

Getting Started with the Oracle Workflow 2.6 Business Event System

An Oracle White Paper
December 2000

Getting Started with the Oracle Workflow 2.6 Business Event System

A Business Event occurs in an internet application whenever something of significance happens, or can happen. An example of a business event is the creation of a new Sales Order.

INTRODUCTION

The purpose of this document is to assist customers, consultants and developers already familiar with Oracle Workflow 2.5, through the set up and operation of the Oracle Workflow 2.6 Business Event System.

The Oracle Workflow Business Event System is a new internet application service delivered with Oracle Workflow 2.6. It extends the existing workflow functionality provided in Oracle Workflow 2.5 by providing the following features:

- The Event Manager
- Workflow Process Event Activities
- Additional Workflow functions to interrogate, modify, and route business event messages
- Enhanced Oracle8i Advanced Queues Integration
- Oracle Message Broker Integration

The Business Event System enables Oracle Workflow 2.6 to function in any of following scenarios:

- Systems Integration Messaging Hubs
- Distributed Applications Messaging
- Message based systems integration
- Non-invasive customization of packaged applications

For more detailed information on Oracle Workflow 2.6 refer to the Oracle Workflow 2.6 Datasheet, the Oracle Workflow 2.6 Guide and the Oracle Workflow 2.6 Server and Client Installation Notes.

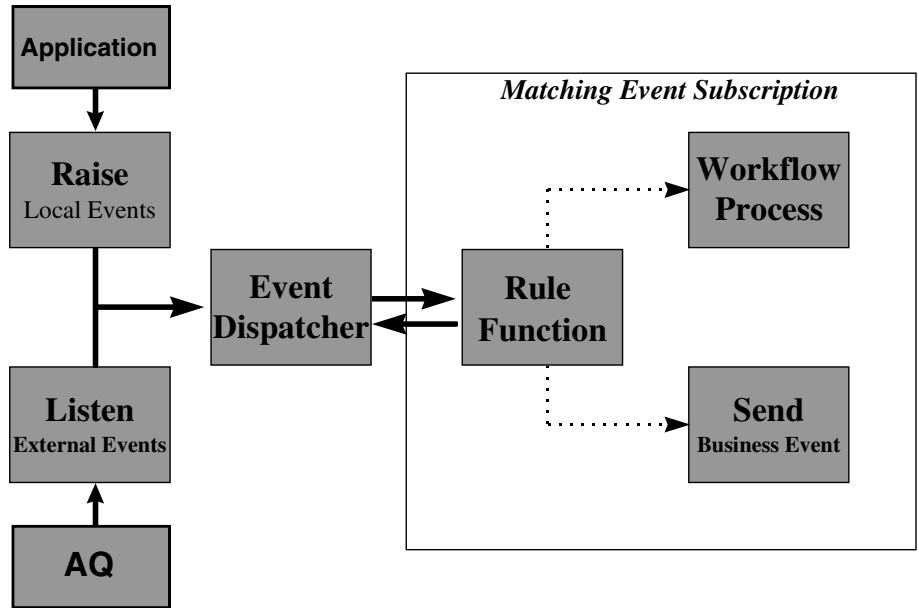
A *system* is a logically isolated piece of software on which either Oracle Workflow Business Event System is installed, or with which Oracle Workflow Business Event System communicates.

An *agent* is a communication node, which exists on a system. Typically, an agent on an Oracle database is an Oracle Advanced Queue. Customers can choose to use Oracle Workflow or custom PL/SQL code to handle enqueue and dequeue operations.

Event Manager

The Event Manager contains a registry of systems, agents, business events and subscriptions to business events. When a business event occurs, the Event Manager will search for matching subscriptions to the event, and execute the subscriptions. Subscriptions to a business event typically perform any or all of the following actions:

1. Execution of custom code on the event message,
2. Sending the business event to a defined workflow process, and
3. Routing the business event to an agent on either a local or external system.



When an event is raised locally or received from an external source, the Event Manager executes a rule function that can send the event to a workflow process or a local or external agent.

Workflow Process Event Activities

Workflow Process Event Activities enable workflow designers to model business events into workflow processes through the addition of a new activity type: Events. Workflow Event Activities may be used to:

- Receive a business event to launch a new workflow process or continue an existing running process,
- Send a business event to an agent on either a local or external system, and
- Raise a business event to execute processing on the local system.

Additional Workflow Functions

Additional workflow functions are provided to enable complex routing within workflow processes based on Event Message properties, and on the contents of any XML documents included in the message.

Oracle Message Broker Integration

Oracle Workflow 2.6 includes code that can enqueue and dequeue from Oracle Advanced Queues created by Oracle Message Broker. This enables Oracle Workflow 2.6 to support HTTP and HTTPS communication protocols, and to integrate with other messaging solutions such as MQ*Series and TIBCO.

Confirming Software Installation

The Oracle Workflow 2.6 Server Installation Notes detail all the tasks necessary to successfully install Oracle Workflow into an Oracle 8.1.6 or Oracle 8.1.7 database.

After completing all these tasks, log in to the Oracle Workflow 2.6 Homepage as the System Administrator user. Only users with the System Administrator privilege have access to Event Manager screens.



The Workflow home page provides access to Event Manager functions for users with System Administrator privilege.

System Creation

During the installation of Oracle Workflow 2.6, the local system should have been created automatically. Go to the Systems screen and confirm that a system has been created which matches the global name of the local database. Check on the Global Preferences screen that the Local System field has been set to this system.

Seed Data Upload

Go to the Agents screen. If all of the installation tasks have been completed, 3 agents should be displayed on this screen: WF_IN, WF_OUT and WF_ERROR. As each agent is specialized to process either inbound or outbound messages, this is the minimum number of agents required for the Business Event System to function. If a 'Load Seed Data' error message is displayed, complete the following tasks:

System administrators may add additional agents as required to support system messaging loads.

1. The upload of the remaining Business Event System seed data requires the Oracle XDK PL/SQL XML Parser to be installed in the local Oracle database and accessible to the Oracle Workflow schema. In Oracle 8.1.7 databases or higher with Jserver, the XML Parser may already have been installed automatically. If required, the Oracle Workflow Configuration Assistant installs the PL/SQL wrapper APIs required by the XML Parser. Loading of the JAR files required by the XML Parser is performed as a conditionally required Post Install step. The following JAR files must be uploaded using the loadjava command line utility: **xmlparserv2.jar** and **xmplsql.jar**. Refer to the Oracle Workflow Server Installation Notes Release 2.6 for details on how to check if the JAR files have been installed, the location of these files and the exact syntax to use.
2. The Business Event System seed data is uploaded into Oracle8i using the UTL_FILE PL/SQL Package. To enable the Oracle database to access file system directories, the **UTL_FILE_DIR** database parameter needs to be defined in your database init.ora file. The seed data file to be uploaded (wfvt.xml) is located in the workflow res directory. Typically the init.ora entry will look like:

UTL_FILE_DIR = <ORACLE_HOME>/wf/res on UNIX
UTL_FILE_DIR = <ORACLE_HOME>\wf\res on Windows NT
3. After this parameter is added, the database must be shutdown and restarted. Refer to your Oracle8i reference manual for more details on the UTL_FILE_DIR initialization parameter.
4. Log into the workflow schema in SQL*Plus, and run the wfdupld.sql script in the Workflow admin/sql directory. This script will prompt for 2 parameters:

Directory: The UTL_FILE_DIR directory entered in step 2

Filename: leave null
5. Confirm seed data is uploaded. Errors such as 'File Location or Filename invalid' indicate that you have either entered the wrong parameters or did not restart the database after modifying the UTL_FILE_DIR init.ora parameter.
6. Click on the remaining Event Manager menu options. Agents, Events, Event Groups and Event Subscriptions should have been automatically created.

Administration

Most administration tasks are performed on the Global Preferences and the Check Setup screens.

Global Preferences System Status

Business Event System functionality is controlled by the System Status field accessible from the Global Preferences screen. The Business Event System can have one of four states:

- Local only (default) - only local subscriptions can be executed,
- External only - only external subscriptions can be executed,
- Disabled - no subscriptions can be executed, and
- Enabled - local and external subscriptions can be executed.

The System Status value should be changed to Enabled so that local and external subscriptions are processed.

Check Setup: Confirm Init.ora parameters

Navigate to the Check Setup screen. The following init.ora parameters are required to be set for Oracle Advanced Queuing functionality:

- AQ_TM_PROCESSES
- JOB_QUEUE_INTERVAL
- JOB_QUEUE_PROCESSES

Confirm that the database parameters on the local database are equal to or above the recommended values on this screen. Refer to the Oracle Workflow 2.6 Server Installation Notes for more details.

Check Setup: Schedule Agent Listener Jobs

The Check Setup screen enables the scheduling of the Local Inbound Agent Listener database jobs. The function of the Agent Listener job is to dequeue event messages off inbound queues. To submit a database job for an Agent Listener, click on the *Create* hyperlink and enter a start date, and approximately how often the database job should be submitted. Submit database jobs for the WF_IN and WF_ERROR agents.

Note: Agent Listeners are only required for Inbound Agents, and are not required for Outbound Agents.

Note: The WF_ERROR Agent Listener must run regularly to ensure that the errored business events are processed and notifications are sent to the System Administrator.

Check Setup: Scheduling Queue Propagation

After an event message is enqueued onto an outbound agent, the event message may be propagated to an inbound agent on the local system or on an external system. As only a Local System and local agents have been created after the

installation and seed data upload, messages may only be propagated locally. Schedule Local propagation.

Note: Propagation is only required for Outbound Agents, and is not required for Inbound Agents.

Oracle Workflow Java Function Activity Agent

The Oracle Workflow Java Function Activity Agent scales to system load by enabling System Administrators to run this program in parallel as many times as required.

Customers and developers can write their own custom Java code and have the Oracle Workflow Java Function Activity Agent execute this code during workflow processing.

Oracle Workflow provides the Java Function Activity Agent, a perpetual running Java program, which utilizes the java based Oracle XML Parser to parse XML documents being routed through workflow processes. This program requires the installation of a Java Runtime Environment (JRE) which can be downloaded from <http://www.javasoft.com>.

The following new standard function activities use the Java Function Activity Agent:

- Get Tag Value
- Compare Tag Value
- Transform XML Document

Two sample scripts, one for NT and one for UNIX, located in the Workflow admin directory, serve as template files which can be edited and then used for submitting the Oracle Workflow Java Function Activity Agent.

On NT, edit the wfjvlsnr.bat file. Replace any references to <ORACLE_HOME> with the actual directory path of the Oracle home into which Oracle Workflow 2.6 is installed. It will also be necessary to edit the file to include the directory path to the Java Runtime Environment rt.jar file. To submit the Oracle Workflow Java Function Activity Agent, enter the following command:

```
wfjvlsnr.bat <workflow schema>/<workflow schema password@database>
```

On UNIX, edit the wfjvlsnr.csh file. Replace any references to <ORACLE_HOME> with the actual directory path of the Oracle home into which Oracle Workflow 2.6 is installed. It will also be necessary to edit the file to include the directory path to the Java Runtime Environment rt.jar file. To submit the Oracle Workflow Java Function Activity Agent, enter the following command:

```
wfjvlsnr.csh <workflow schema>/<workflow schema password@database>
```

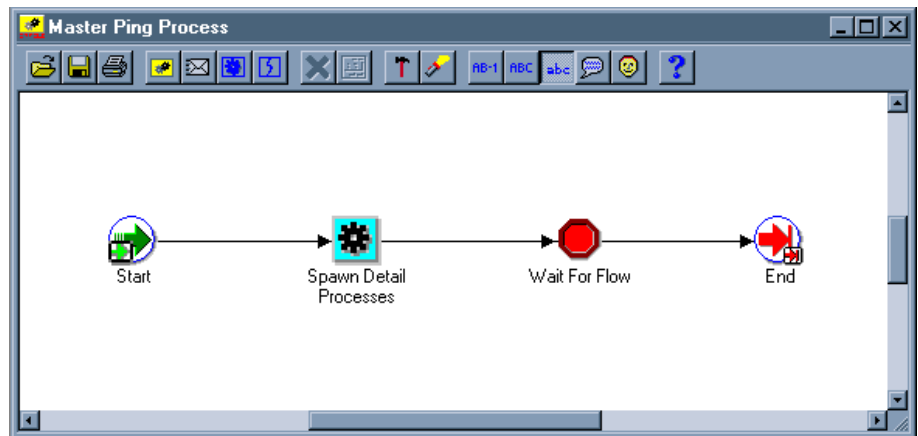
To stop the Oracle Workflow Java Function Activity Agent on either NT or UNIX, the admin/sql/wfjvstop.sql script needs to be run from the workflow schema. If multiple Oracle Workflow Java Function Activity Agents are running against a workflow schema, the wfjvstop.sql needs to be run once for every running Oracle Workflow Java Function Activity Agent.

Workflow Ping/Acknowledge of Inbound Agents

To confirm that Business Event System been set up correctly, a simple workflow process has been included which will automatically route an event message from a local outbound agent, to any inbound agents on local or external systems. This process will not complete until an acknowledgement event message is received back after the event message on the inbound agent has been processed.

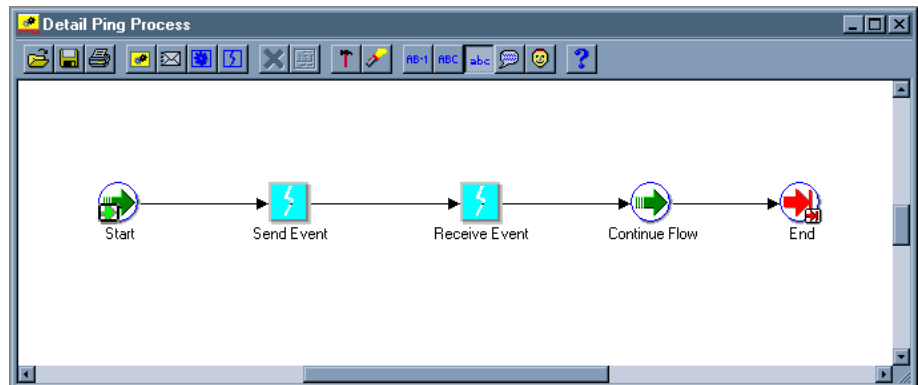
How Workflow Ping/Acknowledge works

The Ping/Acknowledge workflow consists of a master and a detail workflow process. The Master Ping process queries the agent tables for any inbound agents on local or external systems.



The Master Ping process launches a Detail Ping process for each inbound agent on a local or external system.

For each inbound agent, a Detail Ping process is launched. The detail process will send the *oracle.apps.wf.event.test.ping* event message from an outbound agent on the local system to the inbound agent identified by the master process.

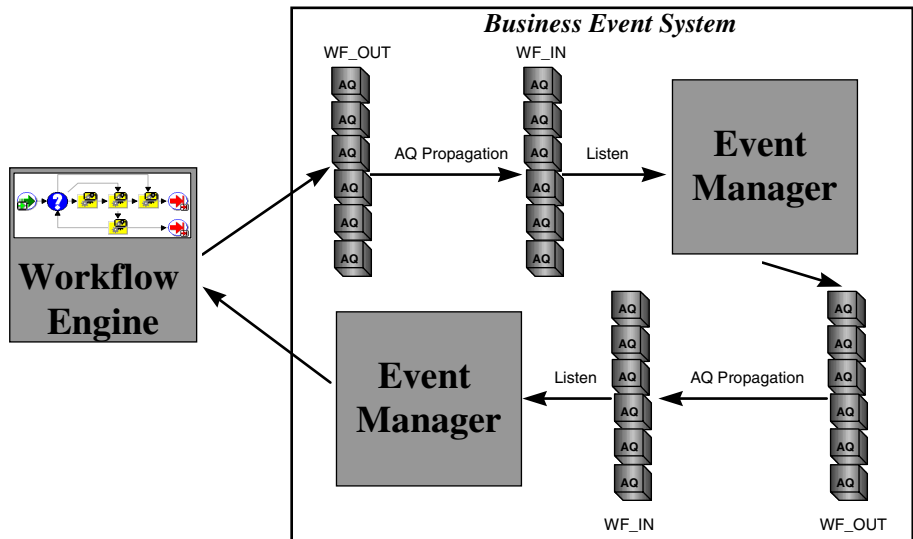


The Detail Ping process sends a ping event message to the inbound agent and waits to receive an acknowledgement.

From this point the following occurs:

1. The ping message is placed on the outbound agent addressed to the inbound agent.
2. Oracle Advanced Queue propagation will transmit the ping message from the outbound agent to the inbound agent.
3. Once the ping message is on the inbound agent, the Agent Listener database job will dequeue the ping message the next time it runs.
4. The Agent Listener will determine the event that has been dequeued, in this case the *oracle.apps.wf.event.test.ping* event, and pass the event message to the Dispatcher.
5. The Dispatcher will look for any active subscriptions for this business event with a source type of 'External'.
6. The Dispatcher will execute the seeded subscription to this event. The Rule Function run by this subscription will send an acknowledgement *oracle.apps.wf.event.test.ack* event message back to the system from which the ping message originated.
7. The acknowledgement message will be placed on an outbound agent addressed to an inbound agent on the originating system.
8. Oracle Advanced Queue propagation will transmit the acknowledgement message from the outbound agent to the inbound agent on the originating system.
9. Once the acknowledgement message is on the inbound agent on the originating system, the Agent Listener database job on this system will dequeue the acknowledgement message the next time it runs.
10. The Agent Listener will determine the event that has been dequeued, in this case the *oracle.apps.wf.event.test.ack* event, and pass the event message to the Dispatcher.
11. The Dispatcher will look for any active subscriptions for this business event with a source type of 'External'.
12. The Dispatcher will execute the seeded subscription to this event. This subscription will pass the message back to the Workflow Engine.
13. The Workflow Engine will complete the detail ping process, matching the event message to the running workflow process using the correlation ID, and the item type and process name defined in the event subscription.
14. When all detail workflow processes complete, the master workflow process will also complete.

The Event Manager Dispatcher controls which event subscriptions are executed in response to a business event.



Oracle Advanced Queue propagation transmits the ping and acknowledgement event messages between Business Event System agents.

Submitting Workflow Ping/Acknowledge Process

The Workflow Ping/Acknowledge process can also be used to ping systems on which Oracle Workflow 2.6 is not installed. Customers simply write their own code on the external system to dequeue the ping message and send an acknowledgement message back to the originating system.

Navigate to the Launch Processes screen. Select the Workflow Agent Ping/Acknowledge item type. Enter a unique item key, and select Master Ping as the process. Click submit, and note that at least 2 workflow processes have been launched - a master process which launches one detail process per inbound agent.

To confirm the processing of the Ping/Acknowledge event messages, navigate to the Event Queue Summary screen, and drill down on the event messages to see the types and statuses of event messages on the WF_IN and WF_OUT queues.

Enabling Communication between Oracle Workflow Business Event Systems

Event Messages can come into the Oracle Workflow Business Event System from Oracle databases and non-Oracle data sources. The installation of Oracle Workflow 2.6 is not a prerequisite to enabling communication between different systems, though it streamlines the process of getting connected.

A simple example of communication is between 2 installations of the Oracle Workflow Business Event System into 2 different schemas on the same Oracle database. Appendix A details the post install steps that are required to be completed after installing Oracle Workflow 2.6 into 2 schemas on the same database. A slightly more complex example, though more relevant to the real world, is the communication between 2 installations of the Oracle Workflow Business Event System in 2 separate Oracle databases.

The following instructions explain how to sign up additional systems to an Oracle Workflow Business Event System, regardless of whether the systems are in a local or external database.

Creation of Database Links

1. Designate one of the systems to be BES#1, and the other to be BES#2.
2. On BES#1, create a database link to connect the schema and database of BES#2.
3. On BES#2, create a database link to connect the schema and database of BES#1.
4. On BES#1, ensure that the database link components of the addresses of the default agents installed by the Workflow Configuration Assistant have the same name as the database link created on BES#2 to connect to BES#1. For example, if the address of the WF_IN agent on BES#1 is WF26.WF_IN@WF817.US.ORACLE.COM, the database link on BES#2 to connect to BES#1 must also be WF817.US.ORACLE.COM.
5. On BES#2, ensure that the database link components of the addresses of the default agents installed by the Workflow Configuration Assistant have the same name as the database link created on BES#1 to connect to BES#2. For example, if the address of the WF_IN agent on BES#2 is WF26.WF_IN@WF816.US.ORACLE.COM, the database link on BES#1 to connect to BES#2 must also be WF816.US.ORACLE.COM.
6. To confirm the set up is still correct, submit Workflow Ping/Acknowledge of Local Inbound Agents in BES#1.
7. To confirm the set up is still correct, submit Workflow Ping/Acknowledge of Local Inbound Agents in BES#2.

Oracle DBA Studio allows you to create database links without needing to know the SQL syntax.

System Sign up

8. On BES#1, click on the System Identifier hyperlink, and save the file to disk.
9. On BES#2, go to the System Signup screen. Cut and paste the contents of the file saved in step 8 into the Event Data field. Enter a unique event key, and submit the event.
10. On BES#2, go to the System and Agent screens and confirm that the BES#1 system and BES#1 inbound agents have been created.
11. On BES#2, click on the System Identifier hyperlink, and save the file to disk.
12. On BES#1, go to the System Signup screen. Cut and paste the contents of the file saved in step 11 into the Event Data field. Enter a unique event key, and submit the event.
13. On BES#1, go to the System and Agent screens and confirm that the BES#2 system and BES#2 inbound agents have been created.
14. On BES#1, go to the Check Setup screen. In the Database Links region, confirm the existence of the database link to connect to BES#2. Schedule propagation from the WF_OUT queue to BES#2. The destination name should be identical to the database link name created on BES#1 to connect to BES#2.
15. On BES#2, go to the Check Setup screen. In the Database Links region, confirm the existence of the database link to connect to BES#1. Schedule propagation from the WF_OUT queue to the BES#1. The destination name should be identical to the database link name created on BES#2 to connect to BES#1.

Confirm System Sign up: Workflow Ping/Acknowledge of External Inbound Agents

16. To confirm that system sign up was successful, launch the Workflow Ping/Acknowledge process on both systems, and verify that the workflow processes complete.

Workflow Send Protocol

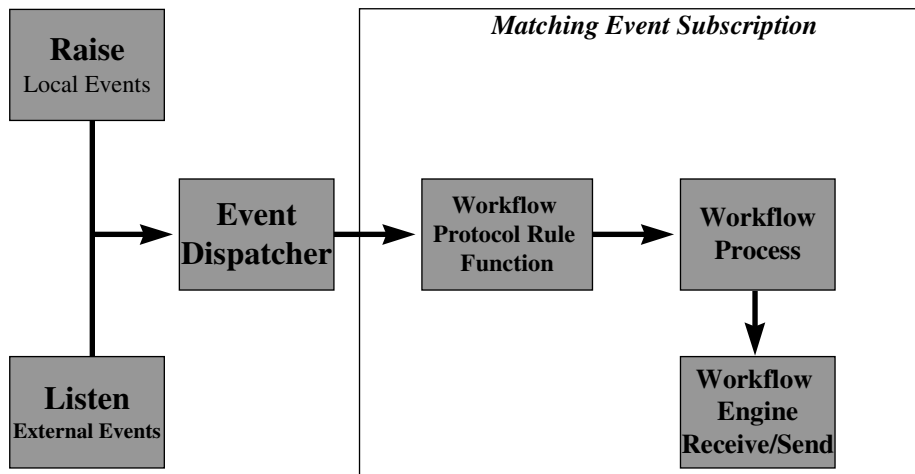
A sample workflow is included with Oracle Workflow 2.6 to demonstrate the sending, receiving and acknowledgement of messages from the Business Event System. The Workflow Send Protocol workflow process sends messages to the To Agent defined in the calling Event Subscription.

Unlike the Workflow Ping/Acknowledge process described above, the Workflow Send Protocol process is launched via the Business Event System.

Workflow Protocol Rule Function

A Rule function is a PL/SQL function, which is executed when an Event Subscription is processed. Rule function logic controls whether the workflow engine is called, or whether an event message is sent. Oracle Workflow provides a default rule function, which launches a workflow process and routes events to agents. The *Workflow Protocol* Rule Function is an example of a specialized rule function, which controls the behavior of an event subscription so that only the workflow engine is called.

The Rule Function API is explained in more detail in the Oracle Workflow 2.6 Guide.

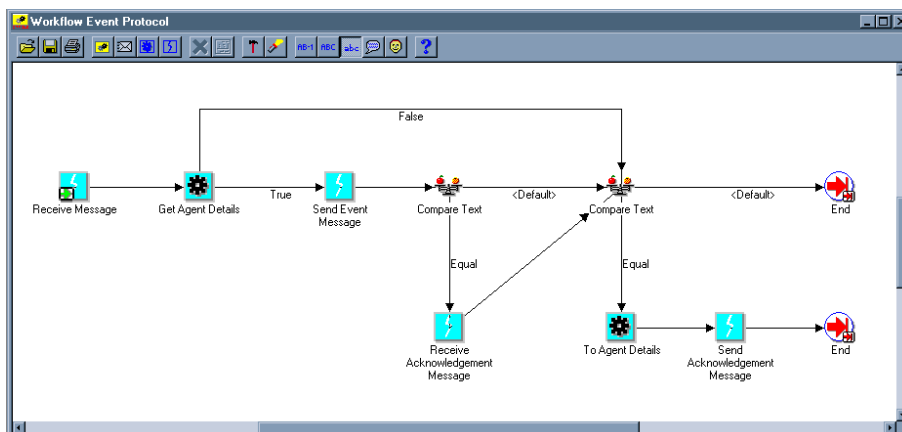


The Workflow Protocol rule function sends an event message to a workflow process, which can in turn route the message to an agent.

Workflow Send Protocol Process

The Workflow Send Protocol workflow process starts with a receive activity which receives the event from the Event Manager. The Get Agent activity retrieves the Out and To Agent details from the calling event subscription, and sends the event message. If the Out and To Agent details do not exist, no send is performed. If an Acknowledgement message is required for the send message, the process will wait on a receive activity which filters on the *oracle.apps.wf.event.wf.ack* event. When the acknowledgement message is received, the workflow process will continue. If the event message that launched this workflow process required an acknowledgement

message, the *oracle.apps.wf.event.wf.ack* event message is sent and the workflow process completes.



The Workflow Event Protocol process can send an event message, wait for an acknowledgement, and send an acknowledgement of the event message it received.

Workflow Send Protocol Demonstration Set Up

1. A local subscription has been defined to the *oracle.apps.wf.event.wf.send* event. Enter WF_OUT on BES#1 as the Out Agent, and WF_IN on BES#2 as the To Agent. The Rule Function for this subscription is *wf_rule.workflow_protocol*. This rule function only calls to the workflow engine, and does not send the event message. If an acknowledgement message is required, the Parameters section of the calling Event Subscription will have a name=value pair ACKREQ=Y. The Workflow Item Type to be run is the Workflow Send Protocol, with the start Process Name of Workflow Event Protocol.
2. On BES#2, verify that an external subscription has been defined to the *oracle.apps.wf.event.wf.send* event. The Rule Function for this subscription is *wf_rule.workflow_protocol*. The Workflow Item Type to be run is the Workflow Send Protocol, with the start Process Name of Workflow Event Protocol.
3. On BES#1, verify that an external subscription has been defined to the *oracle.apps.wf.event.wf.ack* event. The Rule Function for this subscription is *wf_rule.workflow_protocol*. The Workflow Item Type to be run is the Workflow Send Protocol, with the start Process Name of Workflow Event Protocol.

Sending a Business Event Using the Workflow Send Protocol

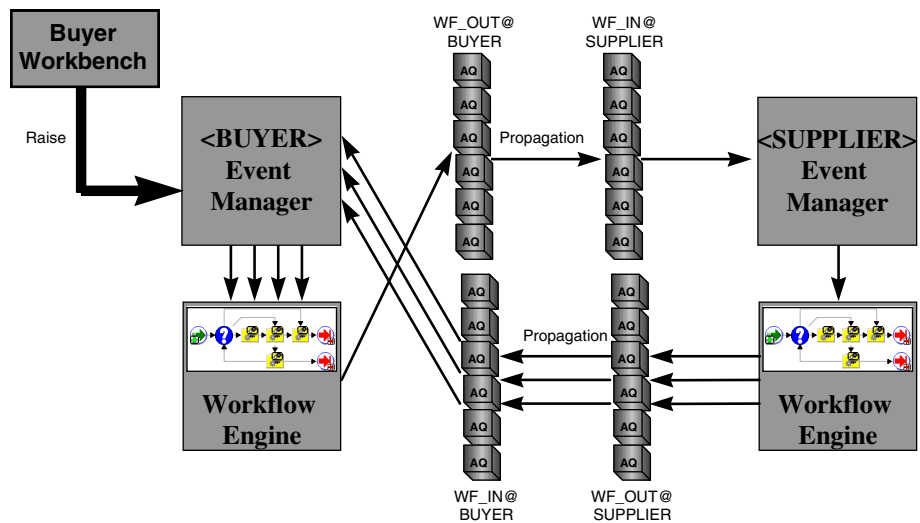
1. The Raise Event screen provides the simplest mechanism to raise business events without writing any PL/SQL code.
2. On the Raise Event screen, select the *oracle.apps.wf.event.wf.send* event as the Event Name.

3. Enter a unique Event Key.
4. Cut and paste the contents of any valid XML document into the Event Data field.
5. Submit the Event.
6. Go to the Find Processes screen and query for the Workflow Send Protocol process with an item key equal to the event key.
7. Verify that the workflow process shows that the message was sent, and that the workflow process is waiting on an acknowledgement message from the destination system (BES#2).
8. On the BES#1 Event Queue Summary, verify that the message has been placed on the WF_OUT queue.
9. When the message has a status of processed, go to BES#2 Event Queue Summary screen and confirm the message has arrived.
10. When the Agent Listener on BES#2 has processed the message, go to the Find Processes screen and query for the Workflow Send Protocol process with an item key equal to the event key.
11. This process should show that as there were no Agent Details defined on the external subscription for the *oracle.apps.wf.event.wf.send* event, the event message was not sent on to any further agents. As an acknowledgement message was required when the message was sent from BES#1, the workflow process should have automatically sent an *oracle.apps.wf.event.wf.ack* message back to BES#1.
12. The workflow process on BES#2 should now be complete.
13. Confirm the acknowledgement message has been sent by looking at the Event Queue Summary screens on BES#2 and BES#1.
14. When the *oracle.apps.wf.event.wf.ack* message on BES#1 has been processed by the Agent Listener, go to the Find Processes screen and query for the Workflow Send Protocol process with an item key equal to the event key.
15. The workflow process on BES#1 should now be complete.

Business Event System Demonstration

Oracle Workflow ships with a new demonstration, accessible from the Demonstrations screen from the Workflow Homepage. This demonstration can only be completed between at least two installations of Oracle Workflow 2.6, which have been signed up to each other.

The demonstration involves the entry of a Sales Order in one system. A Sales Order XML document is generated and transmitted to a second system. The second system parses the XML document to retrieve certain tag values, then transmits back to the originating system XML documents representing an Order Acknowledgement, an Advanced Shipment Notice, and an Invoice.



Oracle Advanced Queue propagation transmits event messages between the BUYER system and the SUPPLIER system.

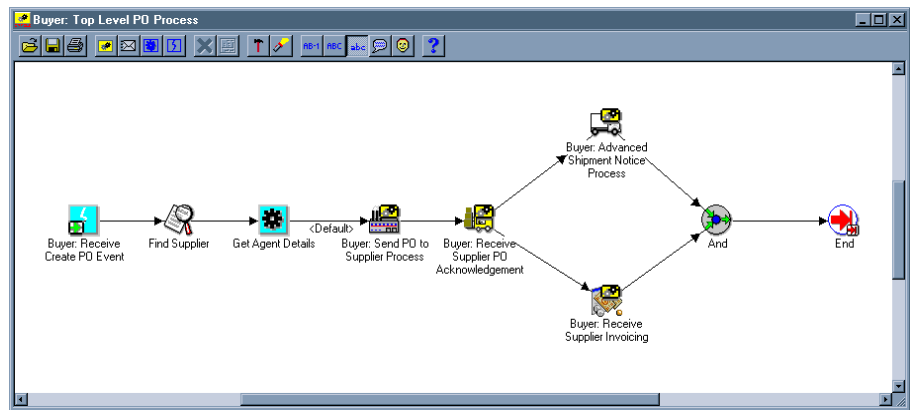
Setting Up the BUSINESS EVENT SYSTEM Demonstration between 2 Systems

1. Designate one of the systems to be BUYER and the other to be SUPPLIER.
2. On BUYER, edit the local subscription for the *demo.oracle.apps.wf.po.create* event. Select the WF_OUT agent on BUYER as the Out Agent, and WF_IN on SUPPLIER as the To Agent.
3. On SUPPLIER, edit the external subscription for the *demo.oracle.apps.wf.po.create* event. Select the WF_OUT agent on SUPPLIER as the Out Agent, and WF_IN on BUYER as the To Agent.
4. Ensure the Java Function Activity Agent is running on BUYER.
5. Ensure the Java Function Activity Agent is running on SUPPLIER.
6. Schedule the Workflow Background Engine database job to run every 10-30 seconds. The Workflow Background Engine can be run manually during the

demonstration by clicking on the Process Order hyperlink on the Track Order screen.

How to Perform the BUSINESS EVENT SYSTEM Demonstration

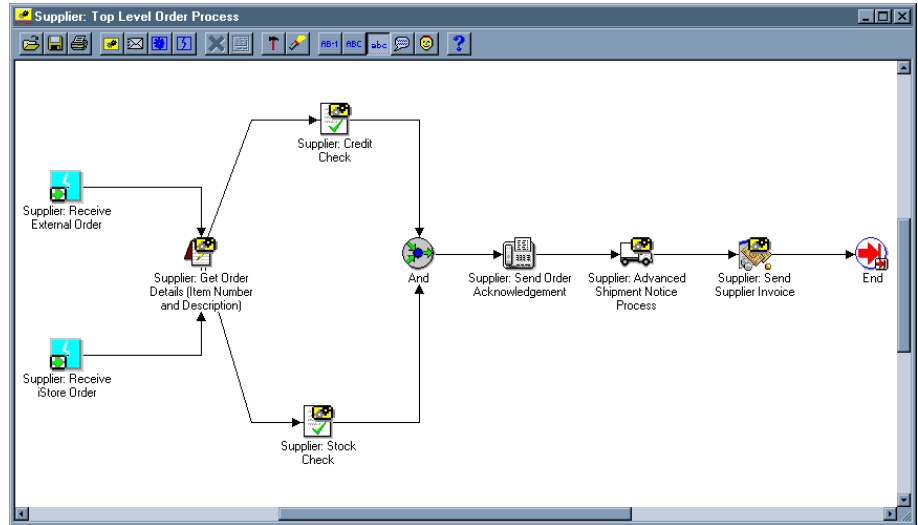
1. On BUYER, go to the Demonstration screen, and then to the Events: Enter Order screen.
2. Enter an Order on this screen. Fill in all details, ensuring that the Order number is unique, and that the date is entered in the format DD-MON-YYYY.
3. Press the Submit Button.
4. Submission causes two things to occur: the Sales Order information is inserted into a database table, and the *demo.oracle.apps.wf.po.create* is raised with the Sales Order number as the event key.
5. The Raise function looks to see if any subscriptions to this event require an XML Document to be generated. As there is a matching subscription that does require a message, the Generate Function for the *demo.oracle.apps.wf.po.create* event is executed, and a valid Sales Order XML document is created.
6. The Raise passes the event message to the Event Manager Dispatcher.
7. The Dispatcher looks for any active matching subscriptions to this event, which have a source type of Local.
8. The Dispatcher executes the local subscription for this event. The *workflow protocol* rule function run for this subscription only passes the event message to the workflow engine with the event key as the correlation ID. It does not send the event message to the out agent.
9. The Workflow Engine creates a new workflow process, with an item key equal to the Order Number.
10. The Track Order screen is displayed.
11. The GetXMLTag function in the Send PO to Supplier workflow process retrieves the Requestor Name. To enable the workflow process to continue, click on the Process Order hyperlink to run the background engine.
12. Click on the Workflow Monitor and view the diagram for this process. Note that the first activity in this process was a Receive Activity.



The Buyer: Top Level PO process sends the order event message to the supplier and waits to receive the supplier's responses.

13. The Find Supplier activity is a dummy activity, which does nothing.
14. The Get Agent function activity retrieves the Out Agent and To Agent values from the event subscription which launched the workflow.
15. The GetXMLTag function retrieved the Requestor Name from the XML Document. Click on the Attributes button in the bottom right hand corner and verify that the Requestor item attribute has been populated with the correct value.
16. The Event Message is then placed on the Out Agent addressed to the To Agent by the workflow send activity.
17. The Workflow Process will have stopped at the Receive Supplier PO Acknowledgement Activity, waiting on an acknowledgement message from SUPPLIER.
18. The WF_OUT queue should have a *demo.oracle.apps.wf.po.create* message addressed to WF_IN at SUPPLIER. Verify this using the Event Queue Summary screen.
19. When the *demo.oracle.apps.wf.po.create* message has a status of Processed, log into the SUPPLIER and confirm that a *demo.oracle.apps.wf.po.create* event message exists on the WF_IN queue.
20. When the Agent Listener process runs on this queue, it will dequeue this message, and pass the event message to the dispatcher.
21. The Dispatcher will look for any active matching subscriptions with a source type of external.
22. The Dispatcher executes the external subscription for the *demo.oracle.apps.po.create* event. The *workflow protocol* rule function run for this subscription only passes the event message to the workflow engine with the event key as the correlation ID. It does not send the event message to the out agent.

23. The Workflow Engine creates a new workflow process, with an item key equal to the Order Number.
24. On SUPPLIER, go the Demonstration screen, and then to the Track Order screen.
25. Select the Order Number entered in step 2.
26. Click on the Process Order hyperlink twice (to run the background engine twice).
27. Click on the Workflow Monitor hyperlink, and view the diagram.



The Supplier: Top Level Order process receives the order, processes it, and sends an acknowledgement, an advanced shipment notice, and an invoice back to the buyer.

28. Things to note about this workflow:
 - Two possible Receive start activities were possible depending on the business event.
 - The Supplier Get Order Details process retrieves the Item Number and Item Description from the Order XML Document. Click on the Attributes button to verify these values were retrieved.
 - The Supplier Credit Check process demonstrates the ability of a workflow process based on an XML tag value.
 - The Oracle Message Broker (OMB) Raise Event Activity only function as reminder to the demonstrator of the Oracle Workflow 2.6 integration with OMB.
 - Three event messages containing XML documents should have been sent back to BUYER from SUPPLIER: an acknowledgement, and advanced shipment notice and invoice.

29. Go the Event Queue Summary Screen and examine the out queue. Verify 3 messages with the same event key as the Order Number, addressed to BUYER. When these messages are processed, go to BUYER.
30. Verify the 3 messages have been received on the in queue. When these messages have been processed, go the Track Order screen and verify the workflow has been completed.

Configuring the Demonstration between 3 Systems - Adding in a Hub

This introduces a third system into this Demonstration.

More complex routing scenarios involving a HUB could be accomplished by using workflow processes to model the routing logic.

1. Designate the third system as HUB.
2. Sign up the HUB system to BUYER and SUPPLIER
3. Sign up BUYER to HUB.
4. Sign up SUPPLIER to HUB.
5. On BUYER, add the local subscription for the *demo.oracle.apps.wf.po.create* event. Select WF_IN on HUB as the To Agent.
6. On SUPPLIER, edit the external subscription for the *demo.oracle.apps.wf.po.create* event. Select the WF_OUT agent on SUPPLIER as the Out Agent, and WF_IN on the HUB as the To Agent.
7. On BUYER, delete the SUPPLIER agents and SUPPLIER system. BUYER does not have any information about the existence of SUPPLIER – the HUB will route any messages to the SUPPLIER.
8. On SUPPLIER, delete the BUYER agents and BUYER system. SUPPLIER does not have any information about the existence of BUYER – the HUB will route any messages to the BUYER.
9. On HUB, create an external subscription to the *demo.oracle.apps.wf.po.create* event. Enter WF_OUT on BUYER as the Source Agent for this subscription. Select the WF_OUT agent on HUB as the Out Agent, and WF_IN on SUPPLIER as the To Agent. The subscription should use *wf_rule.default_rule* as the rule function, with no workflow process details.
10. On HUB, create an external subscription to the *demo.oracle.apps.wf.po.ack* event. Enter WF_OUT on SUPPLIER as the Source Agent for this subscription. Select the WF_OUT agent on HUB as the Out Agent, and WF_IN on BUYER as the To Agent. The subscription should use *wf_rule.default_rule* as the rule function, with no workflow process details.
11. On HUB, create an external subscription to the *demo.oracle.apps.wf.po.asn* event. Enter WF_OUT on SUPPLIER as the Source Agent for this subscription. Select the WF_OUT agent on HUB as the Out Agent, and WF_IN on BUYER as the To Agent. The subscription should use *wf_rule.default_rule* as the rule function, with no workflow process details.

12. On HUB, create an external subscription to the *demo.oracle.apps.wf.po.invoice* event. Enter WF_OUT on SUPPLIER as the Source Agent for this subscription. Select the WF_OUT agent on HUB as the Out Agent, and WF_IN on BUYER as the To Agent. The subscription should use wf_rule.default_rule as the rule function, with no workflow process details.
13. Schedule Propagation between BUYER and HUB, HUB and BUYER, SUPPLIER and HUB, HUB and SUPPLIER.
14. The demonstration can now be performed, with all messages being routed through the HUB system.

Oracle Message Broker Integration

Oracle Workflow 2.6 ships with PL/SQL queue handler code designed to enqueue and dequeue from queues created by Oracle Message Broker with a defined abstract datatype. This enables Oracle Workflow 2.6 to support the communication protocols supported by Oracle Message Broker, including HTTP.

Set up

1. Use Oracle Message Broker to create at least 2 single consumer queues (one for inbound messages, and one for outbound messages) using the `OMBAQ_TEXT_MSG` abstract datatype in the workflow schema.
2. Create the OMB Queue Handler PL/SQL package by running the `wfquhdos.pls` and `wfquhdob.pls` scripts in the workflow schema. These scripts are located in the workflow sql directory
3. Create new Agents which map to the queues created in step 1 and which use the OMB queue handler.

Appendix A: Post Install Steps: Installing Oracle Workflow 2.6 in two schemas in the same database

1. Change the init.ora global_name parameter to be false (this means that the name of the database link does not have to be identical to the host it is connecting to).
2. Change the System name and display name on both installs to be unique, with neither being the same as the database name.
3. Ensure there are 2 different Database Access Descriptors (DAD) to connect to the different workflow installs. Enter each schema's username and password into the DADs so that anyone going to these URLs to access Oracle Workflow 2.6 is automatically pointing to the correct schema.
4. Run the wf/sql/wfsecoub.sql package in both schemas to install the workflow security package.



Getting Started with the Oracle Workflow 2.6 Business Event System
December 2000

Author: Mark Craig

Contributing Authors: George Buzsaki, Kevin Hudson, Clara Jaeckel,
George Kellner, David Lam, Jin Liu, Kenneth Ma, Steve Mayze,
Tim Roveda, Susan Stratton

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

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

Oracle Corporation provides the software
that powers the internet.

Oracle is a registered trademark of Oracle Corporation. Various
product and service names referenced herein may be trademarks
of Oracle Corporation. All other product and service names
mentioned may be trademarks of their respective owners.

Copyright © 2000 Oracle Corporation
All rights reserved.