration section of Transaction Server Instance.

JDBj Database Configuration

These sections of Server Manager control the JDBj settings:

- *JDBj Bootstrap Datasource*
- *JDBj Bootstrap Session*
- *JDBj Connection Pools*
- *JDBC Drivers*
- *JDBj Logging*
- *Oracle Database Settings*
- *JDBj Runtime Properties*
- *JDBj Spec Datasource*

JDBj Bootstrap Datasource

This section defines the data source where OCM and some other system tables reside. JDBj uses this at bootstrap time and later to look up OCM entries on demand. If the information entered here is not valid, you cannot access the JD Edwards EnterpriseOne database.

With a Transaction Server instance selected, the following shows typical values:

ORACLE

Below is a list of the settings and typical values:

- *System Datasource Name*
  Enter the name of your System datasource. For example:
  System - 920

- *Database Type*
  Use the pulldown to select your database type.

- *Database Name*
  Enter your database name.

- *Database Server Name*
  Enter the name of your database server.

- *Database TCP/IP Port*
  Enter the TCP/IP port for your database.

ORACLE

- *Physical Database*
  Optionally you can choose to specify your physical database.

- *Object Owner*
  Enter the object owner for the System datasource. For example:
  SY920

JDBj Bootstrap Session

This information is used to sign on with access to system tables. For JD Edwards EnterpriseOne HTML servers, only the user and environment need to be supplied. JDBj signs this user on during the bootstrap process. If the sign on information is not valid, then no access is granted to the JD Edwards EnterpriseOne database. To configure as a Portal WSRP Producer, you must supply a valid EnterpriseOne user and environment and the password and role settings are optional.

With a Transaction Server instance selected, the following shows typical values:

Below is a list of the settings and typical values:

- *Bootstrap User*
  Enter the name of your Bootstrap User. For example:
  JDE

- *Bootstrap User*
  Enter the password for your Bootstrap User.

- *Bootstrap Role*
  Enter the role of your Bootstrap User. For example:
  *ALL

ORACLE

- *Bootstrap Environment*

   Enter the bootstrap environment. For example:

   JDV920

JDBj Connection Pools

These settings control the database connection pooling (sharing of connections) within JDBj. Connections are pooled at the physical datasource level, not the EnterpriseOne logical datasource level.

With a Transaction Server instance selected, the following shows typical values:

Below is a list of the settings and typical values:

- *Minimum Connections*

   Enter a value. For example:

   0

ORACLE

- *Maximum Connections*

  Enter a value. For example:

  50

- *Pool Growth Size*

  Enter a value. For example:

  5

- *Initial Connections*

  Enter a value. For example:

  5

- *Connection Timeout*

  Enter a value in milliseconds. For example:

  1800000

- *Pool Cleaning Interval*

  Enter a value in milliseconds. For example:

  300000

- *Maximum Prepared Statements*

  Enter a value. For example:

  50

- *Cache Purge Size*

  Enter a value. For example:

  5

JDBC Drivers

These settings specify which JDBC drivers to use for a particular database type.

With a Transaction Server instance selected, the following shows typical values:
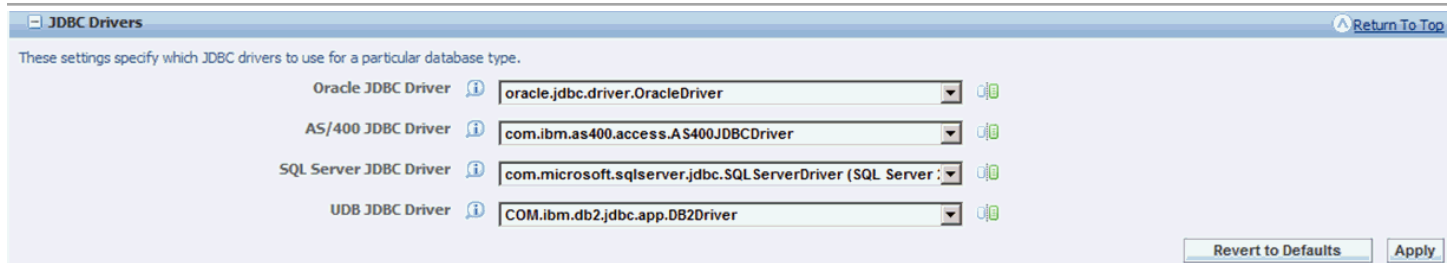
ORACLE

Below is a list of the settings and typical values:

- *Oracle JDBC Driver*

   Enter a value. For example:

   oracle.jdbc.driver.OracleDriver

- *DB2 for i JDBC Driver*

   Enter a value. For example:

   com.ibm.as400.acess.AS400JDBCDriver

- *SQL Server JDBC Driver*

   Enter a value. For example:

   com.microsoft.sqlserver.jdbc.SQL ServerDriver (SQL Server)

- *UDB JDBC Driver* (for *DB2 for Linux, Windows, and UNIX*)

   Enter a value. For example:

   COM.ibm.db2.jdbc.app.DB2Driver

JDBj Logging

This setting enables JDBC tracing from the JDBC drivers.

With a Transaction Server instance selected, the following shows typical setting:
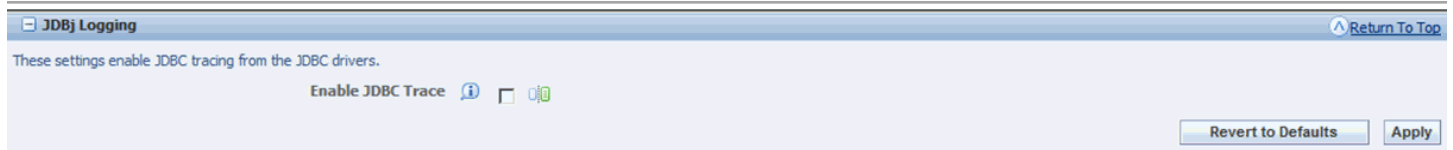
ORACLE

- *Enable JDBC Trace*

  Click the checkbox to enable or disable a JDBC trace from the JDBC drivers.

Oracle Database Settings

This section contains settings that are specific to utilizing Oracle databases. The TNSNAMES.ORA must contain the appropriate connection information for the Oracle database(s) this product will utilize. You may either select an existing location or use the text area to create a new file and update its contents. The TNSNAMES.ORA that is configured for the Management Console (home instance) will be used as a template for additional web products that are installed.

With a Transaction Server instance selected, the following shows typical values:

ORACLE
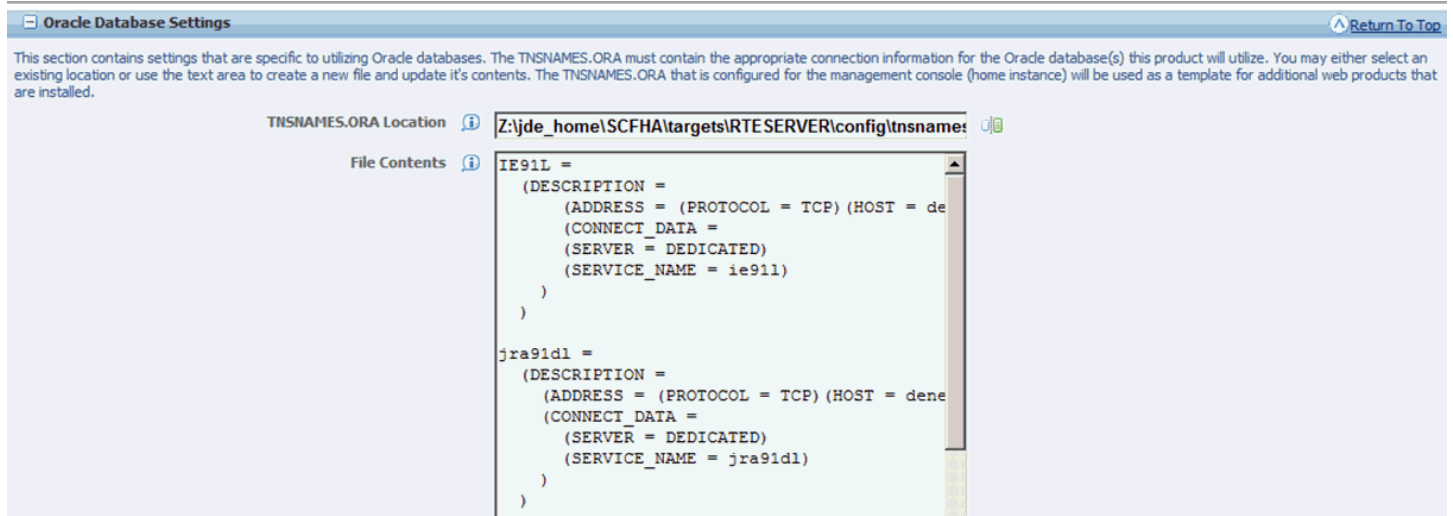
Below is a list of the settings and typical values:

- *TNSNAMES.ORA Location*

  Specifies the location of the TNSNAMES.ORA file.

- *File Contents*

  Displays the contents of the TNSNAMES.ORA file that will be used by the Server Manager Management Console.

JDBj Runtime Properties

These settings correspond to system runtime properties applicable to JDBj logical connections.

With a Transaction Server instance selected, the following shows the typical value:

ORACLE

⊟ **JDBj Runtime Properties**   ⌃ Return To Top

These settings correspond to system runtime properties applicable to JDBj logical connections.

Transaction Timeout ⓘ [0_____] ▯▮

Revert to Defaults   Apply
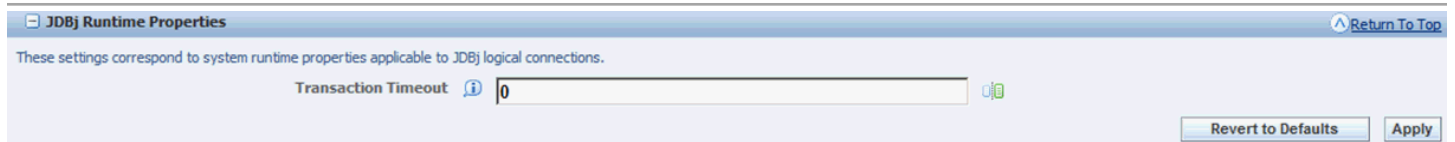
Below is the setting:

- *Transaction Timeout*

    Specifies a value for the transaction timeout. The default is **0**.

JDBj Spec Datasource

This optional section defines the data source where the serialized spec tables reside. JDBj uses these parameters at bootstrap time and after to look up serialized spec objects on demand. If this section is not specified, then JDBj will use the OCM to find the serialized table. If it is not valid, no JD Edwards EnterpriseOne database access is possible.

With a Transaction Server instance selected, the following shows the typical values:

ORACLE

JDBj Spec Datasource ∧ Return To Top

This optional section defines the data source where the serialized spec tables reside. JDBj uses these parameters at bootstrap time and after to look up serialized spec objects on demand. If this section is not specified, then JDBj will use the OCM to find the serialized spec table. If it is not valid, no JD Edwards EnterpriseOne database access is possible.

Field	Value
System Datasource Name	
Database Type	Oracle Database
Database Name	
Database Server Name	
Database TCP/IP Port	0
Physical Database	
Object Owner	
Supports Large Objects (LOBS)	☐
Unicode Database	☐
Database User Name	
Database Password	**********

Revert to Defaults   Apply

Below are the settings:

- *System Datasource Name*

  Specifies a value for the System datasource name.

- *Database Type*

  For example, a valid database type is **Oracle Database**.

- *Database Name*

- *Database Server Name*

- *Database TCP/IP Port*

- *Physical Database*

ORACLE

- *Object Owner*

- *Supports Large Objects (LOBs)*

- *Unicode Database*

- *Database User Name*

- *Database Password*

Real Time Events

This section describes these topics:

- *Event Processor Configuration Settings*

- *Trigger Listener Kernel*

Event Processor Configuration Settings

With a Transaction Server instance selected, the following shows the typical values for Event Processor Configuration Settings:

Event Processor Configuration settings

Sequencing Behavior	Events can be delivered in any sequence.
Processing Concurrency	1
Trigger Listener Delay	10000
Max Transfer Session Size	5000
Initial Context Factory	weblogic.jndi.WLInitialContextFactory
JNDI Provider URL	t3://dnvmiedp12:1014
Event Transfer Transaction Isolation	serializable
Event Client Session Check Interval	60000
Event Client Session Timeout	3600000
App Server JNDI User	**********
App Server JNDI Password	**********

Revert to Defaults   Apply

ORACLE

Below are the settings:

- *Sequencing Behavior*

  This setting describes how the event sequencing occurs during Transaction Server processing.

  Event sequencing behavior values are:

  - *AnySequence*

    Events can be delivered in any sequence.
  - *ByEventType*

    Events are guaranteed to be delivered in sequence by event type.
  - *ByUserSession*

    Events are guaranteed to be delivered in sequence by originating user session.
  - *GlobalFIFO*

    Events are guaranteed to be delivered in sequence by their original generation sequence. This option does not permit concurrent processing and may impact performance. When this option is selected the **Processing Concurrency** value is ignored.

- *Processing Concurrency*

  This setting describes the number of concurrent event processing threads to run. Maximum of 4 concurrent threads can run enhancing the Transaction Server processing performance.

- *Trigger Listener Delay*

  This setting specifies the time, in milliseconds, between timer checks of the database for events waiting to be processed. A negative value will disable the timer.

- *Max Transfer Session Size*

  This setting specifies the maximum number of events to transfer in one transfer session (that is, one result set from the database).

- *Initial Context Factory*

  The InitialContextFactory for the JNDI service of the event server. This value is set as per the transaction server environment.

- *Provider URL*

  The Provider URL for the JNDI service of the event server. This value is automatically set as per the transaction server environment.

- *Event Transfer Transaction Isolation*

  The transaction isolation level to be used by the transfer agent when reading committed events from the database.

  Suggested values based on database server type hosting the F90710 table are shown in the following table:

Server Type	Isolation Level
AS/400	serializable

ORACLE

Server Type	Isolation Level
Oracle	serializable
SQL Server	serializable
DB2 for Linux, Windows, and UNIX	serializable

- *Event Client Session Check Interval*

   This configuration is to check the frequency to check for an expired session.

- *Event Client session Timeout*

   This configuration is to check how long a session can be idle before it is timed out.

- *App Server JDNI User*

- *App Server JDNI Password*

Trigger Listener Kernel

This setting applies to the TriggerListener Kernel thread responsible to receive messages from the Enterprise Server that an event has been triggered. All of the configuration settings found under this section are predefined and should **not** be changed.

ORACLE

Network Settings

This section describes these topics:

- *JDENET Configuration*
- *Security Server Configuration*

JDENET Configuration

These settings configure how JDENET, which is the EnterpriseOne messaging protocol, is configured. Any changes to the JDENET configuration will require restarting the server to take effect.

Below are the settings:

- *Outgoing JDENET Port*
- *Incoming JDENET Port*

ORACLE

- *Kernel Ranges*
- *Timeout Threshold*
- *Pool Size*
- *Back Off Time*
- *Enqueue Timeout*

Security Server Configuration

These settings configure the EnterpriseOne Enterprise Server to use for security services for this instance. Any changes you make here will not take effect until the server is restarted.

Below are the settings:

- *Security Server Count*
- *Primary Security Server*
- *Secondary Security Server*
- *Third Security Server*
- *Fourth Security Server*
- *Fifth Security Server*
- *Token Re-Authenticate Delay*

> **Note:** The Token Re-Authenticate Delay setting can be configured to refresh the RTE token used for connection at regular intervals.

Log File Configuration (jdelog.properties)

This section is used to configure logs. You may add, remove, or configure the log files contained within this section. You must click the **Apply** button on this form to save any changes made to this Log Configuration section.

ORACLE

Log File Configuration

Shown below are the log configurations found in the `jdelog.properties` for this instance. You may add, remove, or configure the log files contained within.

It is necessary to click the **Apply** button below this form to save the changes made to the Log Configuration.

Select [Log Configuration]:   Delete

Select All | Select None

	Log Name ↓ ⓘ	Log File Name ⓘ	Log Level Threshold ⓘ	Log Format ⓘ	Append Log Files ⓘ	Max. Log Size ⓘ	Max. Backup Index ⓘ	Log Components ⓘ
☐	E1LOG	Z:\jde_home\SCFHA\targets\RTESERVER\logs\e1root.log	Warnings and Recoverable Errors ▾	Application Format ▾	TRUE ▾	10MB	20	ALL / LIB

Create New Log Configuration

Apply

3 Configuring JD Edwards EnterpriseOne

After configuring your Transaction Server, you must set up the Transaction Server Components data for your *JD Edwards EnterpriseOne* environments to properly generate events.

Verifying User Defined Codes (UDCs)

The Transaction Server Components requires the specific User Defined Codes (UDCs) to be populated as shown in this table.

> **Note:** By default, the UDCs are delivered and populated correctly. The information in this section is only provided for verification purposes.

UDC	Purpose
H90/EC	Event Category
H90/TP	Event Transport Parameter Type

Verifying Tables

The Transaction Server Components requires the use of the tables listed below. The default configuration for RTE is where all of the F907* tables (event tables) reside in one datasource. Out of the box these tables are included in the System datasource database (that is, SYxxx -- where xxx is the release, for example, SY920). Using this default configuration, only one Transaction Server may be configured to monitor the F907* tables. Verify that these tables were created in the *JD Edwards EnterpriseOne* environments that will generate Real Time Events (RTE) events. Ensure that the server map OCMs and client map OCMs point to the same set of tables.

Table	Description
F90701	Interoperability Event Definition
F90705	Event Activation
F90706	Event Subscriber
F90707	Event Subscription

ORACLE

Table	Description
F907071	Subscribed Events
F907072	Subscribed Environments
F90708	Event Sequence
F90710	Event Transfer
F90711	Event Transport Parameters
F90712	Event Transfer Failure
F90715	Event Subscriber Queues
F90720	Event Metrics
F90730	Unique Sequence

Verifying Serialized Objects

Serialized objects for tables, specs, and data structures must exist on the database machine. Verify that the serialized objects have been generated and copied to the database machine. For more information on performing the generation, see the *JD Edwards EnterpriseOne HTML Web Server Reference Guide*, specific for your platform, and the *Generating JD Edwards EnterpriseOne Serialized Objects* chapter.

Activating Events

Using this procedure, you must use *JD Edwards EnterpriseOne* universal batch engine (UBE) processes to convert the existing event data to be compatible with the new tables.

1. Run the Populate Event Activation Status Table (R90705) to create an activation record for every environment for every event defined in the F90701 table. Through use of the processing options you can process this UBE

for all events and all environments or for a single event and/or environment. You can run this UBE multiple times if necessary.
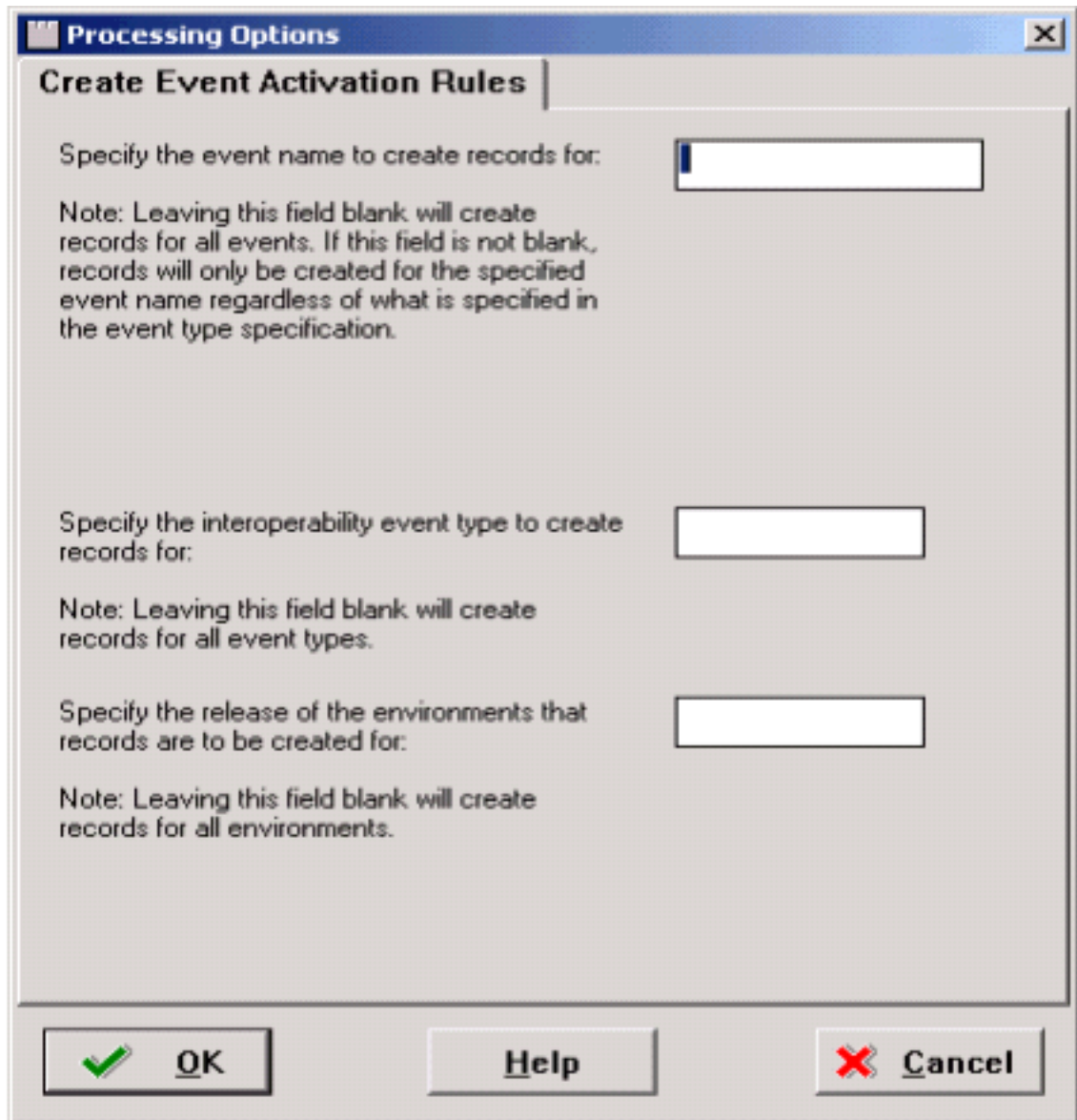
2. On Processing Options, Create Event Activation Rules, complete these fields:

Field	Description
Specify the event name to create records for:	If you leave this field blank, records are created for all events. If you specify a value for this field, records are only created for the specified event name regardless of what is specified in the event type specification.
Specify the interoperability event type to create records for:	If you leave this field blank, records are created for all event types. Valid event types are: o RTE

ORACLE

Field	Description
	○ WF ○ XAPI ○ ZFILE
Specify the release of the environments that records are to be create for:	If you leave this field blank, records are created for all environments. Values are supported releases of *JD Edwards EnterpriseOne*, specified by internal designations.

> **Note:** You may use data selection to select on events that are currently active in the event system to convert.

3. Run the Convert Event Subscriptions (R90706) UBE to populate the Queue Assignment Table (F90715) with the subscriber queues defined in the Transaction Server. You can use processing options to initially create your subscriber queue records and convert your subscribers and subscriptions. Later if you add new subscribers

you can use this UBE to only create the new subscriber queue records. Initially there are 16 subscriber queues delivered with JD Edwards EnterpriseOne.

Note: The R90706 program does not create the subscriber queues. Instead, 16 of these subscriber queues were created during the initial installation of the Transaction Server Components on an Application Server. If you have more than 16 subscribers you will need to create the additional queues using Application Server administrator tools prior to running this program.

Processing Options

Conversion Parameters

Enter the number of subscriber queues to create: `30.00`

Enter the description that will be assigned to all converted subscriptions. If this field is left blank the default description will be "Subscription converted by R90706".

Do not convert subscriptions: `0`

Enter a '1' to prevent batch application from converting subscribers and subscriptions. If value is '1' then batch application will only create the number of subscriber queues specified.

OK   Help   Cancel

4. On the Conversion Parameters tab, complete these fields:

Field	Description
Enter the number of subscriber queues to create:	Set this value to the number of subscriber queues for your installation. You must create a queue for every subscriber.

Field	Description
Enter the description that will be assigned to all converted subscriptions. If this field is left blank the default description will be "Subscription converted by R90706".	Enter a value to describe all converted description. If you leave this field blank the default description is "Subscription converted by R90706".
Do not convert subscriptions	Values are: *0*: Convert subscriptions *1*: Do not convert subscribers and subscriptions. The R90706 program only creates the number of subscriber queues that you specify and does not perform any conversions.

> **Note:** You may use data selection to select on events that are currently active in the event system to convert.

5. Run the Synch Z Events in F47002 with F90701 (R90701A) UBE to create event records in the F90701 for the z-event records in the F47002 table for records that do not currently exist in the F90701 table. There are no processing options for this UBE.

Creating Triggers

You must create a trigger on the F90710 table. Prior to running the trigger creation script, ensure that you have adequate database privileges required to create a trigger. This table lists the privileges by database.

> **Note:** Additionally, as described in the Prerequisites sections of this guide, you must obtain and install a specific ESU to enable the creation of triggers for Transaction Server Components.

Database	Required Privilege
DB2 for i	No special privileges are required.
DB2 for Linux, Windows, and UNIX	Grant the following privileges to the database proxy user. Replace `jde` in the following script with the actual database proxy user name. `GRANT CONTROL ON TABLE SY920.F90710 TO USER jde`

Oracle	Grant the following privileges to the JDE_ROLE or the database proxy user.
	
	`GRANT CREATE ANY SEQUENCE TO JDE_ROLE;`
	`GRANT DROP ANY SEQUENCE TO JDE_ROLE;`
	`GRANT CREATE ANY TRIGGER TO JDE_ROLE;`
	`GRANT DROP ANY TRIGGER TO JDE_ROLE;`

Database	Required Privilege
SQL Server	The database proxy user must have the privileges to create database triggers. Replace **jde** with the actual database proxy user name. `Exec sp_addrolemember db_owner,jde`

You must create the (F90710) Event Transfer and the (F90730) Unique Sequence in the same *JD Edwards EnterpriseOne* data source. You should verify this configuration using OCM.

1. Run this script from a *JD Edwards EnterpriseOne* Microsoft Windows-based Web Development Client:

   `x:\JDEdwards\ E900 \system\bin32\dbtemplates.exe -create`
2. After the scripts complete successfully, you should revoke any special privileges you granted in order to create the triggers. This table lists the revocation commands by database.

Database	Required Privilege
DB2 for i	No special privileges are required.
DB2 for Linux, Windows, and UNIX	Revoke the following privileges to the database proxy user. Replace **jde** in the following script with the actual database proxy user name. `REVOKE CONTROL ON TABLE SY920.F90710 TO USER jde`

Oracle	Revoke the following privileges to the JDE_ROLE or the database proxy user.
	
	`REVOKE CREATE ANY SEQUENCE  TO JDE_ROLE;`
	`REVOKE DROP ANY SEQUENCE   TO JDE_ROLE;`
	`REVOKE CREATE ANY TRIGGER  TO JDE_ROLE;`
	`REVOKE DROP ANY TRIGGER TO JDE_ROLE;`

SQL Server	Replace **jde** with the actual database proxy user name. `exec sp_droprolemember db_owner,jde`

Activating Journaling

> **Note:** This procedure is only applicable to *DB2 for i JD Edwards EnterpriseOne* databases running on *IBM i* machines.

ORACLE

As described by this procedure, for *DB2 for i* databases you must activate journaling on the Unique Sequence (F90730) table.

1. Verify that journaling is activated by entering this command on the *IBM i* machine:

   `WRKOBJ OBJ(library_name/table_name)OBJTYPE(*FILE)`

   Where `table_name` is F90730.

   Select option 8. Page down to see Journaling information. If `Currently journaled` is `No`, it means Journaling is not turned on.

2. To turn journaling on for the table enter this command:

   `STRJRNPF FILE(library_name/table_name JRN(OWJRNL/OW_JRNL)`

   Where `table_name` is F90730.

ORACLE

4 Configuring Subscribers in JD Edwards EnterpriseOne

This chapter discusses the configuration of JMS Queue and JMS Topic subscribers in JD Edwards EnterpriseOne.

Prerequisites

These prerequisites must be met prior to configuring JMS topic in JD Edwards EnterpriseOne and WebSphere:

- Your WebSphere Application Server must be installed. Supported versions of WebSphere are listed in the Certifications. For instructions on accessing Certifications, refer to the chapter of this guide entitled: *Certifications*.
- The JD Edwards EnterpriseOne Transaction Server components are installed, deployed, and functioning properly. You can use Server Manager to perform these tasks.
- Your WebSphere Application Server must be configured for the corresponding JMS Queue/Topic as described in the following tasks in this section entitled: *Configuring Subscriber JMS Queue/Topic in WebSphere Application Server*.

Configuring Subscriber JMS Queue/Topic in WebSphere Application Server

JDEdwards EnterpriseOne supports JMSQUEUE and JMSTOPIC subscribers of JMS type. To use these subscribers, you must create an appropriate Queue and Topic in the Websphere Application Server. It is recommended that you create the Queue and Topic in a container other than the RTE Server container; however subscriber Queue and Topic on RTE Server container is also supported.

Creating JMS Queues in a Transaction Server container Installed in WebSphere Application Server

To create a JMS queue in a Transaction Server container installed in WebSphere Application Server:

1. Start the Transaction Server from the Server Manager Management Console.
2. Open the WebSphere Administrative Console.
3. Expand the **Resources** node in the left panel.
4. Expand the **JMS** node.

ORACLE

5. Click the link for **JMS Providers**.

6. Click the Default messaging provider whose scope contains this value:

   server=RTE_Container

7. Under **Additional Properties**, click **Queue connection factories**.

The following sequence of screens shows the fields and values on the **JMS providers** screen in WebSphere:

ORACLE

JMS providers   ?

JMS providers > Default messaging provider > Queue connection factories > New

A JMS queue connection factory is used to create connections to the associated JMS provider of JMS queues, for point-to-point messaging. Use queue connection factory administrative objects to manage JMS queue connection factories for the default messaging provider.

Configuration

General Properties

   Administration

    Scope

    Node=mcipolla-vm5Node04,Server=TRE_Temp

    Provider

    Default messaging provider

    ＊ Name

    TestQueueConnectionFactory

    ＊ JNDI name

    jms/TestQueueConnectionFac

    Description

    Category

  Connection

    ＊ Bus name

    ORABus-RTE_TMP

    Target

    Target type

    Bus member name

    Target significance

    Preferred

The additional properties will not be available until the general properties for this item are applied or saved.

Additional Properties

   ■ Connection pool properties

Related Items

   ■ JAAS - J2C authentication data

   ■ Buses

ORACLE

Target inbound transport chain

Provider endpoints

Connection proximity

Bus

Quality of Service

Nonpersistent message reliability

Express nonpersistent

Persistent message reliability

Reliable persistent

Advanced Messaging

Read ahead

Default

Temporary queue name prefix

Pass message payload by reference

☐ Applications using this Connection Factory to send messages:
- do not modify the data Object contained in a JMS Object Message
- populate a JMS Bytes Message using a single call to writeBytes(byte[]) and do not modify the byte array once it is contained in the message. Read the help before selecting this option.

☐ Applications using this Connection Factory to receive messages:
- do not modify the data Object obtained from a JMS Object Message. The data Object is treated as read only. Read the help before selecting this option.

ORACLE

Advanced Administrative

☐ Log missing transaction contexts

☐ Manage cached handles

☐ Share data source with CMP

Security settings

Select the authentication values for this resource.

Authentication alias for XA recovery
(none) ▾

Mapping-configuration alias
(none) ▾

Container-managed authentication alias
(none) ▾

8. On the WebSphere **Queue Connection Factories** page, click the **New** button and complete these fields:

Field	Value
Name	TestQueueConnectionFactory
JNDI Name	jms/TestQueueConnectionFactory
Bus name	Select the Bus name created for Transaction Server.

9. Click OK to save the details.
10. Click **Default messaging provider** in the top breadcrumb.

ORACLE

11. Click **Queues** under **Additional Properties** to view a list of queues available for the Transaction Server.

The following screen shows an example of a Queues screen in WebSphere.

ORACLE

12. Click the New button and complete the required fields as shown in the following screens:

ORACLE

Time to live

_____ milliseconds

Priority

_____

Advanced

Read ahead

Inherit from connection factory ▾

Message control across multiple queue points (supported from WebSphere Application Server V7 onwards)

☐ Restrict messages to the local queue point if a queue point is configured on the connected messaging engine

Control across multiple queue points per MessageProducer

Local queue point preference

◉ Prefer to send messages to a local queue point

◯ Do not prefer a local queue point over other queue points

Message affinity across queue points

◯ Send all messages to the same queue point

◉ Messages may be sent to different queue points

Control across multiple queue points per MessageConsumer or QueueBrowser

Message visibility

◉ Only messages on a single queue point are visible

◯ Messages on all queue points are visible

ORACLE

13. For the **Queue name** field, use the pulldown to select the option **Create Service Integration Bus Destination** as shown in the following screen:
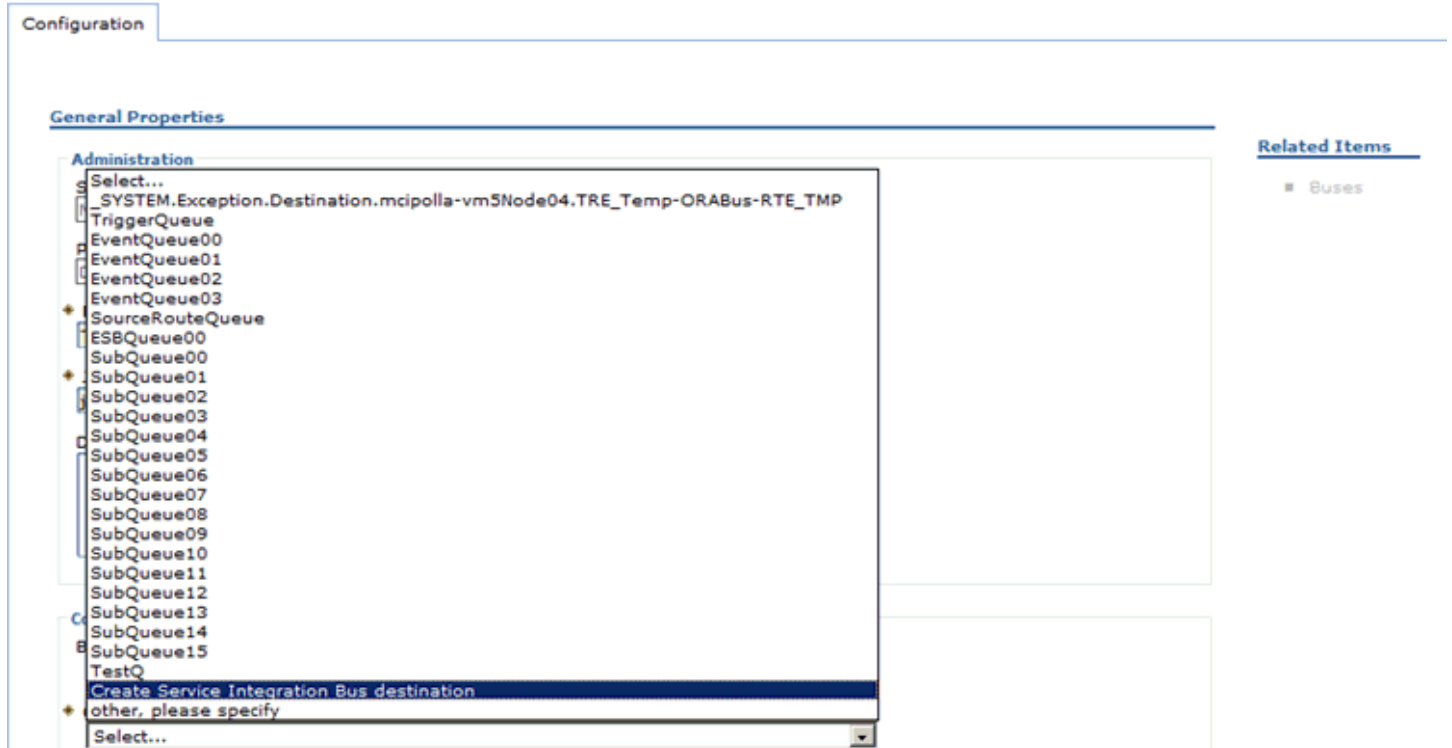
After you select **Create Service Integration Bus destination** from the pulldown, the following **Create new queue** dialog is displayed:

ORACLE
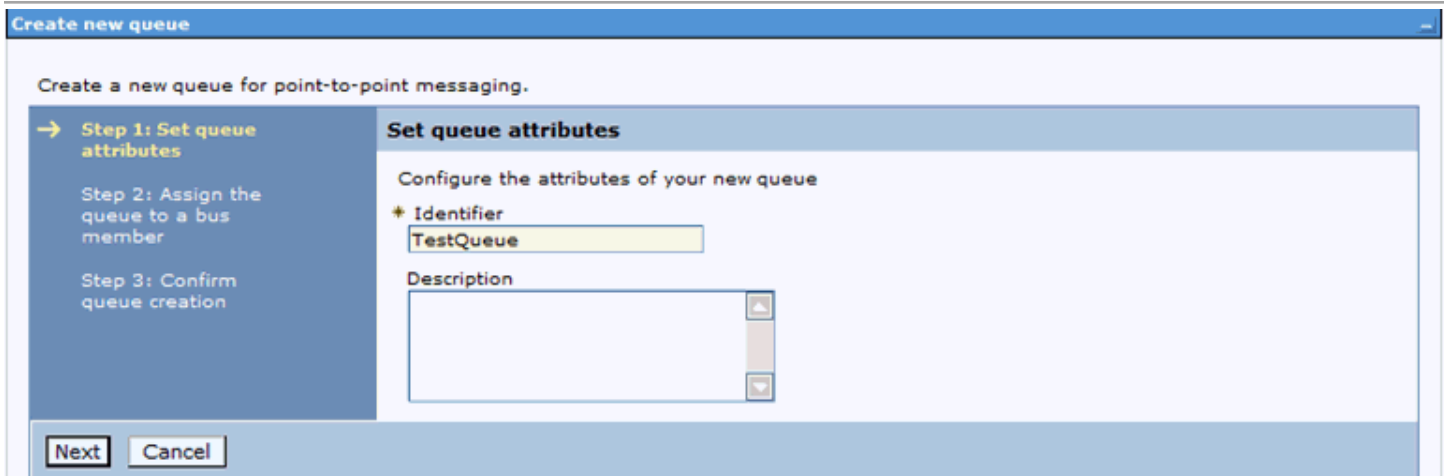
14. On Create New Queue - Step 1: Set queue attributes, enter a value for the required **Identifier** field and click the **Next** button.
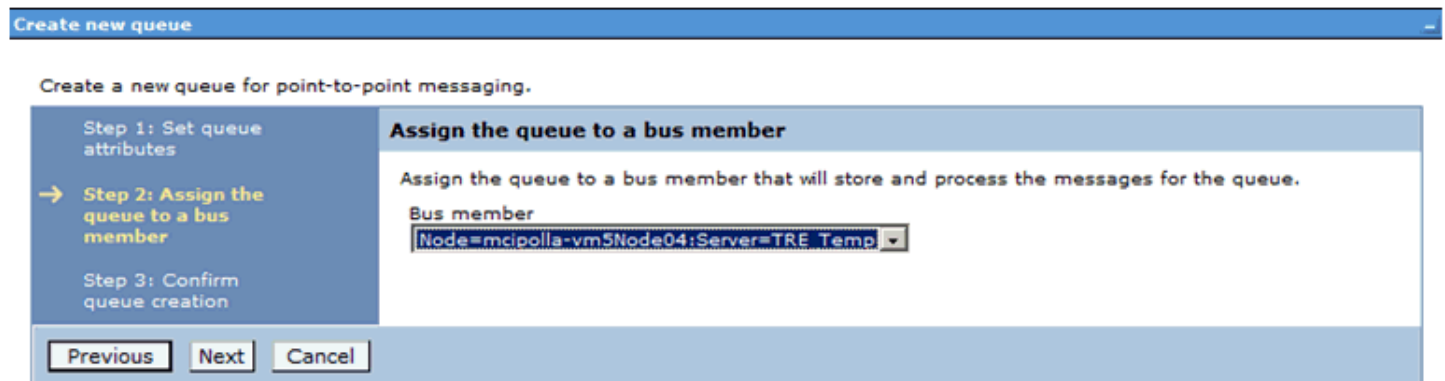
15. On Create New Queue - Step 2: Assign the queue to a bus member, use the pulldown to select The Transaction Server container as the Bus Member.

16. Click the **Next** button.

ORACLE

17. After creating the bus destination, the newly created bus destination is set as the queue name for the configured Queue. The newly created bus destination can be verified by using this navigation:

Service Integration > Buses > <BUS_NAME> Destinations

> **Note:** If any available unused bus destination is available in the Queue list, then it can be selected from the dropdown rather than creating a new bus destination.

> **Note:** The values specified in the table below for **Name** and **JDNI name** are samples only. Any unique value can be given. All other values should remain the same as the defaults.

Field	Value
Name	TestQueue
JNDI Name	jms/TestQueue
Queue name	Create Service Integration Bus destination (TestQueue)
Persistence	Persistent

18. Click the **OK** button to save the settings and view the newly created queue.

Creating JMS Queues in WebSphere Using a Non-Transaction Server Container

To create JMS queues in WebSphere Application Server, perform these tasks:

- *Create the Service Integration Bus (SIB)*
- *Create the JMS Queue Connection Factory*
- *Create the SIB JMS Destinations*

Create the Service Integration Bus (SIB)

To create the Service Integration Bus for the JMS queues:

1. Navigate to Service Integration > Buses.

ORACLE

2. Click the **New** button.

Create a new Service Integration Bus

Create a new Service Integration Bus.

→ **Step 1: Create a new bus**

(The next step of the wizard depends on decisions made in the current step)

Step 2: Confirm create of new bus

Create a new bus

Configure the attributes of your new bus.

✱ Enter the name for your new bus.

ORABus

☐ Bus security

[Next] [Cancel]

3. On Create a new bus, enter this value in the **Name** field:

   ORABus

ORACLE

4. Click the **Next** button.

Create a new Service Integration Bus

Create a new Service Integration Bus.

Step 1: Create a new bus	**Confirm create of new bus**
→ **Step 2: Confirm create of new bus**	The following is a summary of your selections. To complete the bus creation, click Finish. If there are settings you wish to change, click Previous to review bus settings.

Summary of actions:

New bus "ORABus" will be created with bus security setting "Disabled".

[Previous] [Finish] [Cancel]

ORACLE

5. On Confirm create of new bus, review your changes and click the **Finish** button.

Buses

Buses

A service integration bus supports applications using message-based and service-oriented architectures. A bus is a group of interconnected servers and clusters that have been added as members of the bus. Applications connect to a bus at one of the messaging engines associated with its bus members.

⊞ Preferences

| New | Delete |

Select	Name ↕	Description ↕	Security ↕
You can administer the following resources:			
☐	ORABus		Disabled
Total 1			

6. On Buses, click the link for the service integration bus that you just created (in this example, the link is **ORABus**.
7. Under **Topology**, click the **Bus Members** link.

ORACLE

8. Click the **Add** button.

9. On Add a new bus member, select the **Server** radio button.
10. Select the default server value **server1** for the server.
11. Click the **Next** button.
12. Select **Data Store**.
13. Click the **Next** button.
14. Select **Create default data source with generated JNDI name**.
15. Click the **Next** button.
16. Select the **Change heap sizes** radio button to modify the heap sizes to the proposed values.

    On machines with low amounts of physical memory size or large numbers of application server instances, it may be necessary to reduce the proposed values accordingly.
17. Review your changes and click the **Finish** button.
18. Click the **Save** link at the top of the page to save the changes.

Create the JMS Queue Connection Factory

To create the JMS Queue Connection Factory:

1. Navigate to **Resources > JMS** and select *JMS Providers*.
2. Click the **Default Messaging Provider** link which includes **Server1** in the scope.
3. Click **Connection Factories**.

ORACLE

4. Click the **New** button.

ORACLE

JMS providers > Default messaging provider > Queue connection factories > New

A JMS queue connection factory is used to create connections to the associated JMS provider of JMS queues, for point-to-point messaging. Use queue connection factory administrative objects to manage JMS queue connection factories for the default messaging provider.

Configuration

General Properties

Administration

Scope
Node=dndedasvm4Node02

Provider
Default messaging provider

✴ Name
TestQueueConnectionFactory

✴ JNDI name
jms/TestQueueConnectionFac

Description

Category

Connection

✴ Bus name
ORABus

Target

Target type
Messaging engine name

Target significance
Preferred

The additional properties will not be available until the general properties for this item are applied or saved.

Additional Properties

  ■ Connection pool properties

Related Items

  ■ JAAS - J2C authentication data

  ■ Buses

ORACLE

Target inbound transport chain

Provider endpoints

dndedasvm4:7276:BootstrapBasicMessaging

Connection proximity

Bus

Quality of Service

Nonpersistent message reliability

Express nonpersistent

Persistent message reliability

Reliable persistent

Advanced Messaging

Read ahead

Default

Temporary queue name prefix

Pass message payload by reference

☐ Applications using this Connection Factory to send messages:
- do not modify the data Object contained in a JMS Object Message
- populate a JMS Bytes Message using a single call to writeBytes(byte[]) and do not modify the byte array once it is contained in the message. Read the help before selecting this option.

☐ Applications using this Connection Factory to receive messages:
- do not modify the data Object obtained from a JMS Object Message. The data Object is treated as read only. Read the help before selecting this option.

ORACLE

Advanced Administrative

☐ Log missing transaction contexts

☐ Manage cached handles

☐ Share data source with CMP

Security settings
Select the authentication values for this resource.
Authentication alias for XA recovery
(none) ▾
Mapping-configuration alias
DefaultPrincipalMapping ▾
      Container-managed authentication alias
(none) ▾

5. On Queue connection factories, New, complete these fields:

Field	Value
Name	ESBQueueConnectionFactory
JNDI name	jms/ESBQueueConnectionFactory
Bus name	ESBBus
Target Type	Messaging Engine Name
Provider end points	<System-name>:7276:BootstrapBasicMessaging
Connection Proximity	Bus

Note: Leave the default value for the rest of the parameters.

6. Click the **OK** button.
7. Click the **Save** link at the top of the page to save the changes.

ORACLE

Create the SIB JMS Destinations

To create the JMS queue destinations:

1. Navigate to Service Integration > Buses.
2. Click the **ORABus** link.
3. Under **Destination Resources**, click **Destinations**.
4. Click the **New** button.

5. On Step 1: Set queue attributes, in the Identifier field, enter **TestQueue**.

ORACLE

6. Click the **Next** button.

Create new queue

Create a new queue for point-to-point messaging.

Step 1: Set queue attributes	**Assign the queue to a bus member**
→ Step 2: Assign the queue to a bus member	Assign the queue to a bus member that will store and process the messages for the queue.
Step 3: Confirm queue creation	Bus member
	Node=dndedasvm4Node02:Server=server1

[Previous] [Next] [Cancel]
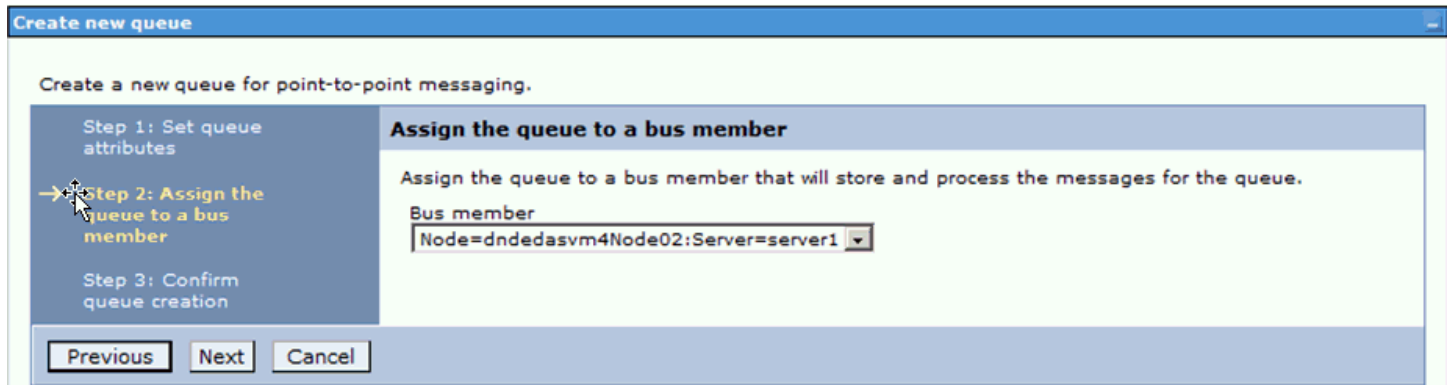
7. On Step 2: Assign queue to a bus member, use the pulldown to assign the queue to a bus member.
8. Click the **Next** button.

Create new queue

Create a new queue for point-to-point messaging.

Step 1: Set queue attributes	**Confirm queue creation**
Step 2: Assign the queue to a bus member	To complete creation of the queue, click Finish. If you want to change any selections, click Previous.
→ Step 3: Confirm queue creation	Summary of actions:
	New queue "TestQueue" will be created.
	A Queue point for "TestQueue" will be created for bus member "dndedasvm4Node02:server1" of bus "ORABus".

[Previous] [Finish] [Cancel]

9. On Step 3: Confirm queue creation, click the **Finish** button.

10. Click the **Save** link at the top of the page to save the changes.

5 Adding a Subscriber

Setting Processing Options for Event Subscribers Application (P90702A)

This section discusses how to set up the default values for the processing option.

To set Processing Options for P90702A:

1. Type IV in the *Fast Path*.
2. Type P90702A in the *Interactive Application* and click *Find*.

ORACLE
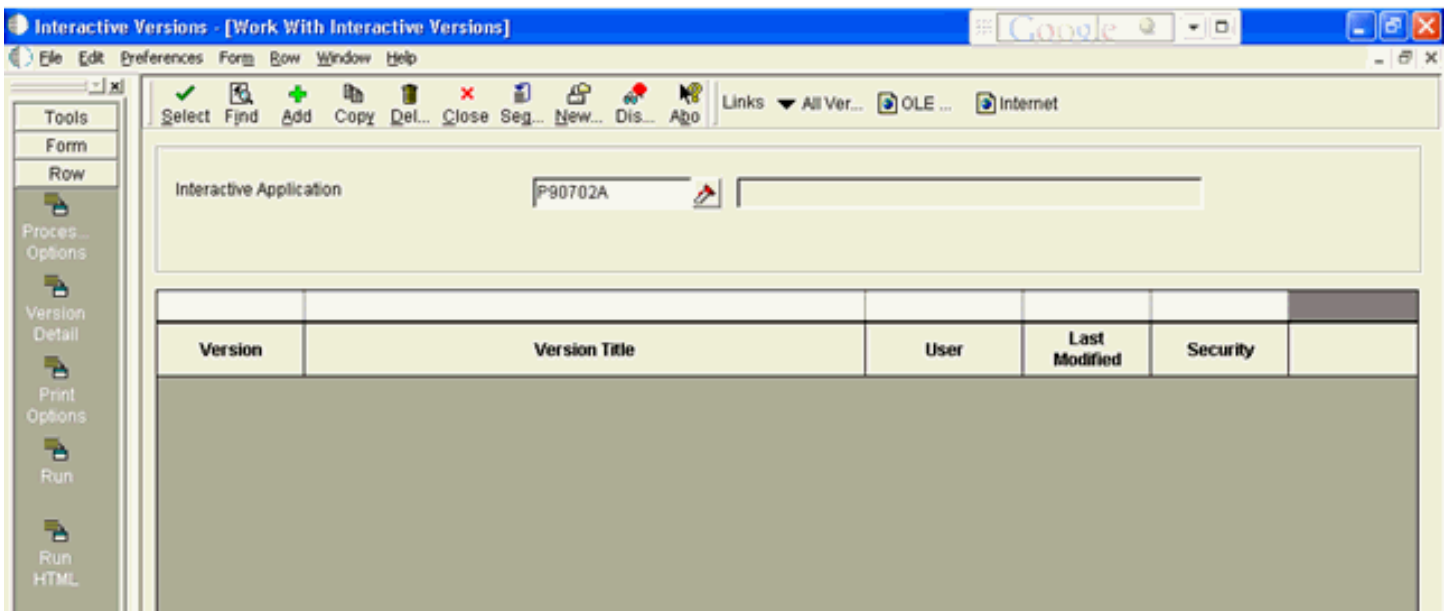
3. Select the row that appears in the grid and then click *Processing Options* from *Row* menu.

ORACLE
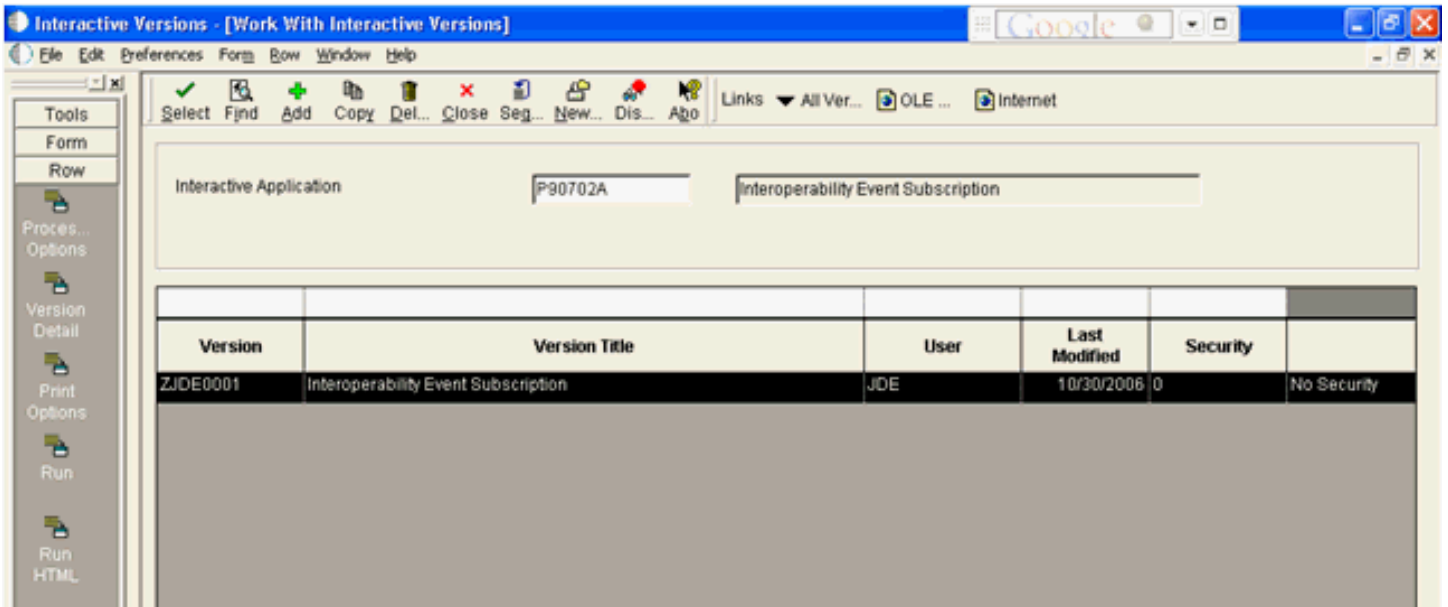
4. On *Processing Options,* edit the default values and click *OK.*

> **Note:** The values for these processing options are set by using an EnterpriseOne application as described in the next section of this chapter in the section entitled: *Using Event Subscribers Application to Add a Subscriber*.

Using Event Subscribers Application to Add a Subscriber

This section discusses how to add a subscriber to EnterpriseOne by executing the application P90702A.

ORACLE

To use P90702A to add a JMSQUEUE or JMSTOPIC subscriber:

1. Run the application P90702A from JAS.
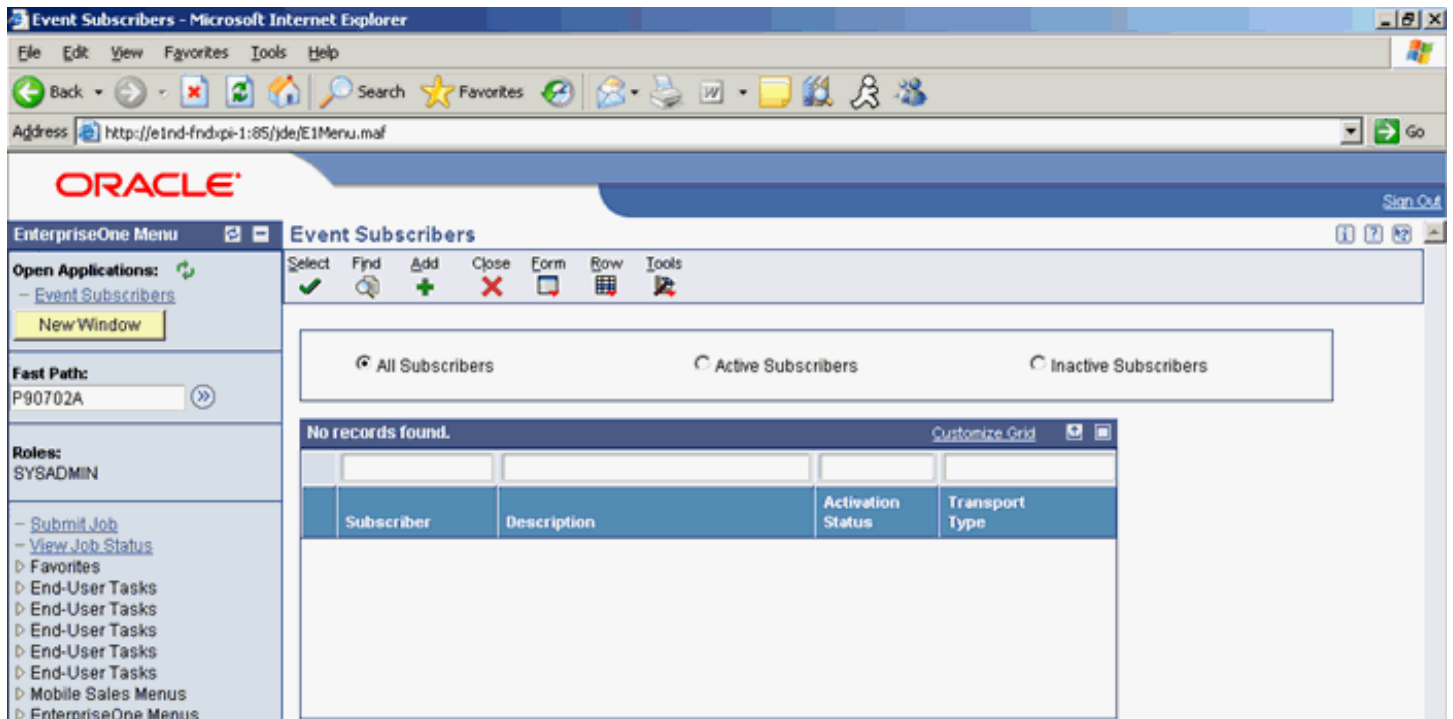
2. Click the *Add* button. Enter the values as per any one of the following options:

The Initial Context Factory and Provider URL values are populated based on the selections for Application Server and Queue Location. Edit these values according to the environment. A processing option determines the default values.

The values for **Connection Factory JNDI Name** and **Queue Name** are the JNDI name for the subscriber Queue or Topic that is being created.

Working with Subscribers in JD Edwards EnterpriseOne
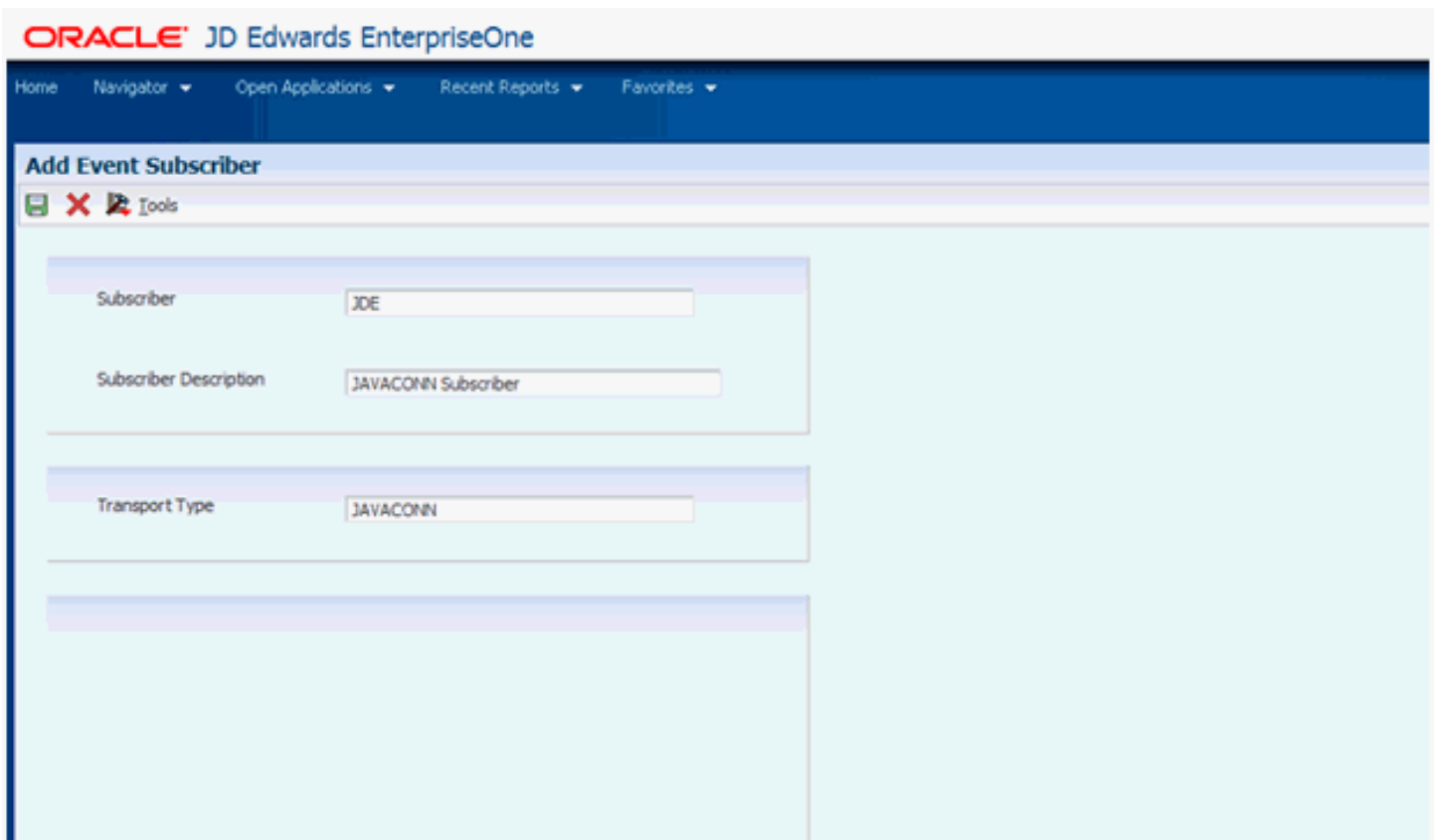
This section describes these tasks:

- *Adding a JAVACONN Subscriber*
- *Creating a JMSQUEUE Subscriber*
- *Creating a JMSTOPIC Subscriber*
- *Creating a Subscription*
- *Adding an Event to the Subscription*
- *Adding Environment(s) to a Subscription*

Adding a JAVACONN Subscriber

Use this procedure to add a JAVACONN subscriber in JD Edwards EnterpriseOne:

1. From a JD Edwards EnterpriseOne client, run P90702A to use the Add Event Subscriber application to add a subscriber.

ORACLE

Field	Description
Subscriber	The *Subscriber* field must have a value matching valid user in JD Edwards EnterpriseOne.
Subscriber Description	Enter this value: **JAVACONN Subscriber**
Transport Type	Enter this value:. **JAVACONN**

2. Click the **OK** button to accept the values.
3. Run the P90702A application and find the JAVACONN Subscriber created in the above step.
4. Activate the JAVACONN Subscriber by taking this Form Exit:

   Row > Change Status
5. Change the status to AV (Active).

Creating a JMSQUEUE Subscriber

To create a JMSQUEUE subscriber:

1. From a JD Edwards EnterpriseOne client, run P90702A to use the Add Event Subscriber application to add a subscriber.

ORACLE JD Edwards EnterpriseOne

Home   Navigator ▼   Open Applications ▼   Recent Reports ▼   Favorites ▼

Add Event Subscriber

💾 ✖ 🖎 Tools

Subscriber	JDE
Subscriber Description	JMSQUEUE Subscriber
Transport Type	JMSQUEUE
Connection Factory JNDI Name	jms/TestConnectionFactory
Queue Name	jms/TestQueue
Message Format	XML XML Format
Application Server	WEBLOGIC Weblogic App Server
Queue Location	LOCAL Local
Initial Context Factory	weblogic.jndi.WLInitialContextFactory
Provider URL	t3://dndedasvm4.us.oracle.com:7002

ORACLE

Field	Description
Subscriber	The *Subscriber* field must have a value matching valid user in JD Edwards EnterpriseOne.
Subscriber Description	Enter this value: **JMSQUEUE Subscriber**
Transport Type	Enter this value:. **JMSQUEUE**
Connection Factory JNDI	You can specify any path and name. **Note:** You must use this same value when creating the Connection Factory in WebLogic as described later in this guide. For purposes of this documentation, the value is: `jms/TestConnectionFactory`
Queue Location	You can choose either LOCAL or REMOTE. **Note:** If the subscriber queue is created on the Transaction Server instance you should specify the value as LOCAL. If the subscriber queue is on remote WebLogic container, specify the value as REMOTE
Initial Context Factory	Enter this value: `weblogic.jndi.WLInitialContextFactory`
Provider URL	Enter this value: `t3://<WLS_SERVER_HOST>:<WLS_SERVER_PORT>` **Note:** `< WLS_SERVER_HOST >` is the host of the machine where the subscriber queue is created and `<WLS_SERVER_PORT>` is the port at which the server is hosted.

2. Click the **OK** button to accept the values.
3. Run the P90702A application and find the JMSQUEUE Subscriber created in the above step.
4. Activate the JMSQUEUE Subscriber by taking this Form Exit:

   Row > Change Status
5. Change the status to AV (Active).

Creating a JMSTOPIC Subscriber

To createa a JMSTOPIC subscriber:

1. From a JD Edwards EnterpriseOne client, run P90702A to use the Add Event Subscriber application to add a subscriber.

ORACLE JD Edwards EnterpriseOne

Home   Navigator ▼   Open Applications ▼   Recent Reports ▼   Favorites ▼

Add Event Subscriber

💾 ✖ 📝 Tools

| Subscriber | JDE |
| Subscriber Description | JMSTOPIC Subscriber |

| Transport Type | JMSTOPIC |

| Connection Factory JNDI Name | jms/TestConnectionFactory |
| JMS Topic | jms/TestTopic 🔍 |

Field	Description
Subscriber	The *Subscriber* field must have a value matching valid user in JD Edwards EnterpriseOne.

ORACLE

Field	Description
Subscriber Description	Enter this value: **JMSTOPIC Subscriber**
Transport Type	Enter this value:. **JMSTOPIC**
Connection Factory JNDI	You can specify any path and name. **Note:** You must use this same value when creating the Connection Factory in WebLogic as described later in this guide. For purposes of this documentation, the value is: `jms/TestConnectionFactory`

2. Click the **OK** button to accept the values.
3. Run the P90702A application and find the JMSTOPIC Subscriber created in the above step.
4. Activate the JMSTOPIC Subscriber by taking this Form Exit:

   Row > Change Status
5. Change the status to AV (Active).

Creating a Subscription

Use this procedure to create a Subscription.

1. From a JD Edwards EnterpriseOne client, run the P90702A application and find the Subscribers you created in the previous steps of this section.
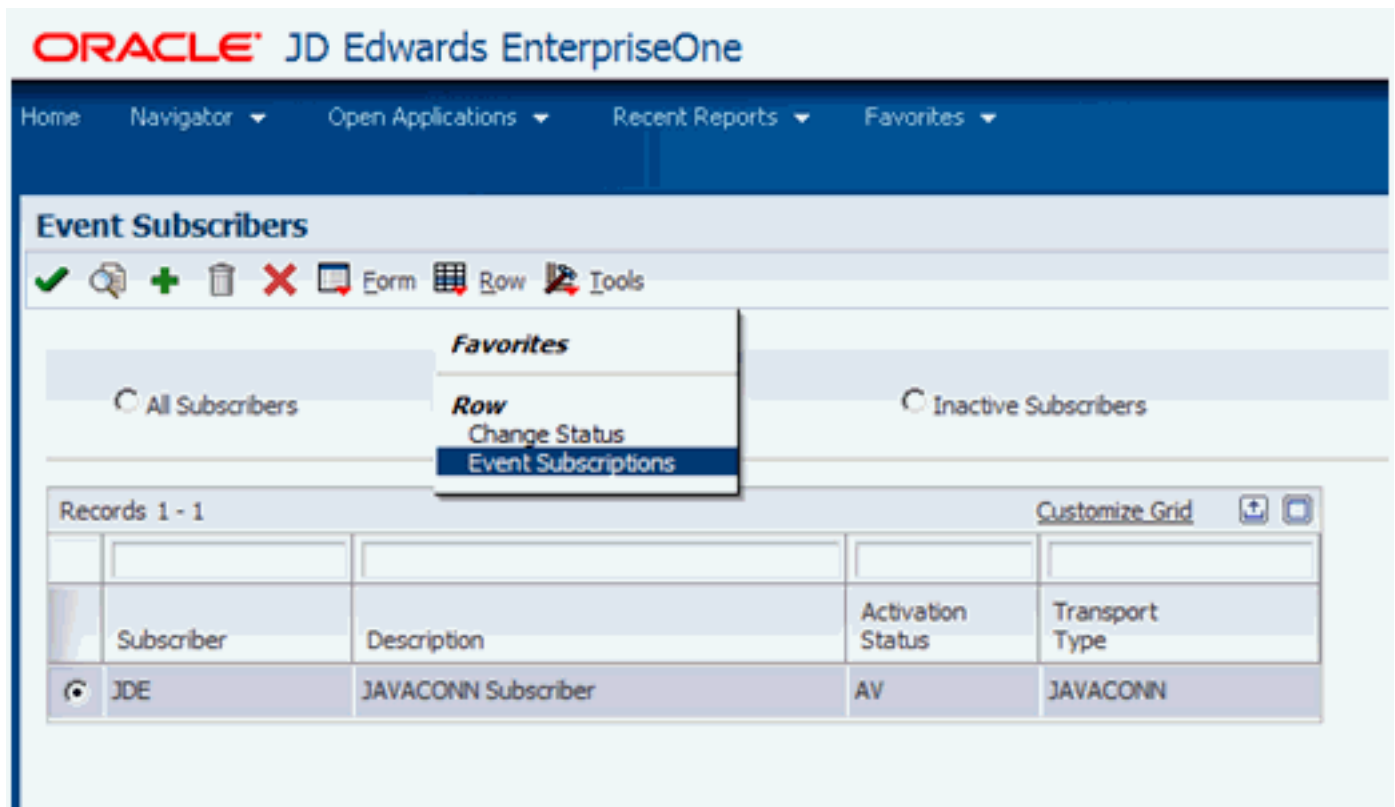
2. On Event Subscribers, take the Row Exit for Event Subscriptions.

ORACLE
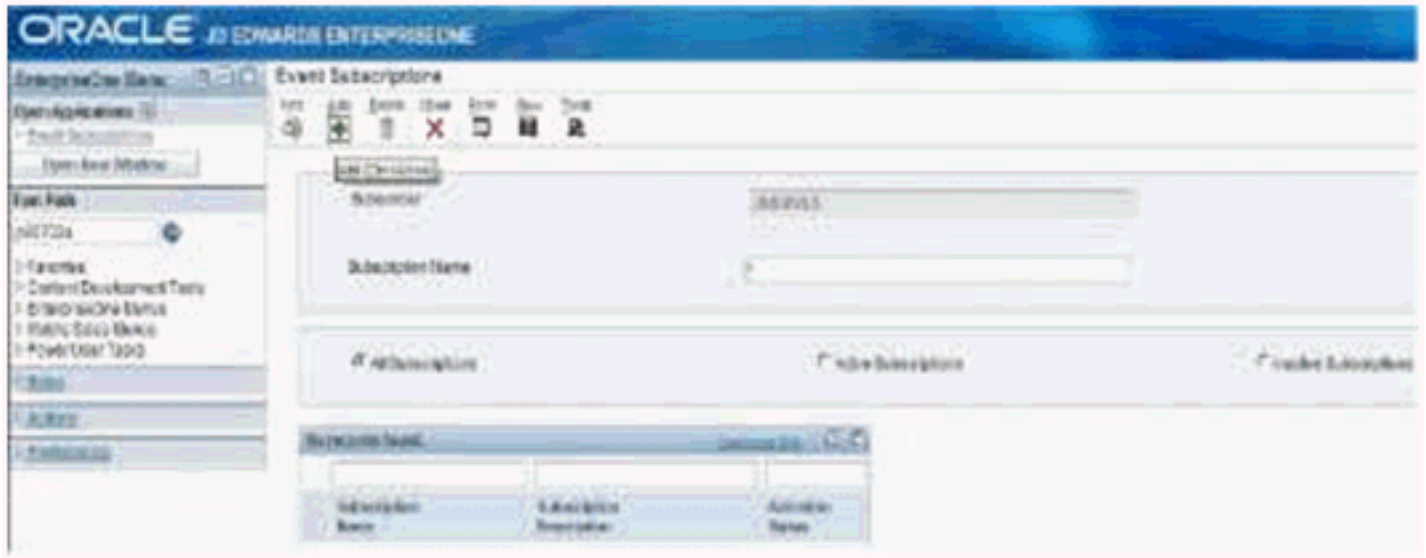
3. On Event Subscriptions, click the **Add** button.

4. On Add Event Subscription, complete the *Subscriber Name* and *Subscriber Description* fields with any value. In this example, the both values are **JDE**.
5. Click the **OK** button to accept the values.
6. Activate the Subscriber by taking this Form Exit:

   Row > Change Status
7. Change the status to AV (Active).

Adding an Event to the Subscription

Use this procedure to add an Event to the Subscription.

> **Note:** In this example, the added event is named RTABOUT.

ORACLE

1. Run P90702A and find the Subscriber created in the previous step of this section.

   Note: In this documentation, that subscription was called **JDE**.

ORACLE

2. On Event Subscriptions, select the *Subscribed Events* Row Exit.

3. On Subscribed Events, add *RTABOUT* as an *Event Type* by clicking the radio button next to the value.
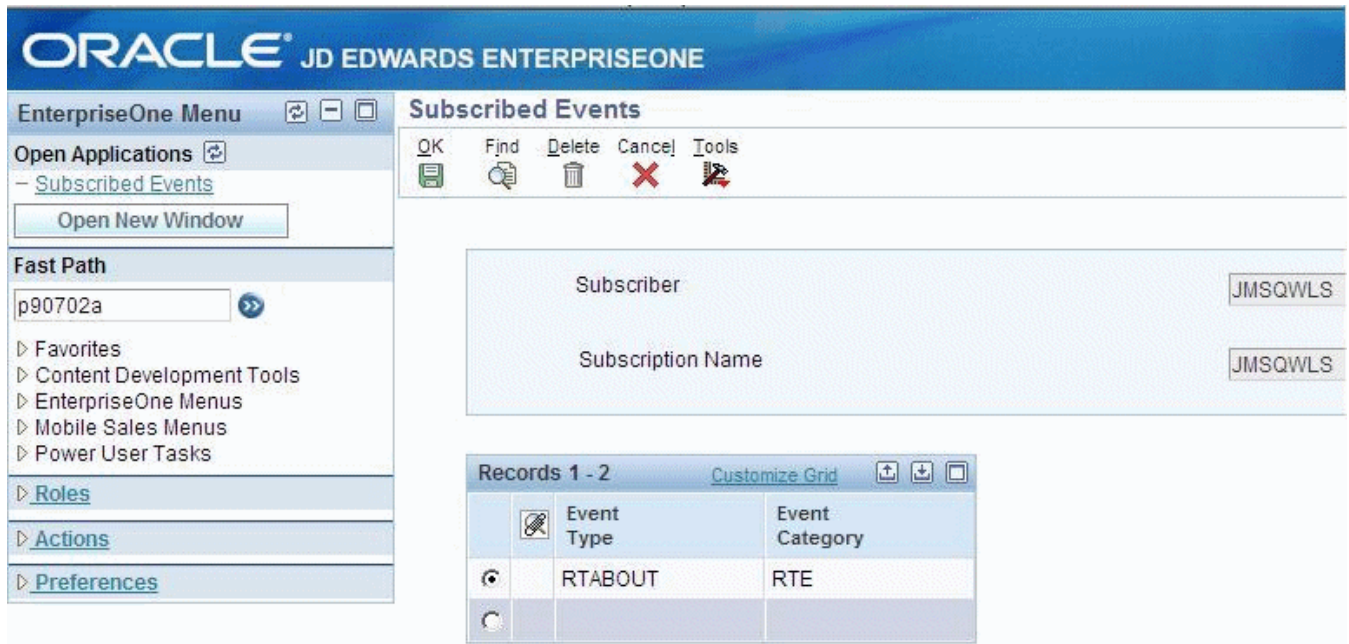
4. Click the *OK* button to add *RTABOUT* as a Subscribed Event to the Subscription.

Adding Environment(s) to a Subscription

Use this procedure to add an Environment to a Subscription.

1. Run P90702A and find the Subscriber you created earlier in this section of the guide.

2. On Event Subscriptions, select the *Subscribed Env* Row Exit.
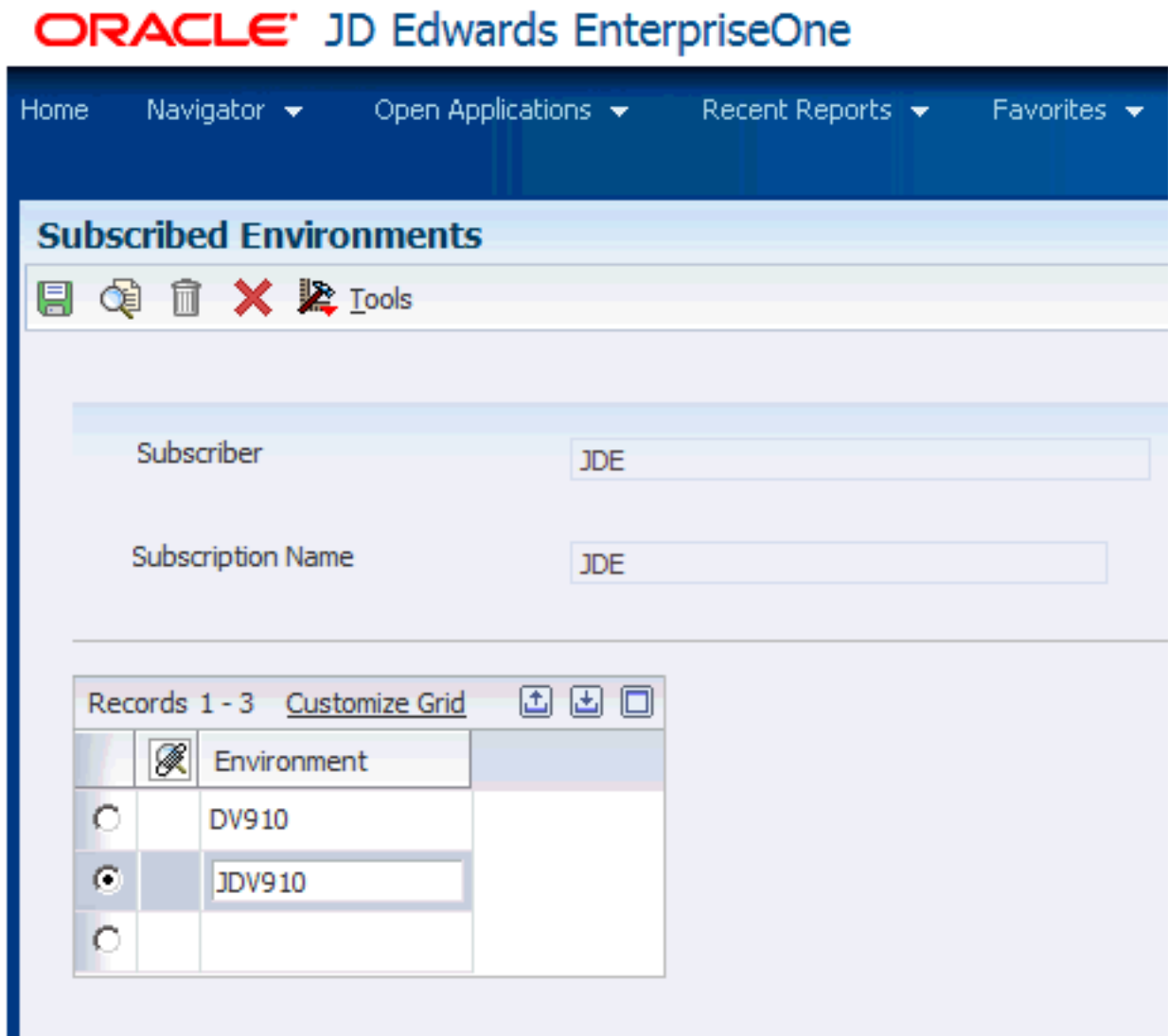
3. On Subscribed Environments, click the radio buttons to add Environments as required.
4. Click the *OK* button to add the selected *Environments*. to the Subscription.

ORACLE

6 Troubleshooting Real Time Events Processing

This chapter discusses troubleshooting Real Time Events processing.

Problem: There are some events in the table F90710 which are not getting picked up by the RTE server.

Resolution: Check if the event sequence number stored in the table F90708 is greater than lowest event sequence number available in table F90710. In this case the event sequence number in table F90708 must be reset to 0 manually

Problem: Some of the clustered servers are not processing events.

Resolution: Probably some servers in the cluster are having some problem. Refer the app server log to identify the problem, rectify the problem and restart the cluster.

Resolution: There could be very high number of events processed exceeding the filestore size. It is advisable to consume events while they are being produced and processed. This keeps the filestore clean.

Problem: Getting error in the RTE Server on WAS (Failed to send committed event message: CWSIA0053E: An exception was received during the call to the method JmsSessionImpl.getTransaction (#1): javax.resource.spi.IllegalStateException: CWSJR1121E: An internal error has occurred. During the call to the method getManagedConnection the exception javax.resource.spi.ResourceAllocationException: CWSJR1028E: An internal error has occurred. The exception com.ibm.ws.sib.processor.exceptions.SIMPConnectionUnavailableException)

Resolution: One of the messaging engines is down and unable to process event. This should not be a concern since the failover service is active and event will be processed by other available messaging engines, assuming they are available.

Problem: Error message "JNDI lookup of Queue failed" seen in RTE server log.

Resolution: It is likely that the JNDI resources are not loaded yet. This error most commonly occurs when auto migration is enabled. A retry lookup feature exists which will try to lookup until successful. If the problem persists, then check the application server log

Problem: All the messages are being sent to a single queue in a cluster or not evenly distributed.

Resolution: A failover may have occurred and the Queue configuration has been changed due to failover. Stop the clustered servers, modify the configuration as stated in this document and start the cluster to process events.

ORACLE

JD Edwards EnterpriseOne
Transaction Server Reference Guide for Microsoft Windows
using WebSphere

Chapter 7
Upgrading the Transaction Server RTE Software
Component Using Server Manager

7 Upgrading the Transaction Server RTE Software Component Using Server Manager

Use this procedure to upgrade the Transaction Server RTE component that was installed using Server Manager.

1. Obtain the RTE software component to which you wish to upgrade from Oracle eDelivery.
2. Place the new RTE component into this folder:

   `<AGENT_LOC>\components`
3. Log on to the Server Manager console.
4. Click on the RTE instance.

5. In the Software Component Version section, click the **Change** button.
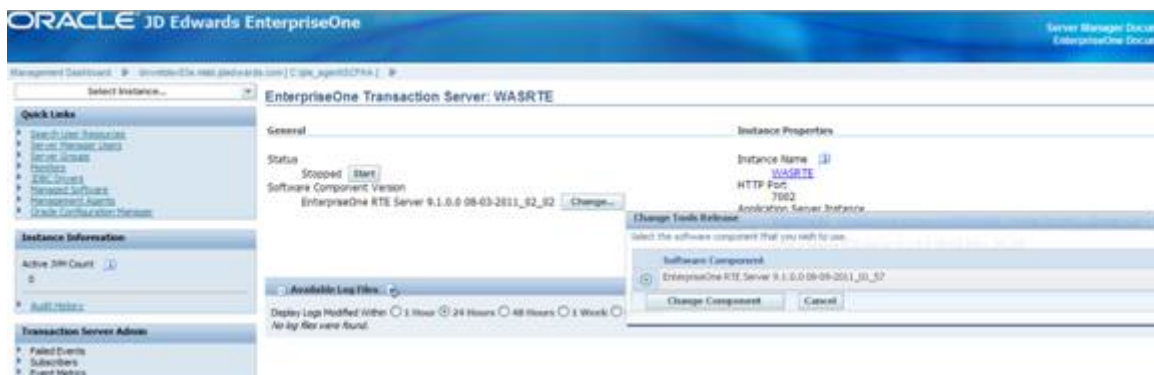
6. On the Change Tools Release layout, select the desired version to install from the component by selecting the radio button.

ORACLE

JD Edwards EnterpriseOne
Transaction Server Reference Guide for Microsoft Windows
using WebSphere

Chapter 7
Upgrading the Transaction Server RTE Software
Component Using Server Manager

7. Click the **Change Component** button.

> **Note:** An upgrade of RTE server only upgrades the Transaction Server RTE component of JD Edwards, and not the JMS components.

Refer to the *JD Edwards EnterpriseOne Tools Server Manager Guide* to upgrade the Server Manager Agent. You should only update the Server Manager Agent after you have upgraded the Server Manager Console to the latest release, which can be independent of a JD Edwards Tools Release.

JD Edwards EnterpriseOne
Transaction Server Reference Guide for Microsoft Windows
using WebSphere

Chapter 8
Uninstalling the Transaction Server RTE Software
Component using Server Manager

8 Uninstalling the Transaction Server RTE Software Component using Server Manager

Use this procedure to uninstall the Transaction Server RTE Software Component using Server Manager.

1. Log on to the Server Manager console.
2. Click on the managed home location.
3. Select the Transaction Server Real Time Events (RTE) instance.
4. Click the **Remove Instance** button.
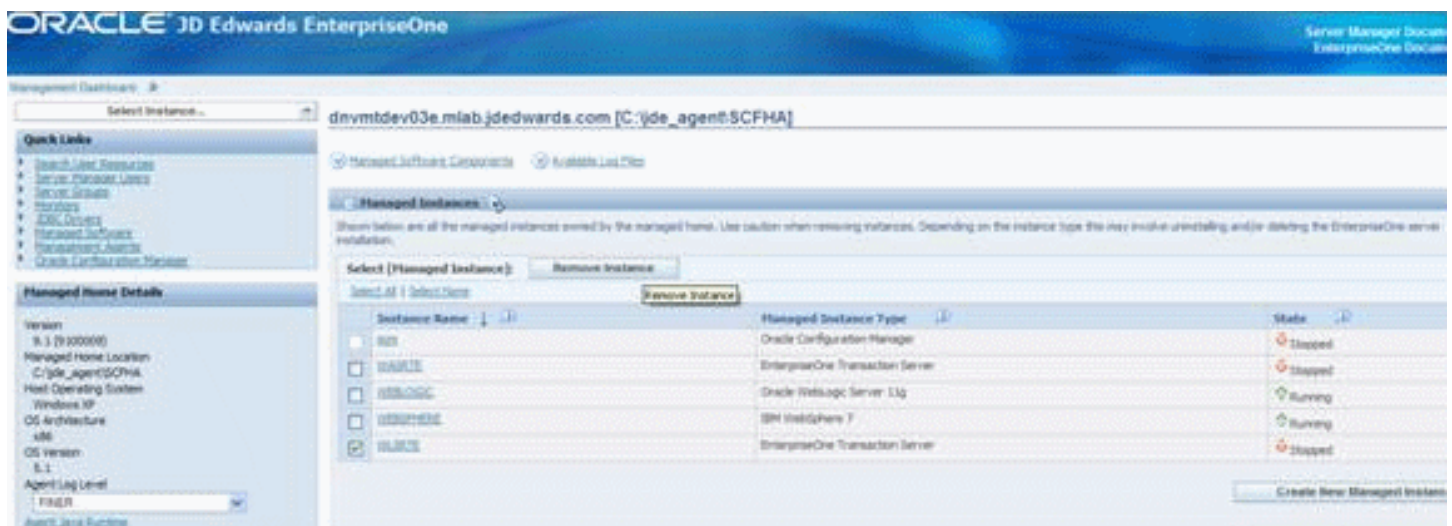
> **Note: Post Uninstallation.** If after you uninstall the Transaction Server RTE instance, this folder is present, you must manually delete it: `<WAS_ PROFILE_LOC>\filestores\ORABus-<RTE_INSTANCE_NAME>`

ORACLE

JD Edwards EnterpriseOne
Transaction Server Reference Guide for Microsoft Windows
using WebSphere

Chapter 8
Uninstalling the Transaction Server RTE Software
Component using Server Manager

JD Edwards EnterpriseOne
Transaction Server Reference Guide for Microsoft Windows   Appendix A - Architecture of the Guaranteed Event System
using WebSphere

Chapter 9

9 Appendix A - Architecture of the Guaranteed Event System

This appendix presents an architectural diagram of the Guaranteed Event System, which is one of the *JD Edwards EnterpriseOne* Transaction Server Components. The diagram depicts the general sequence that occurs when an event is published. These steps correspond to the numbered sequence of events in the diagram:

1. An HTML client user executes a business function (BSFN) request.

   This request is sent to the *JD Edwards EnterpriseOne HTML Server*.
2. The *HTML Server* forwards this request to a CallObject kernel on the Enterprise Server.
3. The CallObject kernel executes the BSFN, which calls the Event API to send the event data to the F90710 Event Transfer table.

   If the event is a Z event, the data is sent to F90710 in a final XML format.
4. A database polling application available with the JD Edwards EnterpriseOne Transaction Server polls any available records in F90710.
5. The transaction server retrieves the event data from F90710 and (for Real-Time and XAPI events) converts the data to an XML document in the appropriate format.
6. The transaction server routes the event to the subscriber queue(s) for each subscriber that has established an active subscription for that event.
7. When a subscriber connects to the Transaction Server, the subscriber is sent all the events that exist in its subscription queue at that time.
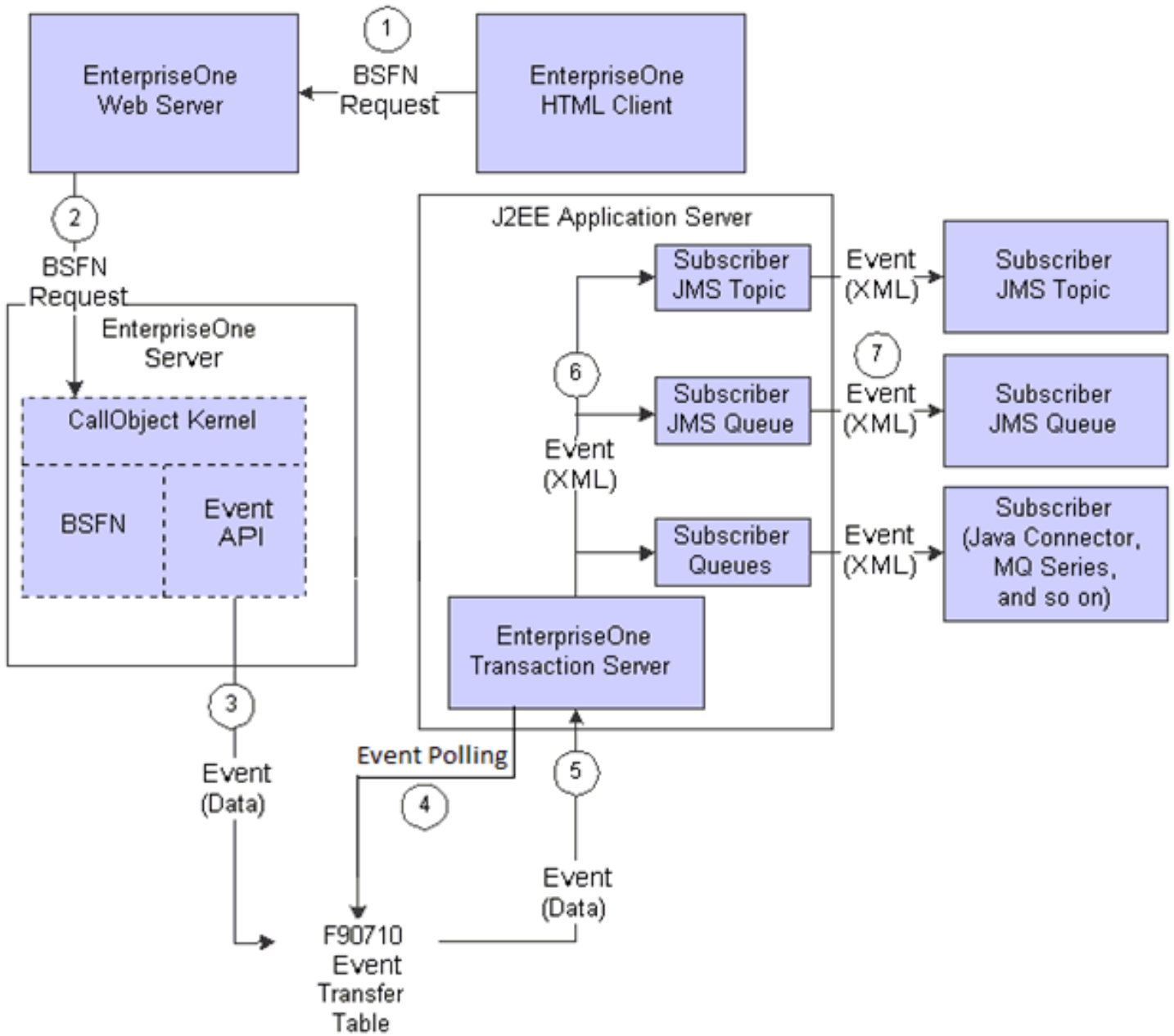
JD Edwards EnterpriseOne
Transaction Server Reference Guide for Microsoft Windows
using WebSphere

Chapter 10
Appendix B - Creating and Configuring Multiple Transaction
Servers per Environment or User/Role

10 Appendix B - Creating and Configuring Multiple Transaction Servers per Environment or User/Role

The default configuration for the Transaction Server defines the F907* tables (event tables) to reside in one datasource. These tables are included in the System datasource database (for example, SY920). Using this default configuration, only a single Transaction Server may be configured to monitor the F907* tables.

This appendix provides instructions to set up and configure multiple Transaction Servers so that transactions can be separated by one or more of these criteria:

- Environment
- Role/User

This process allows you to:

- Separate transaction events by environment so that two different transaction servers can service the events
- Use OCM mappings to point to the different environments

This appendix describes these tasks:

- *Creating and Configuring Multiple Transaction Servers per Environment*
- *Creating and Configuring Multiple Transaction Servers per Role/User*

Creating and Configuring Multiple Transaction Servers per Environment

These general guidelines apply:

- *Production Environments*

  The F907* tables should remain in the System database as defined by the standard setup and configuration. Since the transaction server is designed to work with only one set of F907* tables, the normal procedure is to configure the transaction server with these system tables.

- *Test and Dev Environments*

  In order to separate Test and Dev environment transactions from Production, you can copy the F907* tables into the business data database and change the OCM mappings accordingly. A second transaction server is necessary in order to access these copied tables because, by design, each Transaction Server can only work with only one set of F907* tables.

- *Multiple Transaction Servers*

  In order to support the separation of the F907* tables per environment, you must install an additional Transaction Server per environment or environment combination. For example, if you want to isolate your

JD Edwards EnterpriseOne
Transaction Server Reference Guide for Microsoft Windows
using WebSphere

Chapter 10
Appendix B - Creating and Configuring Multiple Transaction
Servers per Environment or User/Role

Production environment and then combine your Test and Dev environments, you would need a total of two Transaction Servers. If you further wanted to separate your Test and Dev environments, you would need a total of three Transaction Servers.

Do not start any added Transaction Servers until you complete the procedure in this appendix.

To create and configure the Transaction Server per environment:

1. With the exception of the tables listed in the *Exceptions* below, copy all remaining F907* tables from the System database to the business data database.

   Exceptions

   The following tables are initially cached on JD Edwards EnterpriseOne service startup and contain the necessary information needed for all environments and subscribers. Therefore, these tables and their OCM mappings should remain in the System datasource:

   ○ F90701

   ○ F907011

   ○ F90705

2. Use OCM to remap the F907* tables copied in Step *1* to the new business data database for both the system and the server map.

   For example, the completed mappings might look like these:

   ```
   JPY920 F90701  TBLE  System - 920 *PUBLIC AV
   JPY920 F907011 TBLE  System - 920 *PUBLIC AV
   JPY920 F907012 TBLE  System - 920 *PUBLIC AV
   JPY920 F90703  TBLE  Business Data - CRP *PUBLIC AV
   JPY920 F90704  TBLE  Business Data - CRP *PUBLIC AV
   JPY920 F90705  TBLE  System - 920 *PUBLIC AV
   JPY920 F90706  TBLE  Business Data - CRP *PUBLIC AV
   JPY920 F90707  TBLE  Business Data - CRP *PUBLIC AV
   JPY920 F907071 TBLE  Business Data - CRP *PUBLIC AV
   JPY920 F907072 TBLE  Business Data - CRP *PUBLIC AV
   JPY920 F90708  TBLE  Business Data - CRP *PUBLIC AV
   JPY920 F90710  TBLE  Business Data - CRP *PUBLIC AV
   JPY920 F90711  TBLE  Business Data - CRP *PUBLIC AV
   JPY920 F90712  TBLE  Business Data - CRP *PUBLIC AV
   JPY920 F90715  TBLE  Business Data - CRP *PUBLIC AV
   JPY920 F90720  TBLE  Business Data - CRP *PUBLIC AV
   JPY920 F90730  TBLE  Business Data - CRP *PUBLIC AV
   ```

3. Activate the RT* event for added environment (for example, JPY920) in P90701 by:

   ○ Taking the Form exit to Event Activation

   ○ Doing an Add to add the new Event per environment

   ○ Using the Row exit to change the status to Active

4. Refresh the JD Edwards EnterpriseOne server cache by restarting the EnterpriseOne services.
5. Add subscriber information for the added environment (for example, JPY920).
6. Refresh the Transaction Server cache by restarting the transaction server.
7. Trigger the event.
8. Verify the event is recorded in the F90710 Business Data Source table.
9. From a JD Edwards EnterpriseOne client machine, use this command to run the Table Trigger script for F90710 in the business data for the added environment (for example, PY Business Data):

ORACLE

```
<home>\system\bin32\dbtemplates.exe -create
```

where `<home>` is the installation directory of the JD Edwards EnterpriseOne Administration client that is named according to the JD Edwards EnterpriseOne release. For example, E812, E900, E910, or E920.

10. Verify the F986112 table is populated and the new event in F90710 has the status 3 instead of 2.

Creating and Configuring Multiple Transaction Servers per Role/User

This section describes a configuration where multiple Real Time Events can be processed by multiple Transaction Server instances within the same JD Edwards EnterpriseOne environment. You can process events by Role/User to enable specific users to have their events processed by a specific Transaction Server. This setup is similar to that in the previous section, *Creating and Configuring Multiple Transaction Servers per Environment*. Instead of separating the events by environment you are separating the Real Time Event messages by Role/User.

Multiple Transaction Servers

In order to support the separation of the F907* tables per Role/User, you must install an additional Transaction Server per User or Role combination. For example, you have set up a role for processing Sales Orders and another role for processing all other Real Time Events. In this case, you would need an additional Transaction Server for each role addition.

To configure the system to route Real Time Events to specific Transaction Servers based on the JD Edwards EnterpriseOne Role/User user that is generating the event:

1. From JD Edwards EnterpriseOne, create a new Role which will include a group of users that will initiate the Real Time Event messages.

2. Create these tables in the same environment datasource (Business Data - PROD) or create a custom Data Source (for example, TS2 / PD920):

   - F90710 - Event table
   - F90730 - Unique sequence table

3. Copy these tables from the System datasource to the same environment datasource you used in Step 2:

   - F90706 - Event Subscriber
   - F90707 - Event Subscription
   - F907071 - Subscribed Events
   - F907072 - Subscribed Environments
   - F90708 - Event Sequence

   These tables contain the subscriber information and whether they are active or not. It is important to note that when you copy these tables, whenever you want to make changes to the subscriber information you must ensure you are logged in as the appropriate user. This is required to ensure you are making changes to the correct table. If you do not copy these tables, you will continue to see errors in the Transaction Server log where events cannot be delivered to all the subscribers, although this does not affect the delivery of the events to the correct queue.

4. Create new OCM mappings for all the tables listed in Steps 2 and 3 for the new Role (for example, role associated with the PD920 environment and the copied F90710 table and the role associated with the PD920

ORACLE

JD Edwards EnterpriseOne
Transaction Server Reference Guide for Microsoft Windows
using WebSphere

Chapter 10
Appendix B - Creating and Configuring Multiple Transaction
Servers per Environment or User/Role

environment and the F90730 table) to point to Business Data - PROD. You must make mappings for both SYSTEM and Server Map.

5. On a JD Edwards EnterpriseOne client machine, run the program to create the table trigger on the new F90710 table. From a command prompt, navigate to this directory:

   ```
   e920\system\bin32\
   ```

6. From the above directory, run this command:

   ```
   dbtemplates.exe -create
   ```

   When prompted ensure you log in as the new user (assigned to your new role) so that the program uses the new OCM mappings to locate the correct table in which to create the new trigger.

   Henceforth when logging in as a user defined by this role, a generated Real Time Event sends the message into the appropriate F90710 table and updates the F90730 table.

7. Install a new Transaction server that will be used to process the events for each new Role/User.

8. Configure the newly installed Transaction Server with one of the users in the new Role as the bootstrap user.

9. Process a transaction that triggers an event for the new Role.

10. Start the newly installed Transaction Server and check the queue to make sure the event was delivered to the new Transaction Server.

11. Verify the previous configuration still functions as expected by logging in as any user *not* belonging to the new Role and generate an Event and see that the message goes into the F90710 (System - 920) and is delivered to the original Transaction Server.

JD Edwards EnterpriseOne
Transaction Server Reference Guide for Microsoft Windows
using WebSphere

Chapter 11
Appendix C - Testing and Validating the JMSQUEUE
Configuration

11 Appendix C - Testing and Validating the JMSQUEUE Configuration

If you have followed the steps to activate RTABOUT, a Real Time Event will be created whenever you Add, Change, or Delete an Address Book record. However, before performing any of those functions you should follow the procedure outlined in the following documents found in My Oracle Support. Access the document according to your Tools Release.

- E1: OUTBND: Creating and Using a JMSQUEUE in WebSphere for RTEs. Tools Release 8.98.x (Doc ID 1159783.1)

- E1: OUTBND: Creating and Using a JMSQUEUE in WebSphere for RTEs. Tools Release 9.1.x (Doc ID 1576889.1)

- E1: OUTBND: RTE Server (Transaction Server) Setup and Configuration for WebLogic - JMSQueue Subscriber (Doc ID 1137704.1)

ORACLE

JD Edwards EnterpriseOne
Transaction Server Reference Guide for Microsoft Windows
using WebSphere

Chapter 11
Appendix C - Testing and Validating the JMSQUEUE
Configuration

JD Edwards EnterpriseOne
Transaction Server Reference Guide for Microsoft Windows
using WebSphere

Chapter 12
Appendix D - Creating an Oracle JD Edwards EnterpriseOne
Service Request for Transaction Server Support

12 Appendix D - Creating an Oracle JD Edwards EnterpriseOne Service Request for Transaction Server Support

To minimize the resolution time for any Service Requests you might have to create with Global Support, it is recommended that you provide this information:

- JD Edwards EnterpriseOne Enterprise Server JDE.INI file

- Exact JD Edwards EnterpriseOne Tools Release on which you are running (for example, 9.2.1.1)

- Indicate that you are running Transaction Server on WebSphere Oracle WebLogic Server and the exact release and version your application server.

- The content of the `config` folder within the Transaction Server Agent where the Transaction Server instance resides

ORACLE

JD Edwards EnterpriseOne
Transaction Server Reference Guide for Microsoft Windows
using WebSphere

Chapter 12
Appendix D - Creating an Oracle JD Edwards EnterpriseOne
Service Request for Transaction Server Support

- Logs from Server Manager generated when starting the Transaction Server

> **Tip:** To obtain a more detailed set of logs, you can use Server Manager to enable enhanced logging. For the *Log Name* with a value *RTE*, change the *Log Components* value to *EVENTPROCESSOR* as shown below.
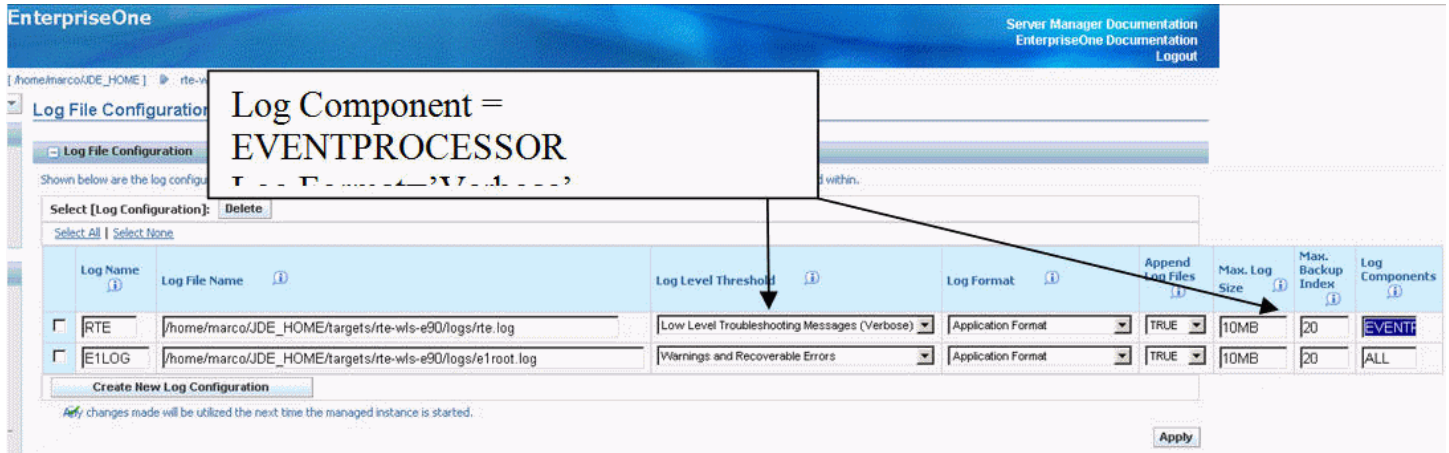
- Once the additional jdelog.properties definition is added as described in the preceding step, stop the Transaction Server, clear the RTE logs and produce the logs generated after going through the steps outlined in *Testing and Validating the JMSQUEUE Configuration*.

- Description of the results obtained when running the test described in *Testing and Validating the JMSQUEUE Configuration*

Specifically, you should be able to provide the status of the F90710 record after the RTE is started.

13 Appendix E - Clustering Real Time Events on WebSphere

Understanding WebSphere Clustering

The implementation of WebSphere clustering permits the deployment of application components and services to several machines with a single face to the client. When a client requests a service, it should make no difference if the service runs on a single server or across a number of servers. The clustering abstraction provides you with a clear route to improving the performance and scalability of the applications with increased administration of hardware and network resources. WebSphere clustering provides these benefits:

- Scalability

  A clustered solution allows you to create additional capacity by introducing more servers to the cluster, thereby reducing the load on existing servers. For the JD Edwards EnterpriseOne Transaction Server with the Real Time (RTE) Application, you can add as many servers and messaging engines as you like to meet the performance requirement. The RTE application on WebSphere achieves scalability with the implementation of a messaging engine policy.

- Load balancing

  The ability to distribute requests across all members of the cluster, according to the workload on each server. WebSphere handles load balancing as defined by the Load Balancing Algorithm that you configure on cluster setup. The Transaction Server with RTE on WebSphere only supports round-robin load balancing policy for the RTE application.

- High availability

  A mix of features that ensure applications and services are available even if a server or machine fails. Clients can continue to work with little or no disruption in a highly available environment. RTE on WebSphere achieves high availability with the implementation of a messaging engine policy.

ORACLE

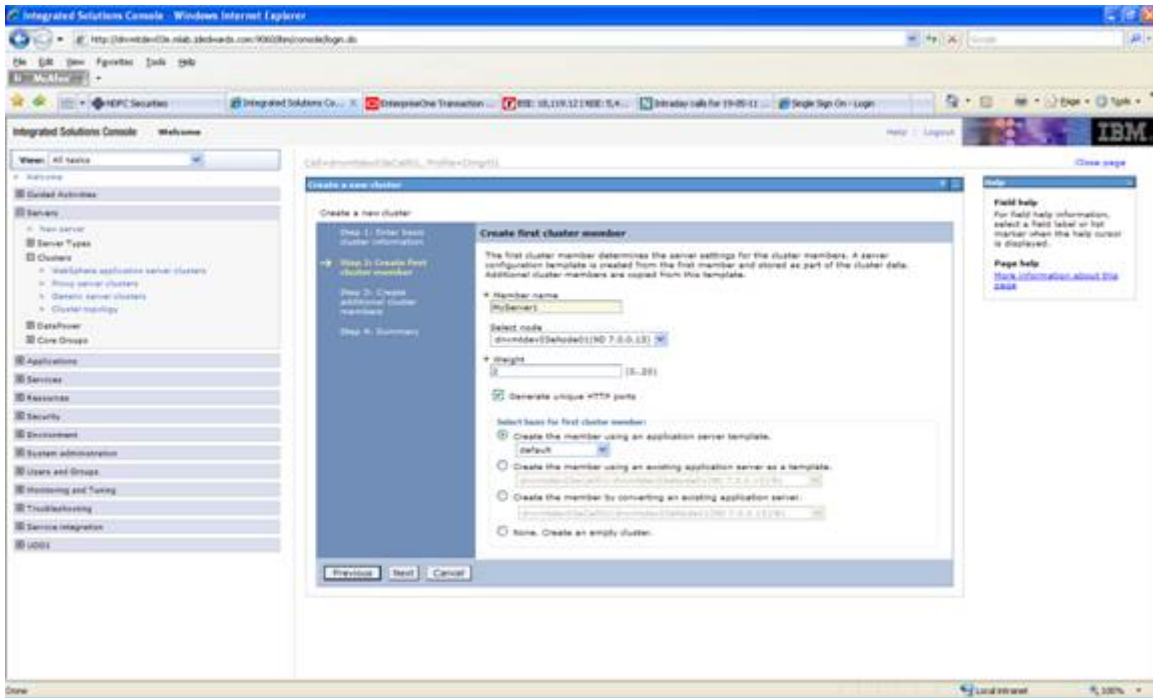# Creating a Cluster on the IBM WebSphere Application Server

To create a cluster on the WebSphere Application Server:

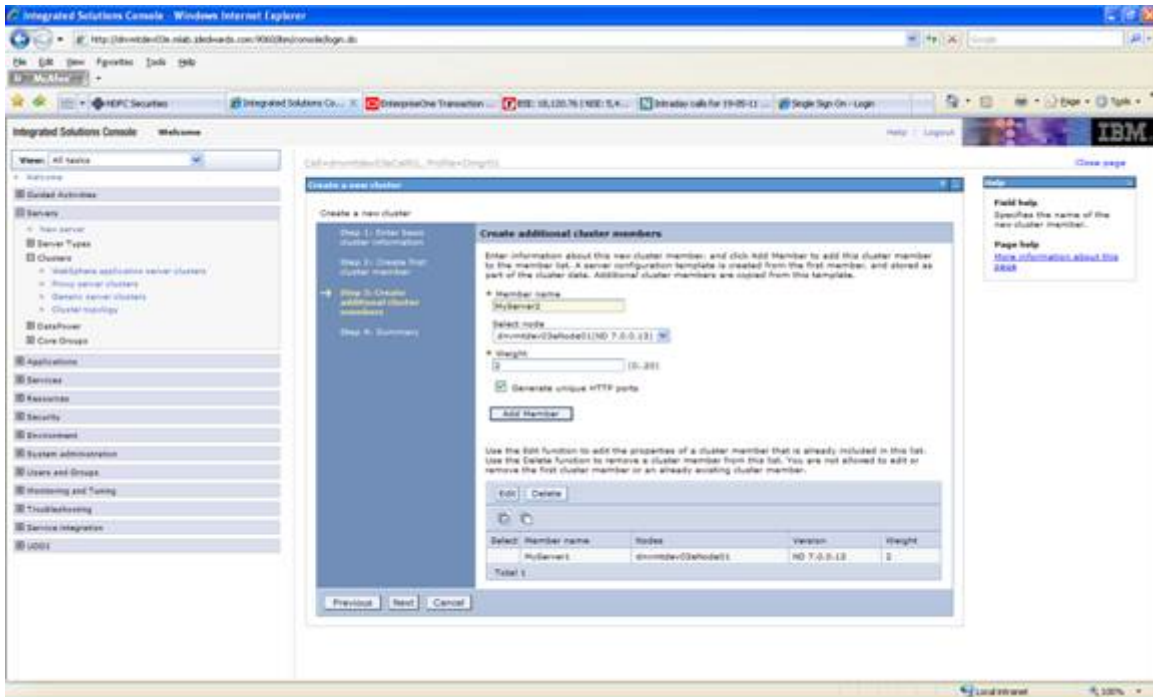1. Go to Servers > Clusters > WebSphere application server clusters.
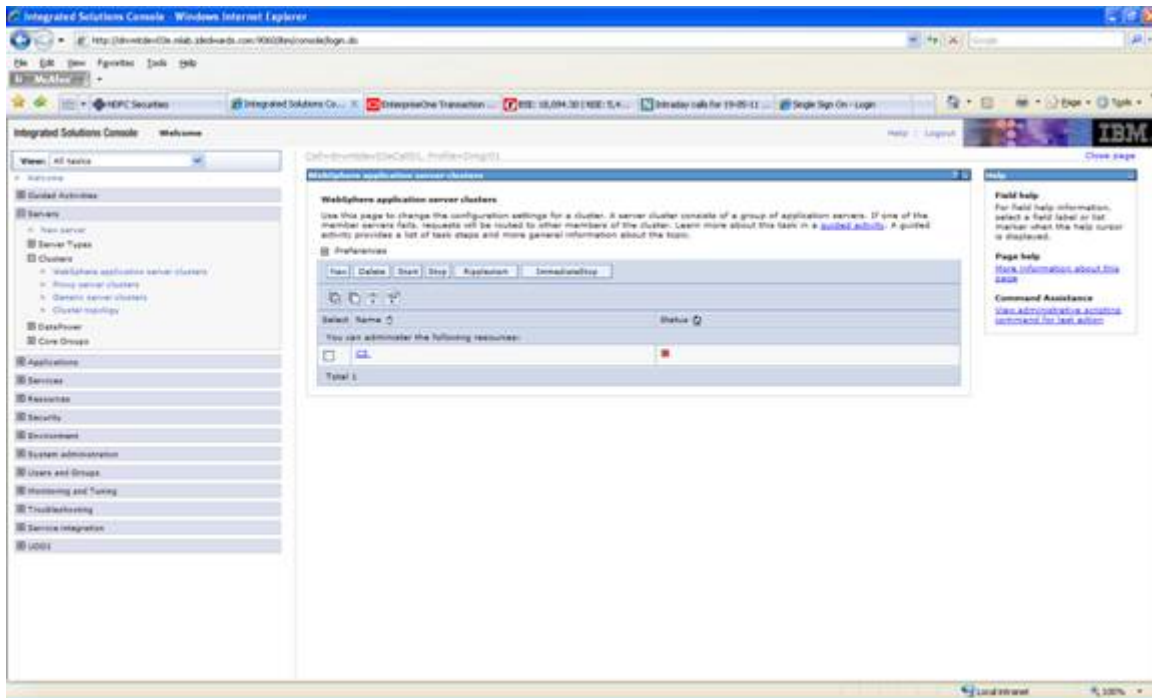
2. On WebSphere application server clusters, click the New button.

3. On the Create a new cluster dialog, enter the name of the cluster.

ORACLE

4. Click the Next button.

5. On Create first cluster member, in the Member name field, enter a name for the first cluster member.
6. Click the Next button.

ORACLE

7. On Create a new cluster, click the Add Member button.

Continue this step to add your additional cluster members.

> **Note:** When creating a horizontal cluster, you must add a server with the available node targeted to the remote host name.

ORACLE
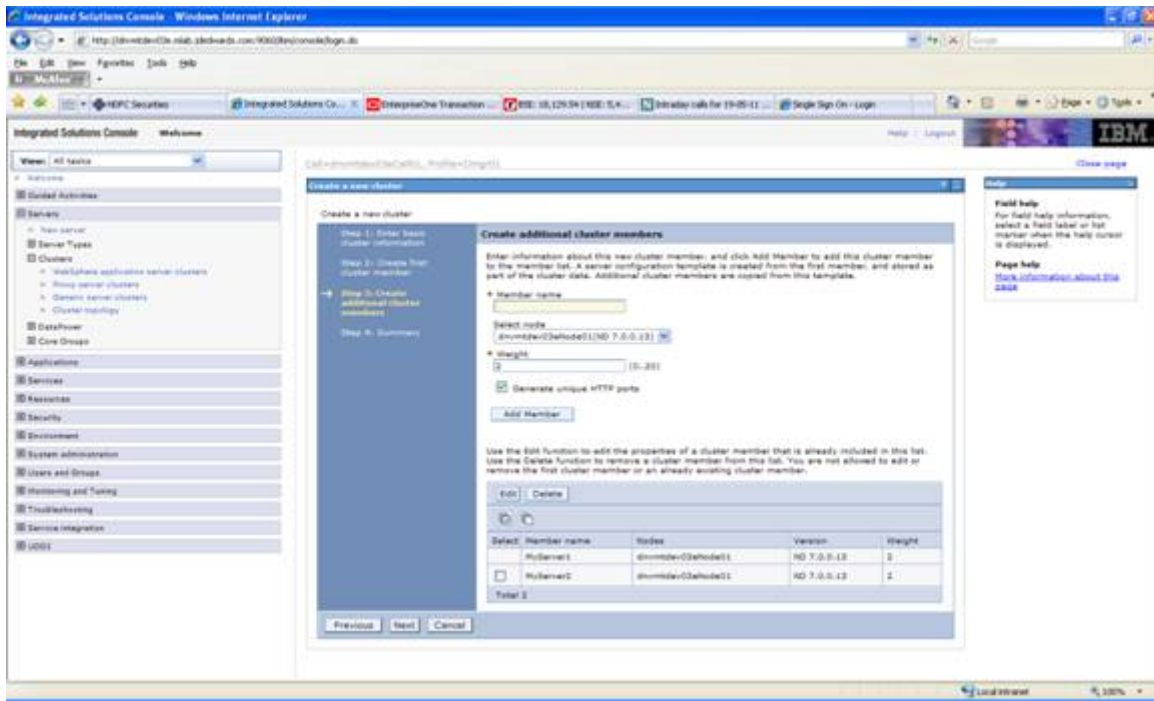
8. When you have finished adding members, click the Cancel button to exit the Create additional cluster members part of the wizard.
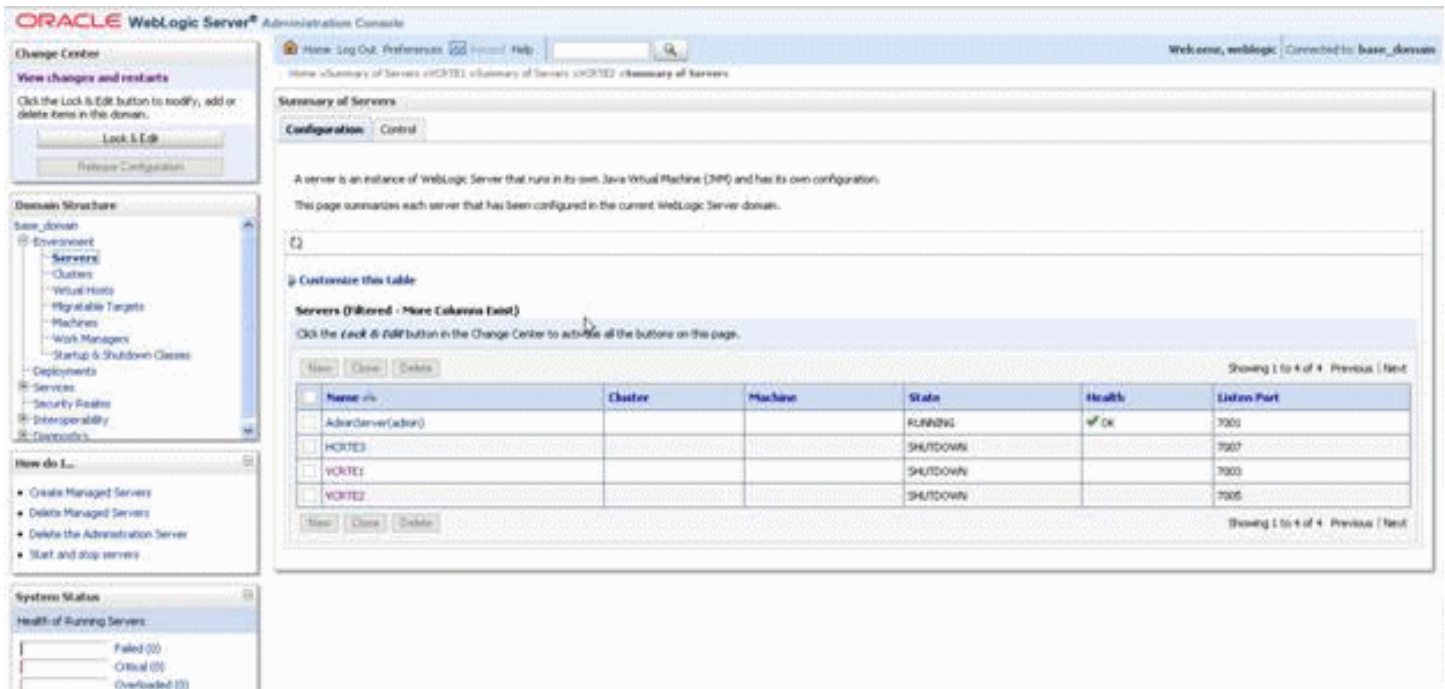
9. On Create a new cluster, click the Finish button.
10. Save the cluster configuration.

ORACLE

Installing the Transaction Server RTE Component on the WebSphere Application Server using Server Manager

1. Go to the Server Manager Console.

ORACLE
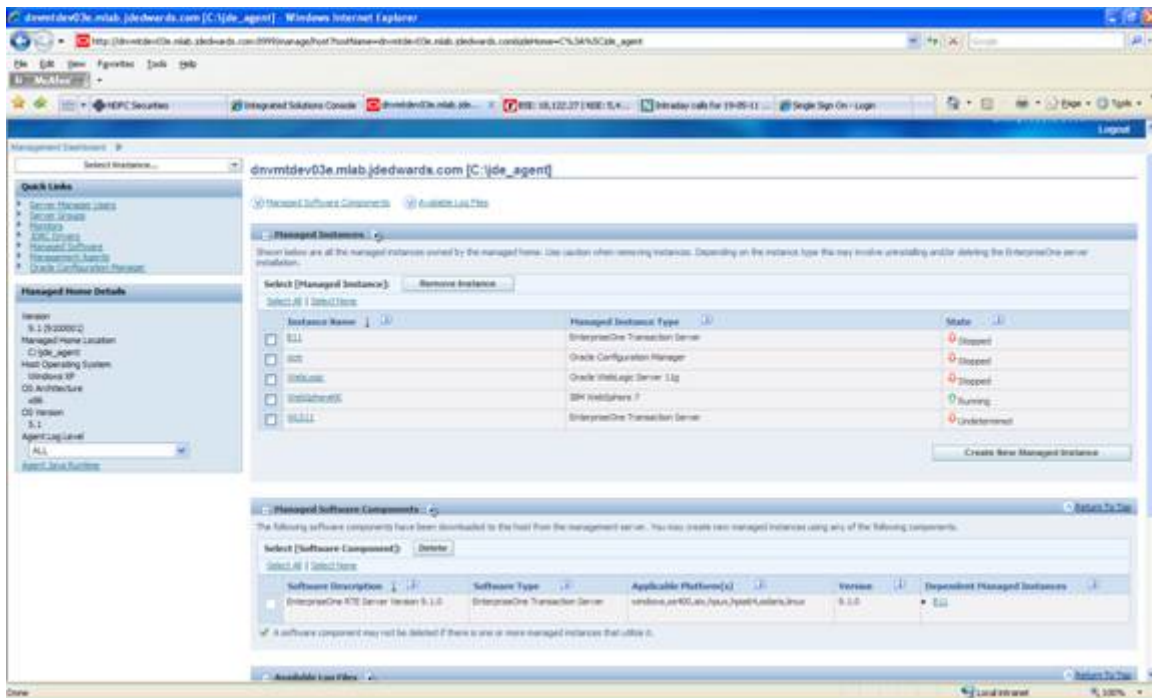
2. On the Server Manager Console, with the Managed Instances displayed, click the Create New Managed Instance button.
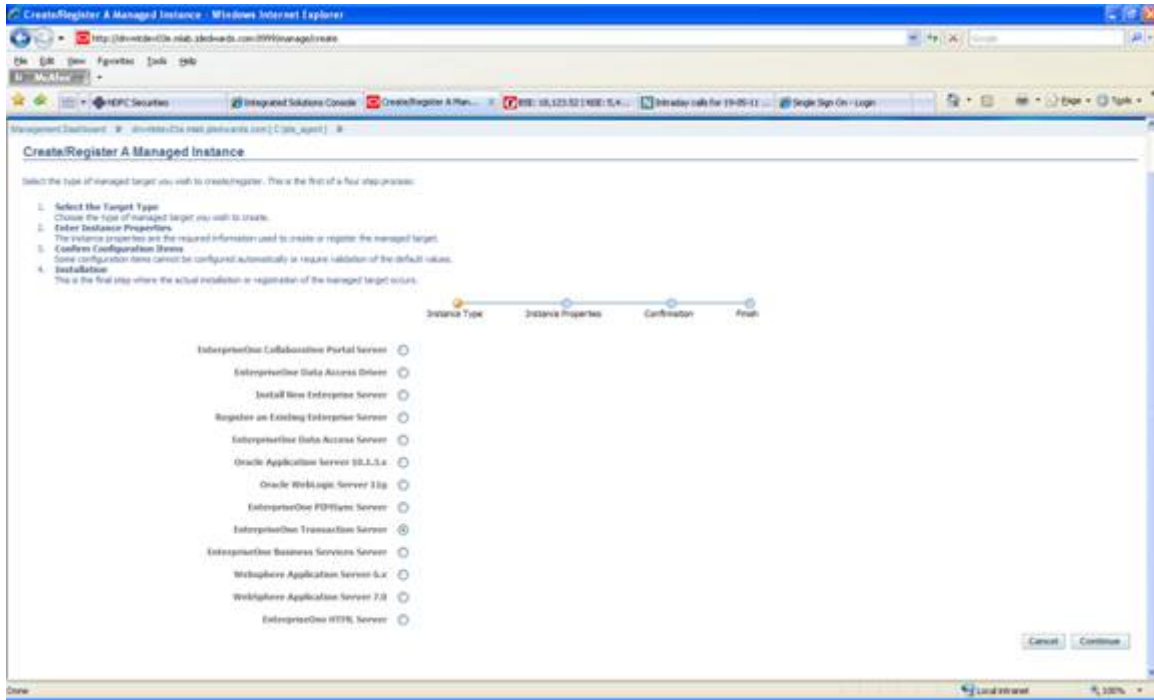
3. On Create/Register A Managed Instance, Instance Type, select the EnterpriseOne Transaction Server radio button.

4. Click the Continue button.
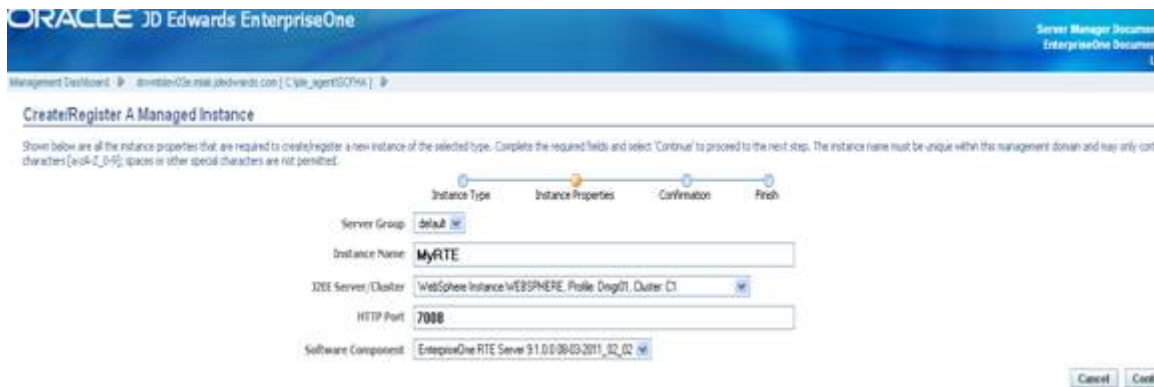
5. On Create/Register A Managed Instance, Instance Properties, complete these fields:
   o Instance Name
     Enter a value for your installation.
   o J2EE Server/Cluster
     Use the pulldown to select your cluster from the list of automatically detected clusters.
   o HTTP Port
     You must specify an unused port number.
   o Software Component

ORACLE

Use the pulldown to select the desired version of the EnterpriseOne RTE Server software component.

6. Click the Continue button.

7. On Create/Register A Managed Instance, Confirmation, click the Continue button.
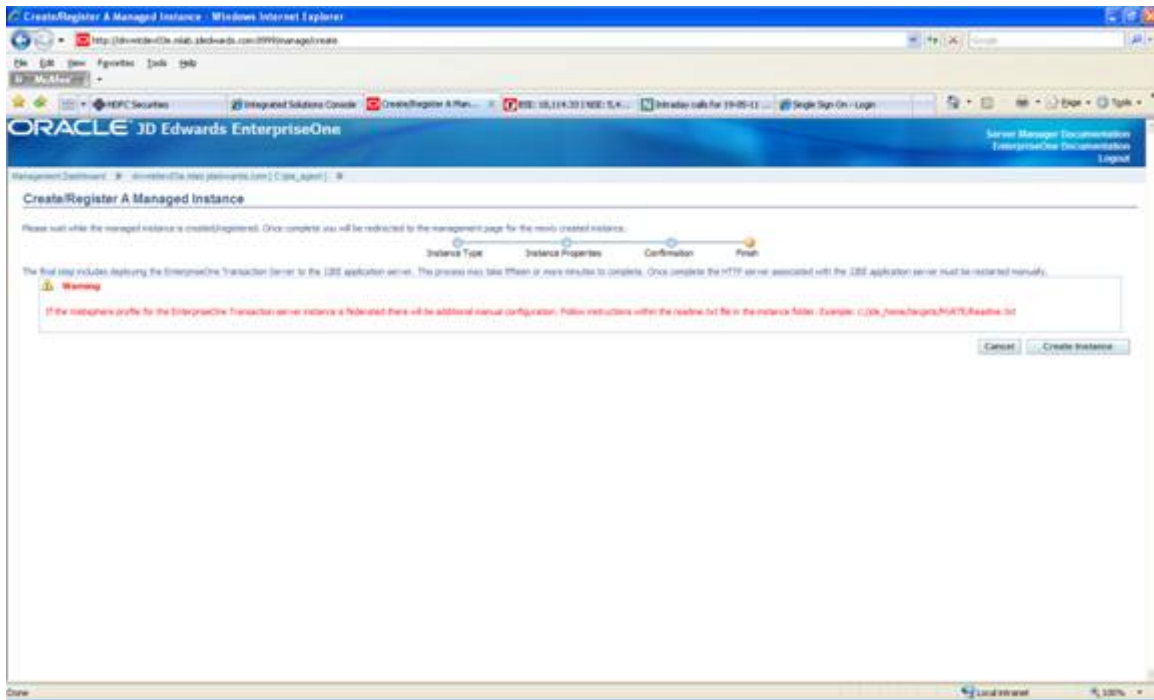
8. On Create/Register A Managed Instance, Finish, click the **Create Instance** button.
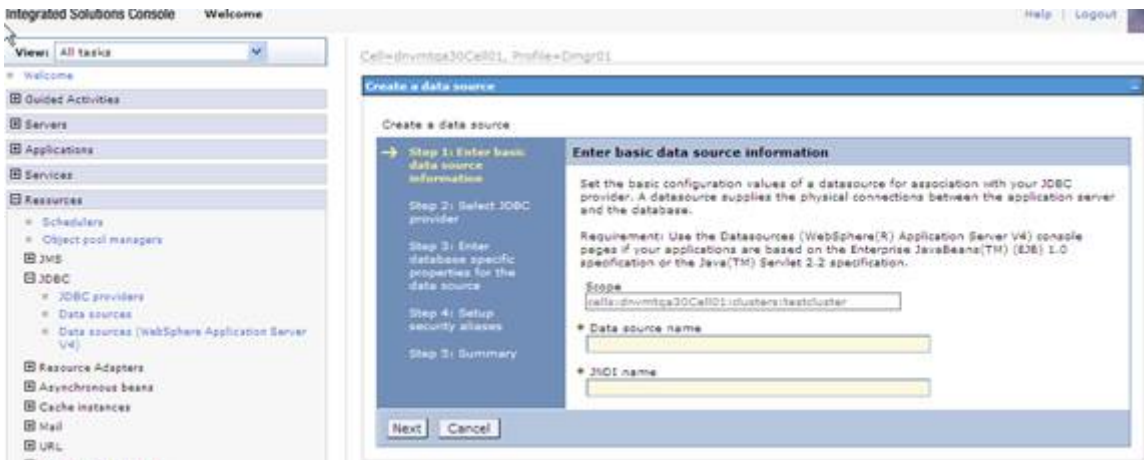
Using JDBC Store
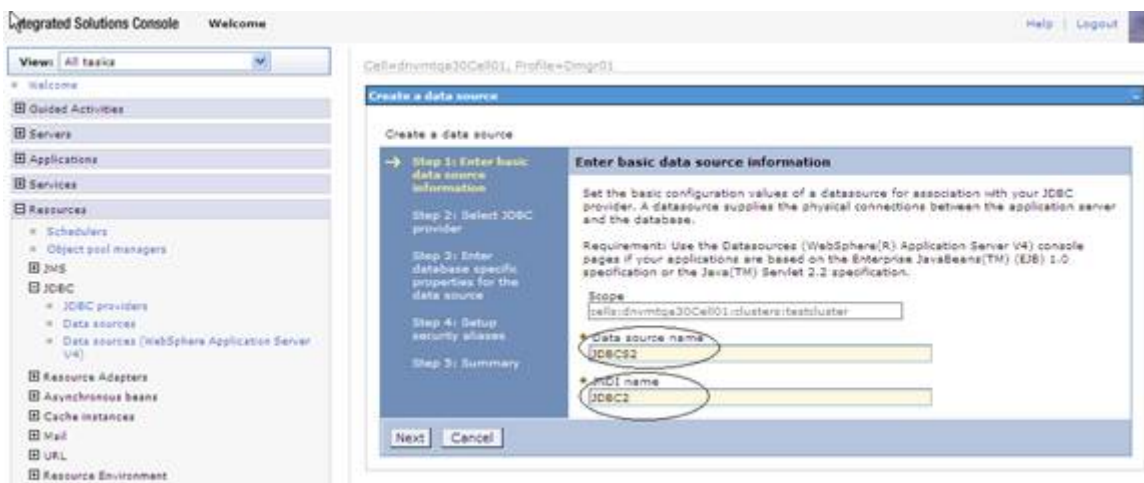
To use JDBC Store:

1. Login to WAS admin console.

2. Select Resources ->JDBC ->Data Sources from left panel of the console and choose the cluster scope.
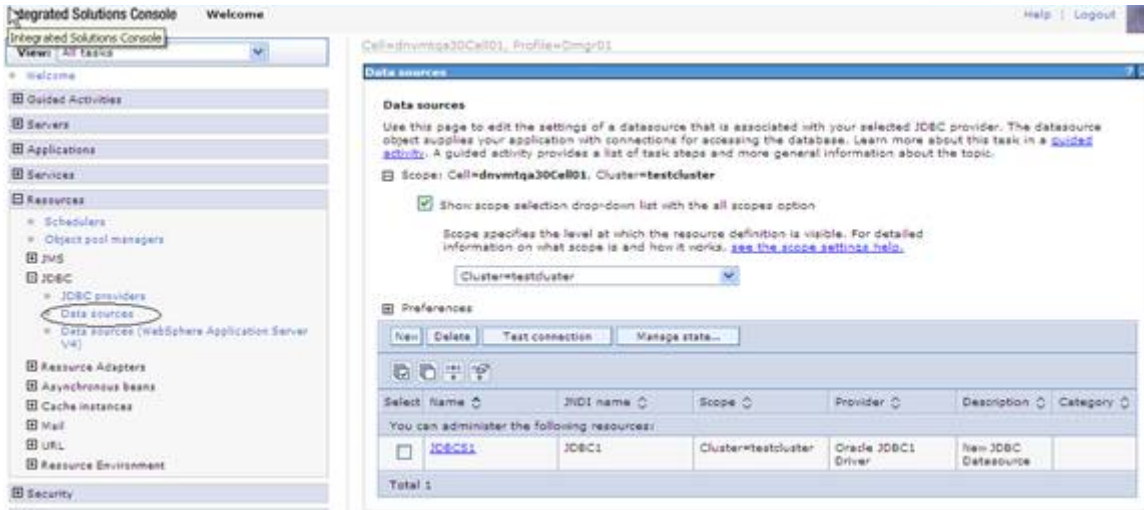
3. Click New and navigate to Create Data Source screen

4. Enter the Data Source name and JNDI name as shown in below screen shot and click Next

ORACLE

5. Select "Create new JDBC provider and click Next as shown in below screen


6. Fill in all the details as shown in the below screen shot and proceed to next screen clicking Next

   Note: Other Database types may be selected and similar configuration parameters may be selected.
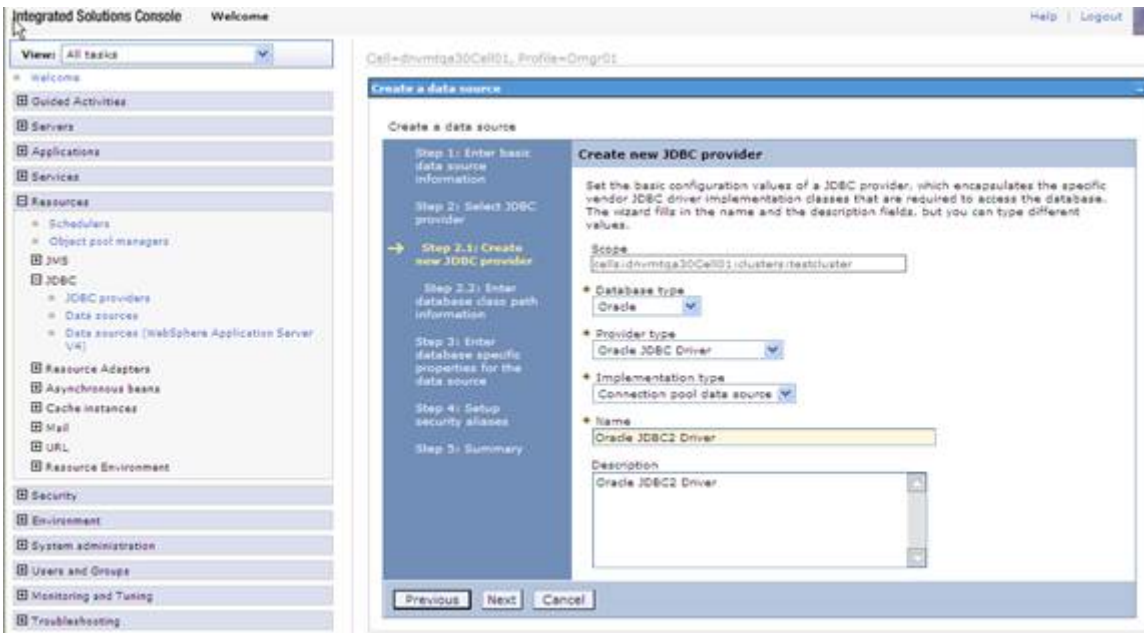
ORACLE

7. Give the directory location for ojdbc6.jar as seen in the below screen shot and Click Next.

> **Note:** This is a typical example for using Oracle database. For other database similar type 4 JDBC driver location is provided.
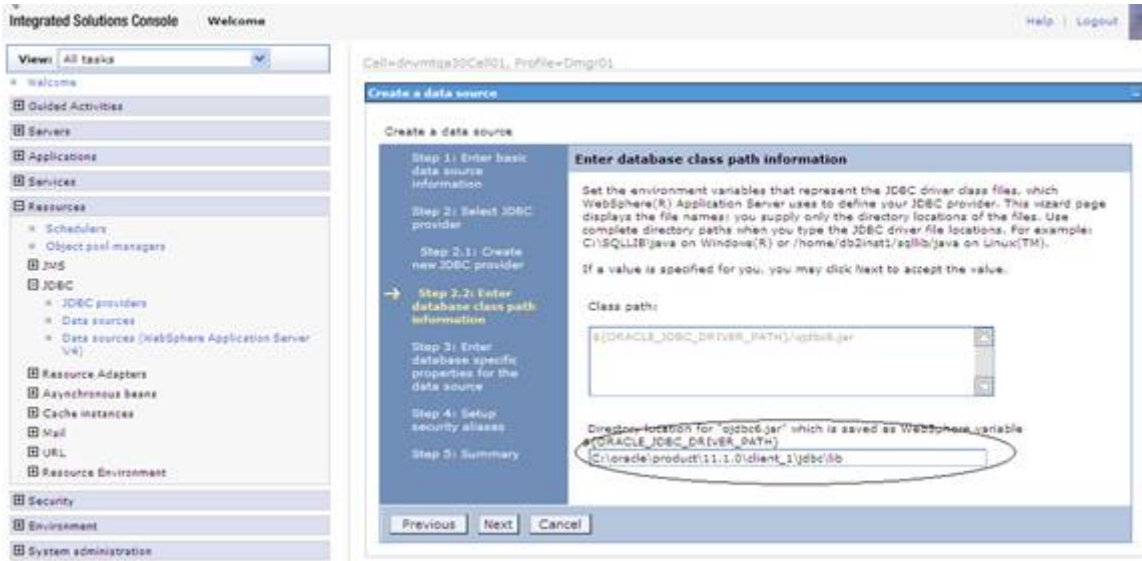
8. Fill in the appropriate URL for the database from which the data source obtains the connection and data store helper class name as shown in below screen shot and click the **Next** button.
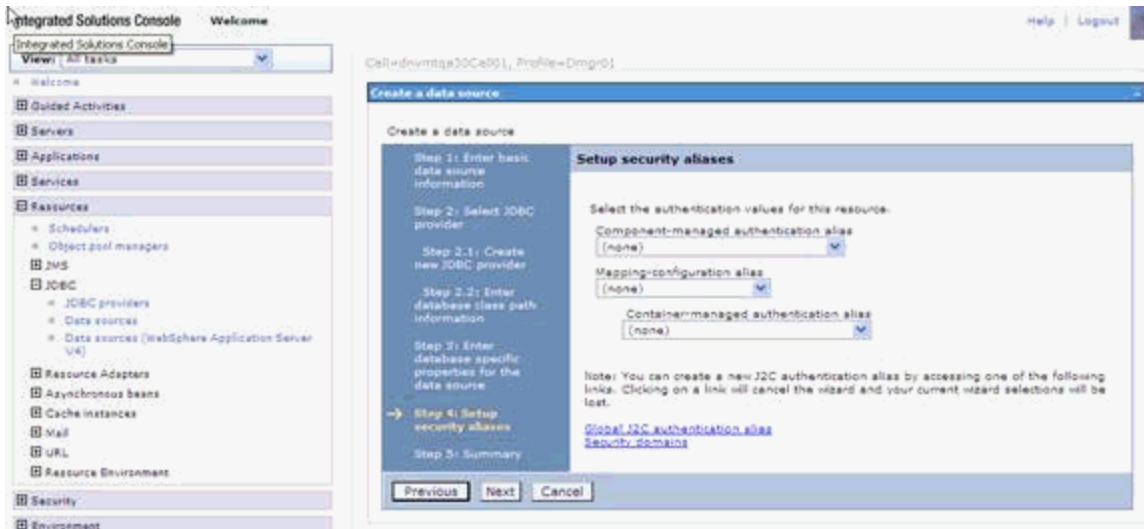
The URL should be similar to this:

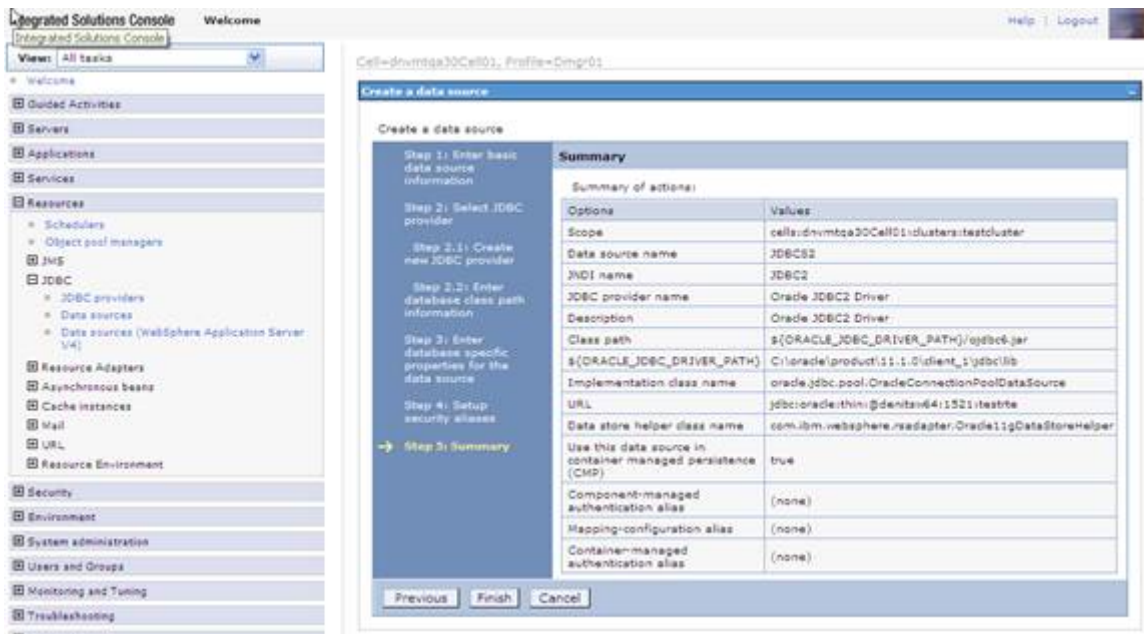jdbc:oracle:thin:@localhost:1521:sample" for thin driver

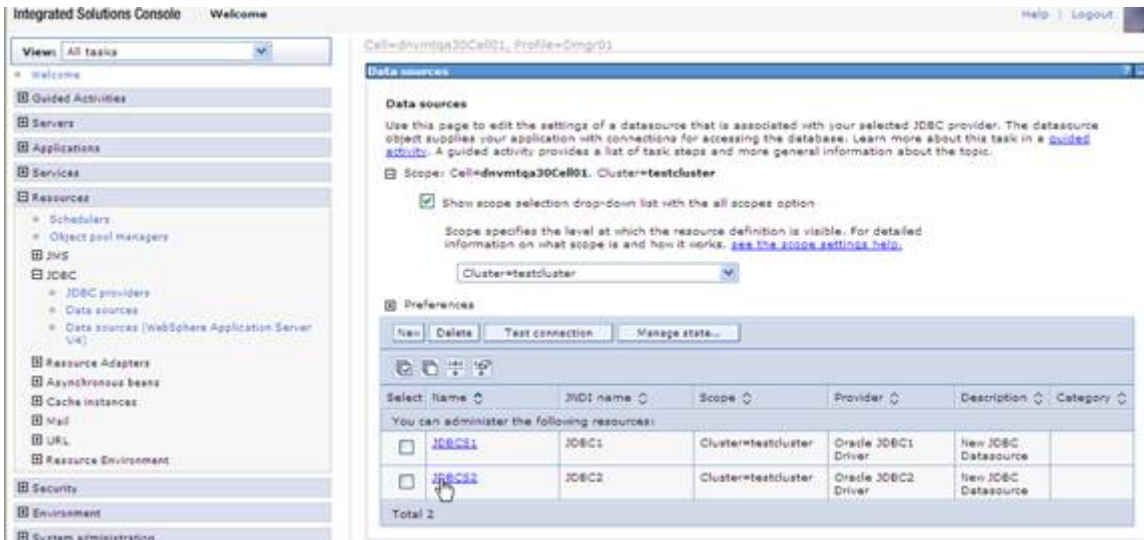> **Note:** The URL for other databases will differ.

ORACLE
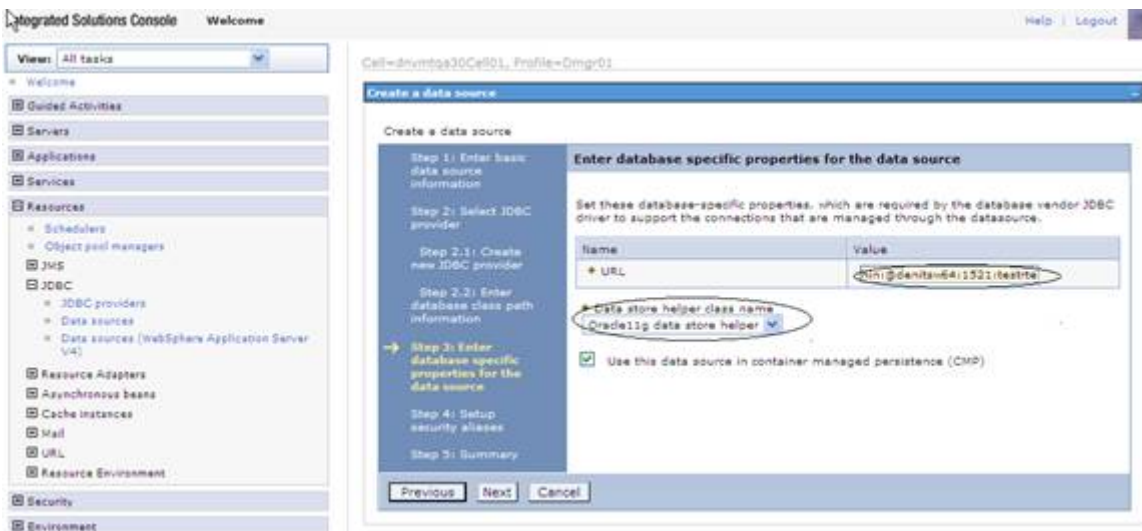
9. Click the Next button on setup security aliases screen as you would lose your current wizard selections by setting them up now. This can be set later.
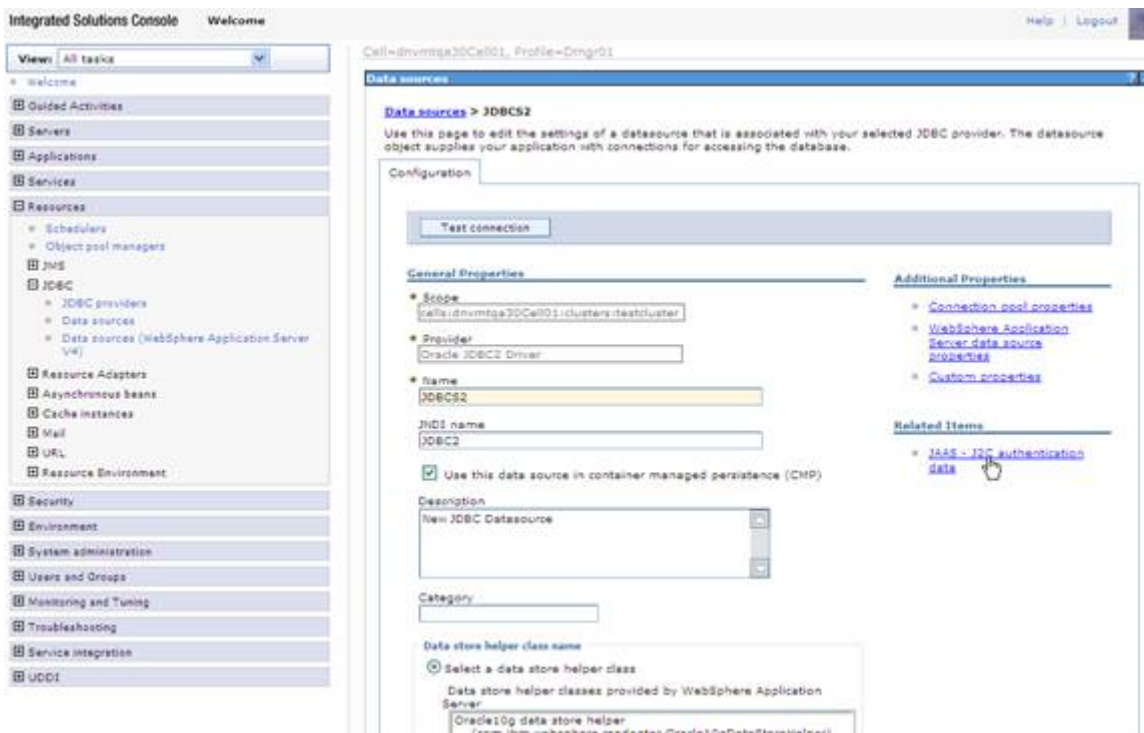

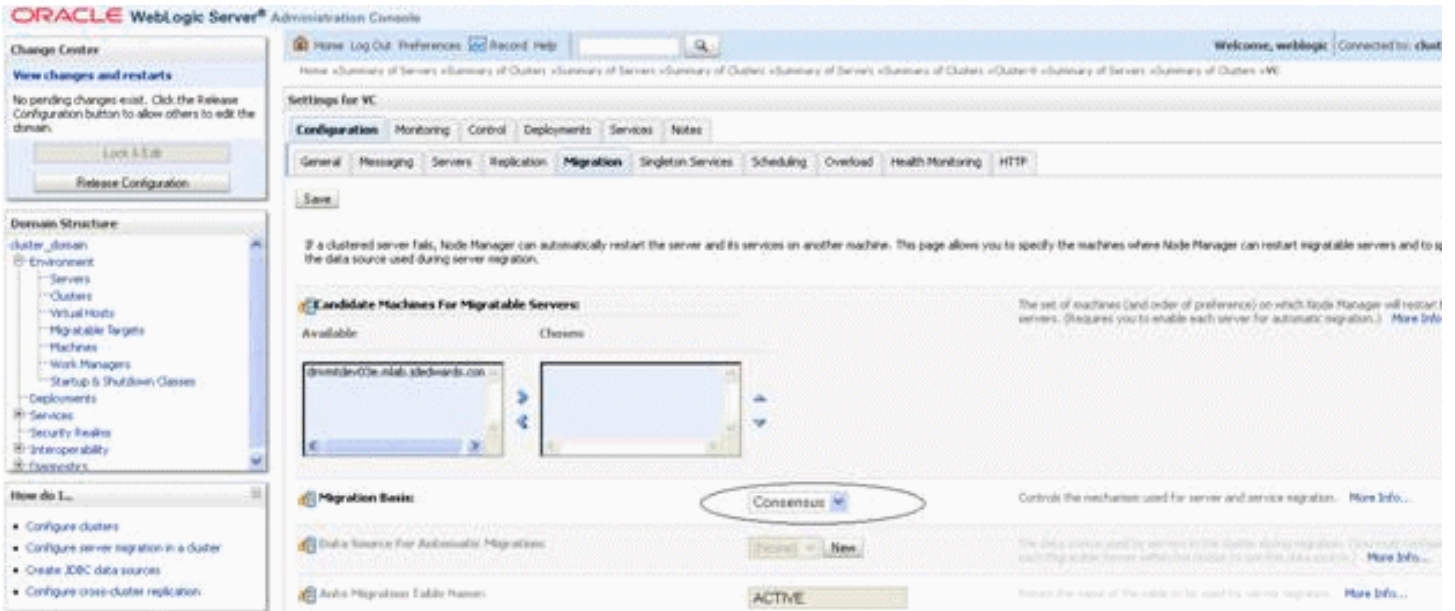10. Click Finish and navigate to the Create data sources screen.

ORACLE

**11.** Select the newly created Datasource and navigate to Test connection screen

12. On this screen select JAAS -J2C authentication data under "Related Items" as shown in below screen shot.
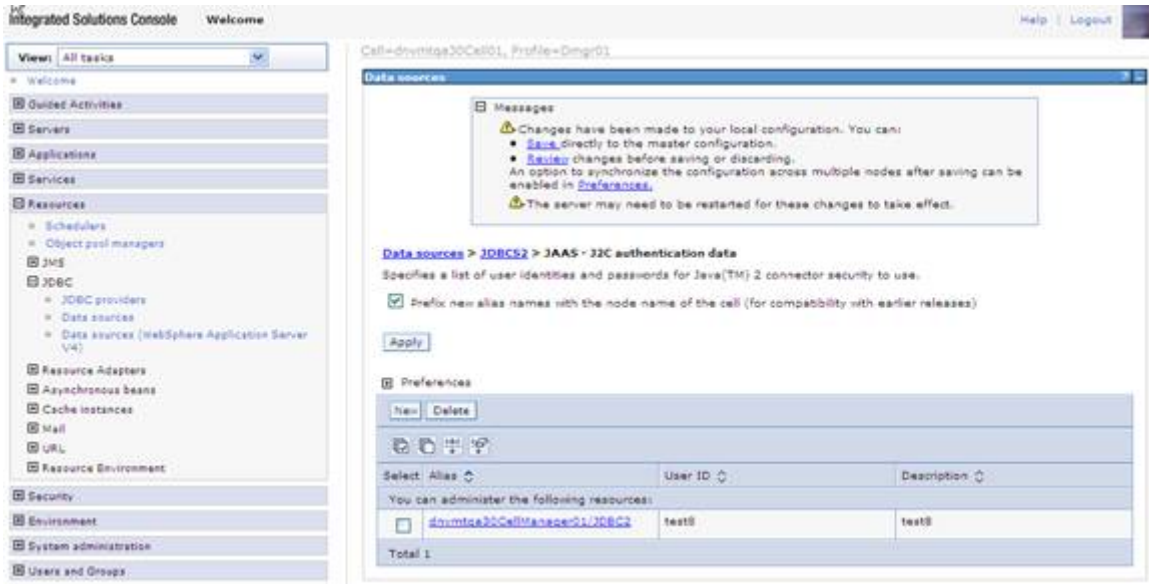
ORACLE

13. Select the **New** button.

14. Fill in appropriate values for all the fields of this screen as shown in below screen where User ID specifies the J2C authentication data for database connection.

ORACLE
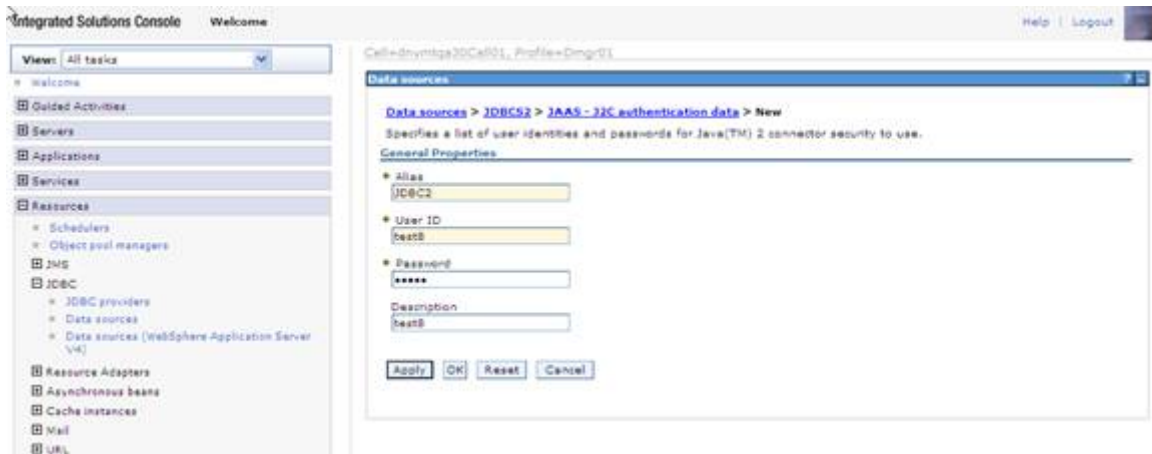
15. Click the OK button to save changes to master configuration.

16. Again select the data source from top of this screen as shown below.
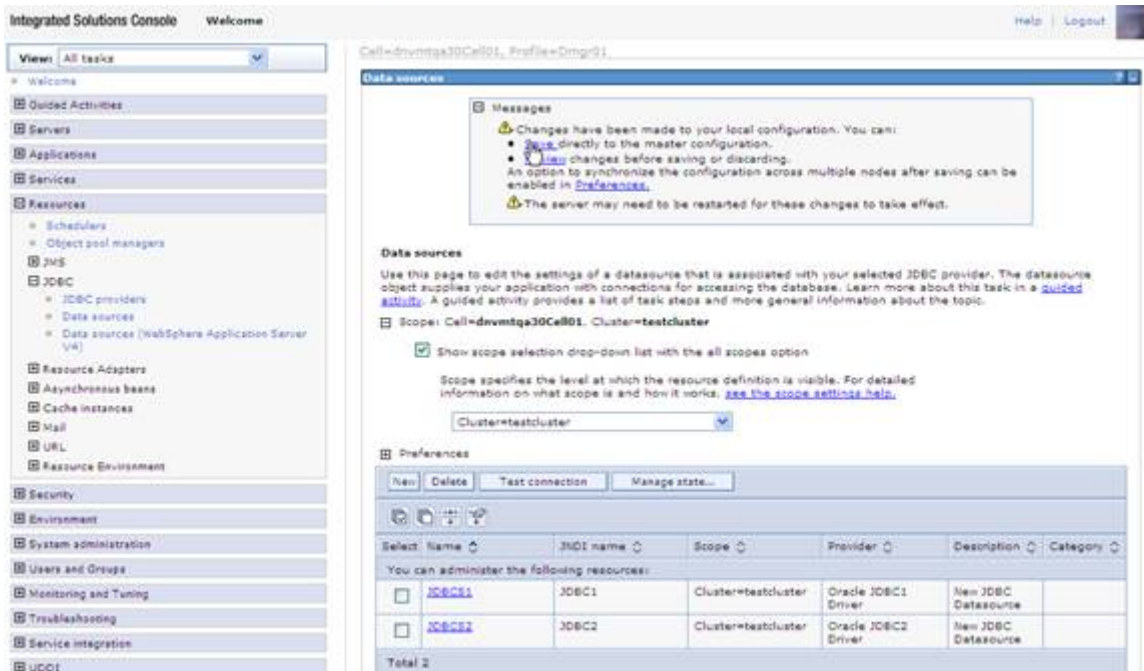

17. Go back to Data Source screen by clicking on the data source name from top of the screen.

18. On this screen chose the security settings newly created as shown in below screen shot and click the OK button.

19. Save changes to master configuration.

ORACLE

20. Check the newly created data source and click the Test Connection button.

21. If prompted, save changes to master configuration, then synchronize the changes and repeat the above step.

**22.** The test connection should be successful.

23. Repeat the steps in this section to create any number of data sources in the cluster scope.

Using Horizontal Clusters on WebSphere

The JD Edwards EnterpriseOne Transaction Server installation is configured with filestore for the messaging engines by default. If you use Horizontal cluster it is mandatory that you use the JDBC store.

ORACLE

To use horizontal clusters on WebSphere:

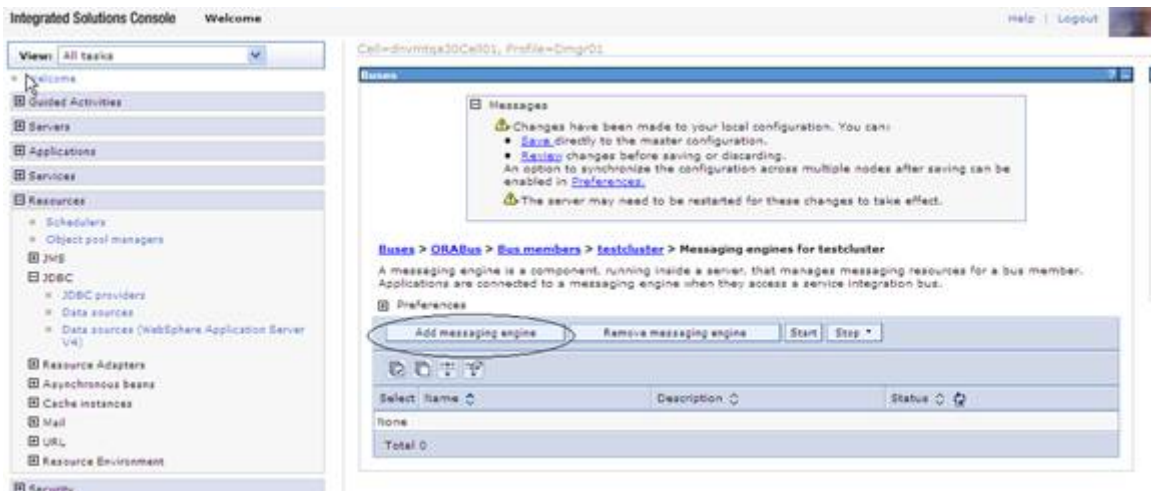1. Go to Service Integration > Buses.
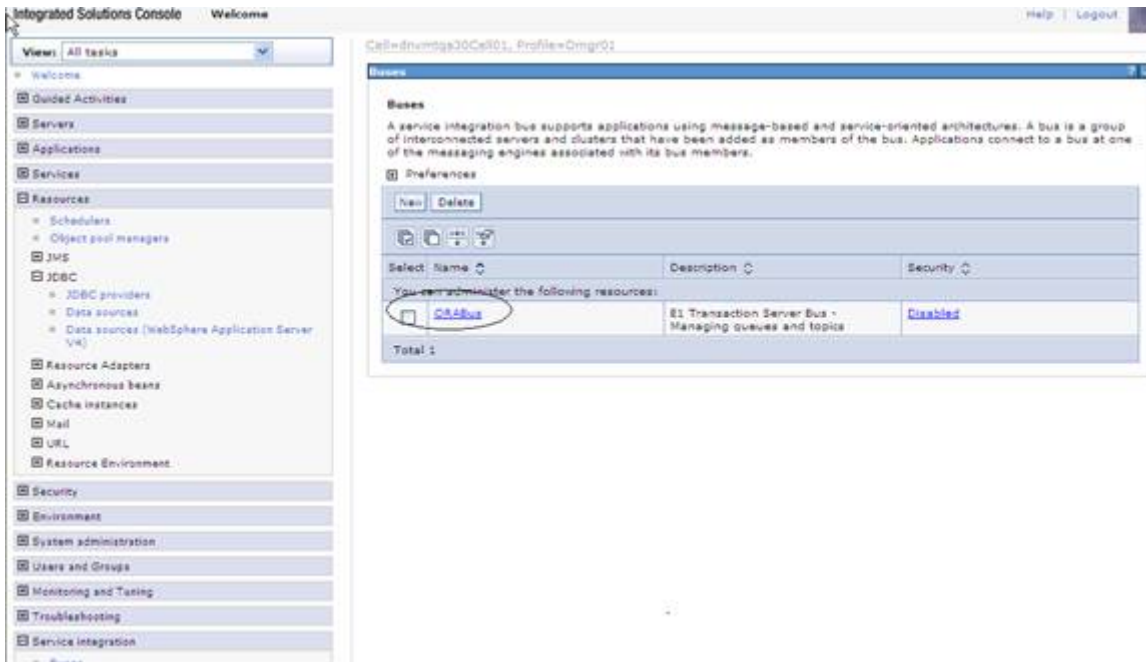
2. On Buses, select ORABus.
3. On Buses, select the Bus Members.

4. On Buses, Bus members, click on the link for the cluster.
5. On Bus members, click Additional Properties > Messaging engines.
6. Select the messaging engines created by default with the RTE installation.

ORACLE

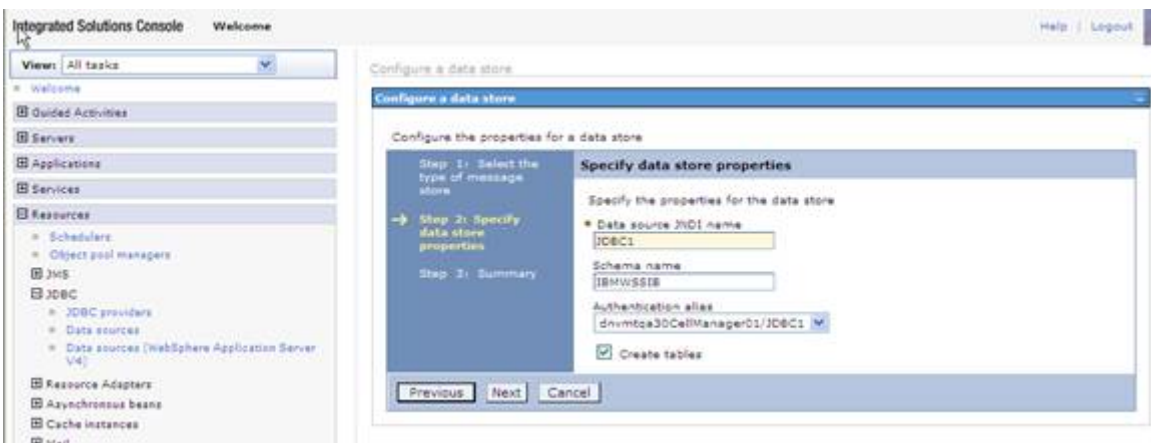7. Click the Remove messaging engine button to delete them.

8. On Buses, OraBus, Bus members, testcluster, Messaging engines for testcluster, click the Add messaging engine button.

9. On Select message store type, select the radio button for **Data store**.
10. Click the Next button.

11. On Specify data store properties, complete these fields:

    ○ Data source JNDI name

    Enter a name for the JNDI. For example, JDBC1.

    ○ Schema name

    Enter a name for the schema. For example, IBMWSSIB.

    ○ Authentication alias

    Enter an authentication alias. For example, <machine>CallManager01/JDBC1.

12. Click the Finish button.
13. Save changes to master configuration.
14. Repeat the same for creating Message Engines for other members of the cluster.
15. On deleting the default message engines created during the installation of the RTE application, all the Queues are also deleted. As a result, you must re-add these queues manually or you can run the rte.py script passing the required parameters.
16. Run this wsadmin command before starting the clustered servers/RTE instance.

    Go to the <WAS_DMGR_PROFILE_LOC>\bin on command prompt.

    Run the following command on the prompt:

    ```
    wsadmin -lang jython -f <JDE_AGENT_HOME>\bin\rte.py createSIBDestination <Bus-name> <Nodename>
    <Server_name> <Cluster_Name> <profilepath> <queue_names>
    ```

    For example:

    ```
    Z:\IBM\WebSphere\AppServer\profiles\Dmgr01\bin>wsadmin -lang jython -f Z:
    \JDE_HOME_WAS7\bin\rte.py createSIBDestination ORABus dnvmtqa30Node01
    MyServer1 MyCluster1 Z:\IBM\WebSphere\AppServer\profiles\Dmgr01
    SubQueue00,SubQueue01,SubQueue02,SubQueue03,SubQueue04,SubQueue05,SubQueue06,SubQueue07,SubQueue08,SubQueue09,Su
    ```

    Pass input as NULL where not applicable. For a cluster, input for Nodename and Server_name can be passed as NULL.