

BPEL FOR WORKFLOW DEVELOPERS

Jerry Ireland, Rightsizing Inc

As a key enabler for SOA development, it will be essential to understand how to build Business Processes using BPEL (Business Process Language). BPEL has been a standard for describing business process for several years. Oracle has embraced it fairly recently as an important building block to their SOA Suite strategy and for Fusion. Most importantly it is a key part of the future E-Business Suite applications. E-Business Suite processes have been built on its own proprietary tool, Workflow. The use of Workflow has grown to the point where every module utilizes it in some way. With the latest versions, 11.5.10 and now 12, BPEL has been introduced in a few modules. Workflow outside of E-Business suite and Warehouse Builder will be desupported as of September. Because Fusion middleware includes BPEL as the glue that holds SOA together, its usage will grow over time. So how do the two workflow solutions compare? I'm going to assume a certain familiarity with Workflow and concentrate more on BPEL.

Major Differences

The major difference in the two business process tools is in their original design goals. Workflow was designed to manage a flow within a single database, and to communicate with people. The event system was added around 2001 to begin to address the need for communication outside the database. BPEL was designed to address the need to perform complex integration scenarios between disparate systems via web-services. Workflow is mostly PL/SQL based with portions in the database kernel. BPEL is primarily XML and java based and resides in the middle tier. BPEL is based on industry-wide standards while Workflow is more proprietary. The other major difference as of the last time I looked is that Workflow was free as part of the database and BPEL has a separate license fee.

What is BPEL?

So what is BPEL and how does it work? First of all BPEL makes use of XML and Web services; it uses an XML-based language that supports the Web services technology stack. To run BPEL code that has been generated, the code is parsed by a BPEL engine, which does the same kind of parsing job as other XML interpreters. Each process that's run is defined by a Web Services Description Language (WSDL) document, and its messages are transmitted across the Web by the Simple Object Access Protocol (SOAP). Processes that look up available Web services can use the Universal Description, Discovery and Integration (UDDI) directory.

To define how processes should be executed, BPEL has XML definitions or commands that specify the order of operations, the looping of operations, and synchronous and asynchronous requirements for operations. Synchronous operations block requesters until a request is either fulfilled or denied. Asynchronous operations allow requesters to continue without waiting for a response. BPEL also has commands to take care of fault conditions and commands to undo or reverse operations. These capabilities take XML beyond its traditional definitional role into the role of an executable language.

BPEL Process Manager Components

The tool set that helps to define orchestrate and manage business processes is currently referred to as BPEL Process Manager. It has several parts to it (see figure 1 below).

The **BPEL Designer** provides a graphical and user-friendly way to build BPEL processes. It uses BPEL as its native format. This means that processes built with the Designer are portable and in addition it enables developers to view and modify the BPEL source at any time.

The core **BPEL engine** provides an implementation of a BPEL server. The Oracle BPEL Process Manager executes standard BPEL processes and provides a “**dehydration**” capability so that the state of long-running flows is automatically maintained

in a database. The BPEL Server uses an underlying J2EE application server, with support for most major commercial application servers and a bundled version available.

The **built-in integration services** enable developers to easily access advanced connectivity and transformation capabilities from standard BPEL processes. These capabilities include support for XSLT and XQuery transformation as well as bindings to hundreds of legacy systems through JCA adapters and native protocols. Human Workflow, which was initially an important missing capability, is provided as a built-in BPEL service to enable the integration of people and manual tasks into BPEL flows. The JDeveloper BPEL Designer provides wizards to build these complex workflows and a simple GUI to map transformations.

The extensible **WSDL binding framework** enables connectivity to protocols and message formats other than SOAP. Bindings are available for Java, JMS, email, JCA, HTTP GET and POST and many other protocols enabling simple connectivity to hundreds of back-end systems. WSIF allows binding directly and transparently to any backend protocol or programming construct in order to get the benefits of a loosely-coupled Web services architecture with the performance and transactionality of native protocols.

The **BPEL Console** provides a mature web-based interface for management, administration and debugging of processes deployed to the BPEL server. Audit trails and process history/reporting information is automatically maintained and available both through the BPEL Console and via a Java API.

Most of the work is going on in the middle tier.

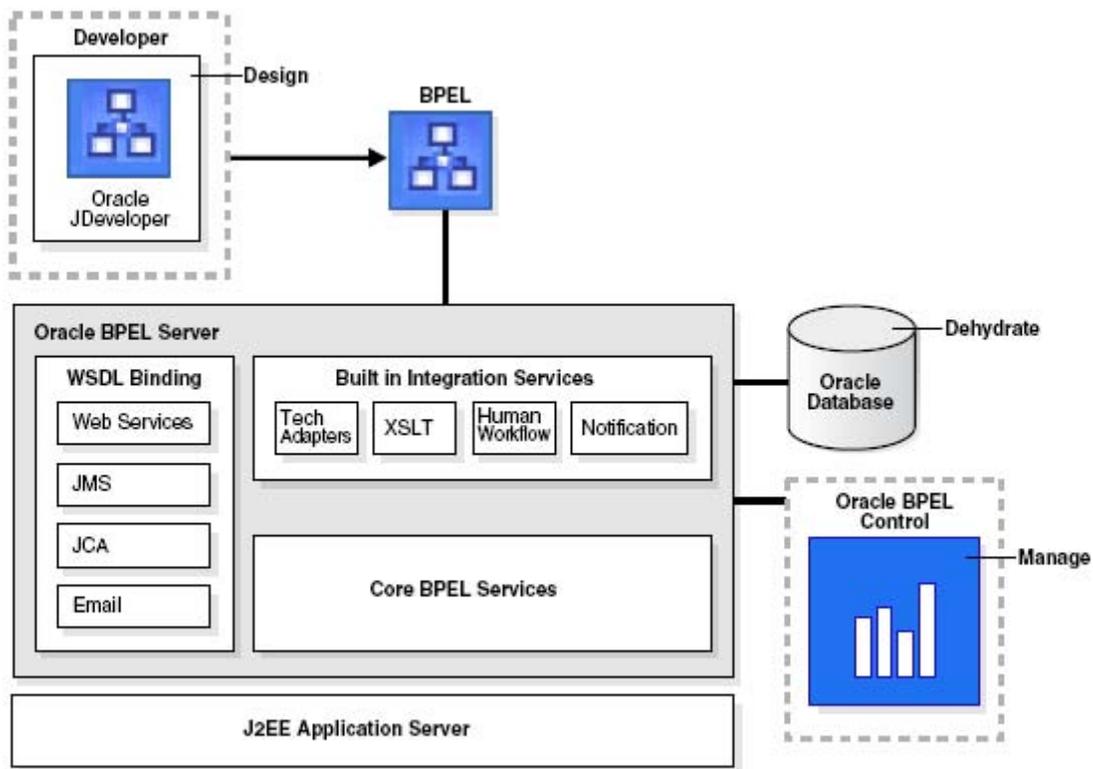


Figure 1: BPEL Process Manager Components

Workflow Components

Although the two tools have similar functionality for the components, Workflow, in contrast, has the majority of its components residing in the database server. Its designer is a client based tool with a proprietary file format that has to be

loaded into database tables for execution. The middle tier is utilized only as a base for email and access from the worklist, analysis and monitoring tools.

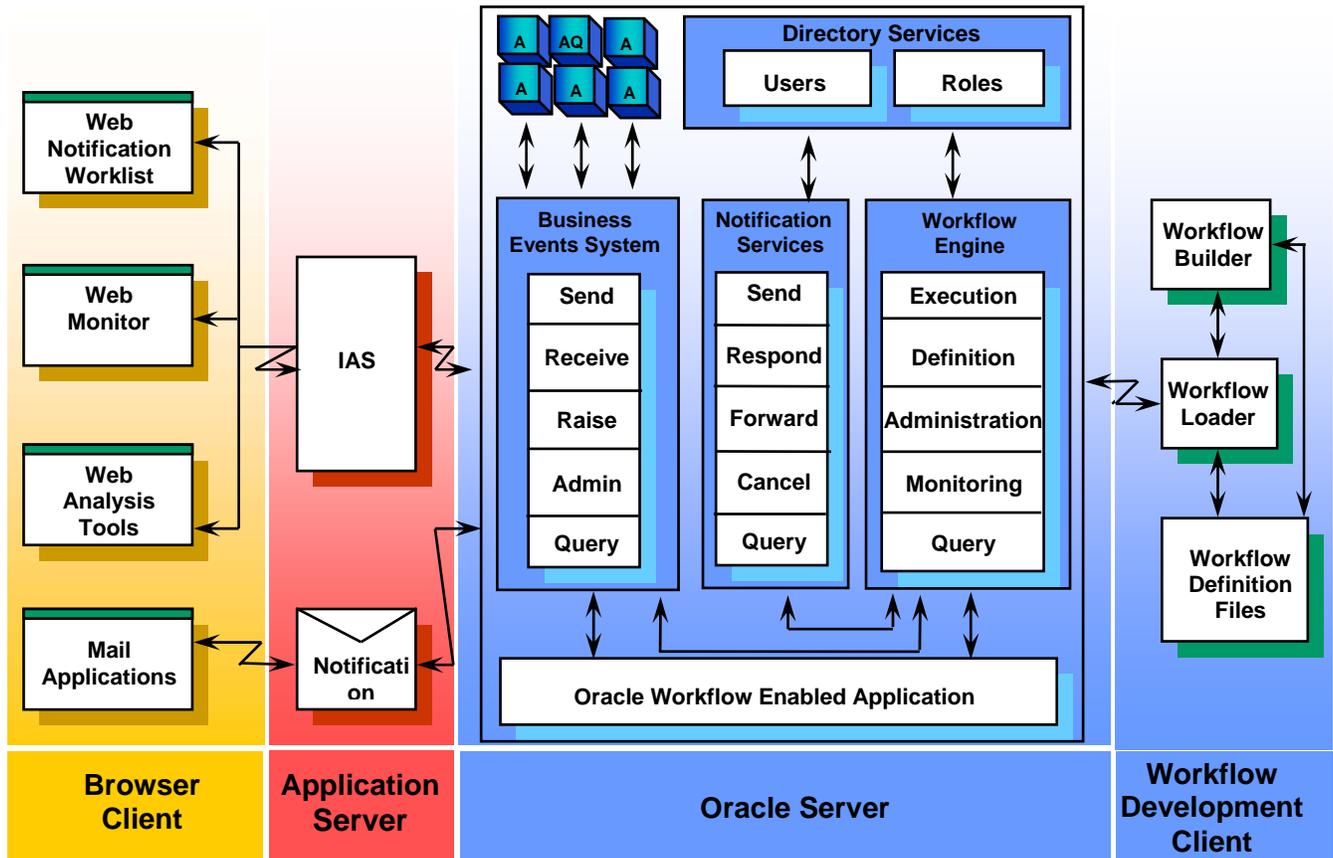


Figure 2: Workflow Components

BPEL Designer (JDeveloper)

The design environment (Oracle JDeveloper) provides a rich IDE environment where BPEL processes can be designed and deployed. BPEL processes can be designed by dragging and dropping elements (known as activities) into the process and editing their property pages. There are also a number of wizards that make some of the steps easier. BPEL processes can be integrated with external services that can also be designed and edited (known as partner links). There are also many technology adapters and services such as workflows, worklists, transformations, notifications, sensors, and business rules that can be integrated with the process. Processes are organized into Projects within Applications.

JDeveloper has many sections that allow you to perform various tasks. The figure below shows these sections. These sections stay in sync with each other to show different views of an object.

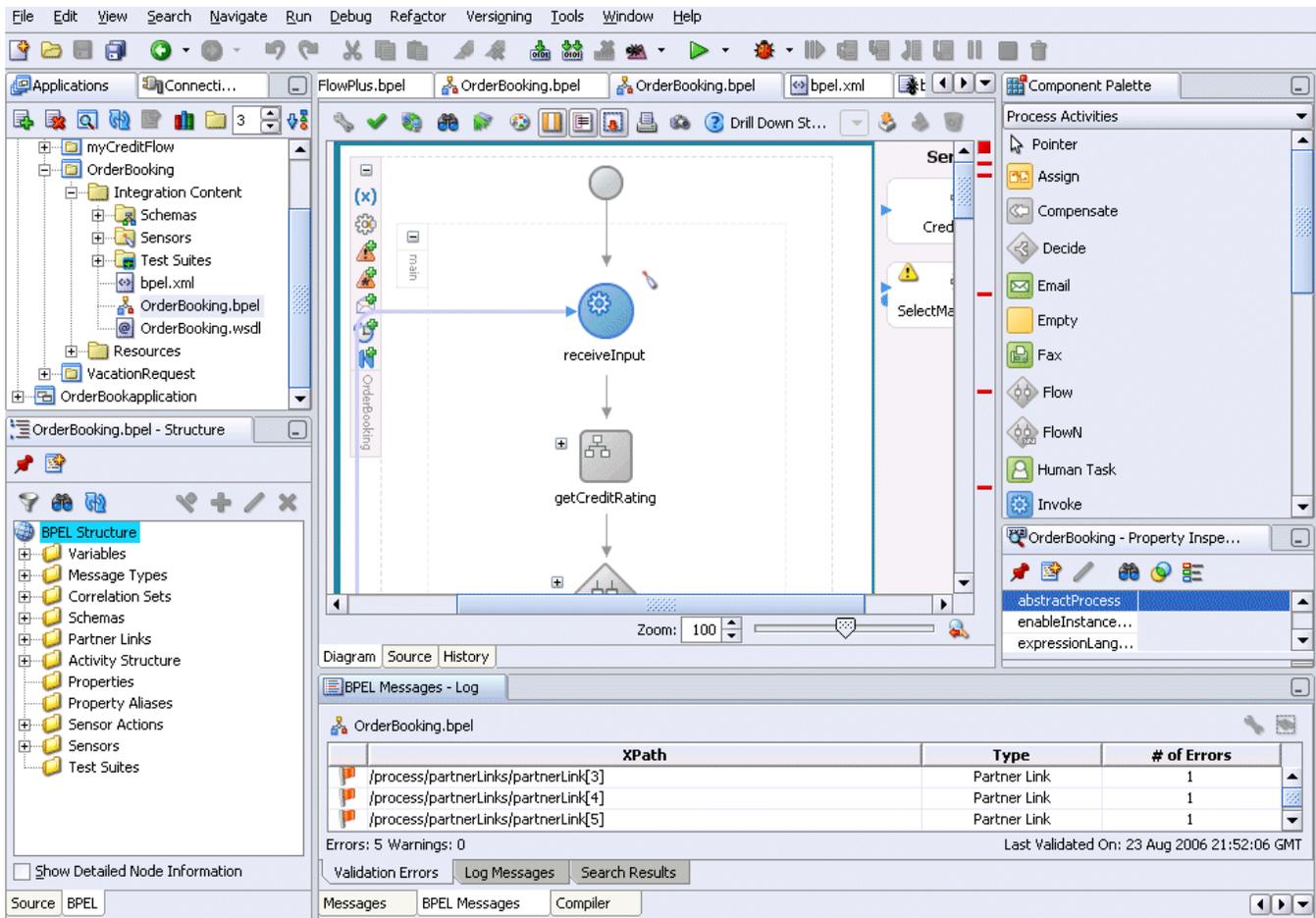


Figure 3: JDeveloper Sections

Application Navigator

This section displays the application and projects along with the various files that they contain. Double-clicking on any object will cause that object to be displayed in the other sections. Right-clicking an object shows a context-sensitive menu of commands relevant to the object.

Diagram Window

The Diagram Window provides a visual view of an object. The one above is a BPEL process. This window can be toggled back and forth between the visual view and the underlying source as they are always kept in sync no matter where changes are made. The window also has a handy History tab that compares two versions of the source in a side-by-side display.

Structure Window

The Structure Window displays a tree structured representation of the object that is displayed in the diagram window. It also will toggle to a source tab to stay in sync with the diagram window. It is particularly handy for finding a particular place in the source or diagram. Clicking on an object in the structure window will highlight that portion of the source or diagram in the diagram window.

Component Palette

The component palette window is a context sensitive list of objects that can be dragged and dropped onto the diagram window. The Process Activities palette (shown above) and the Services Palette are the building blocks of a BPEL process.

Property Inspector

The Property Inspector displays the details of an object. A single click on an object in either the Diagram Window or the Structure Window will cause that object to be displayed. Objects can also be modified using the Property Inspector and all the other views will be kept in sync.

Log Window

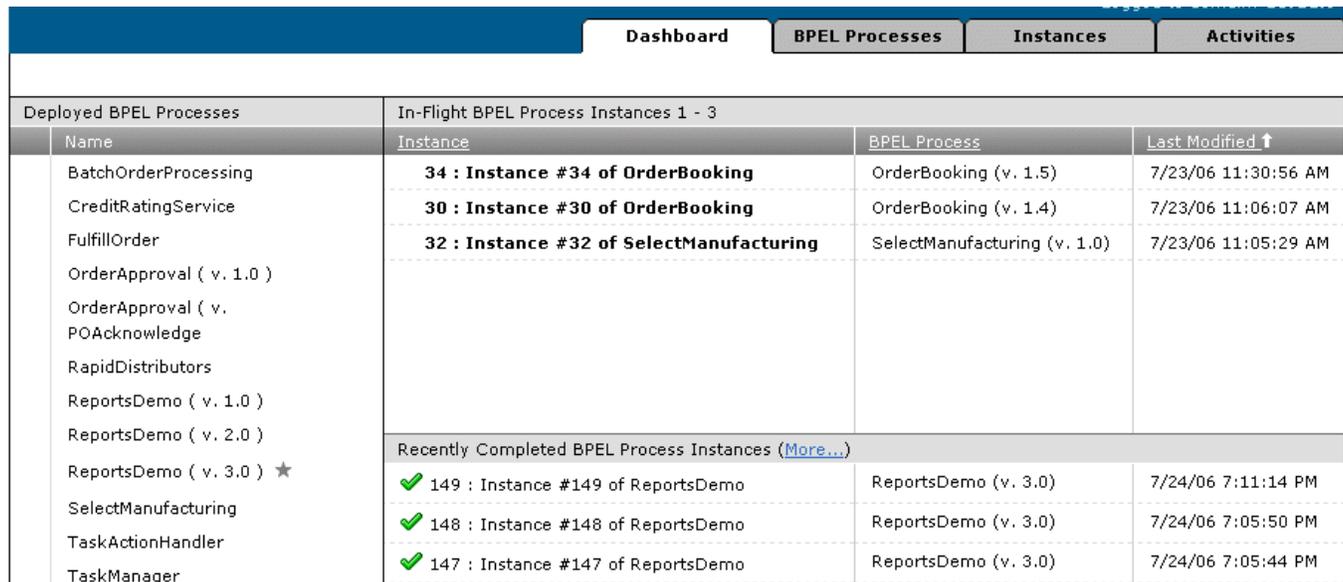
This window displays progress and errors from the various tasks the designer is asked to do. If errors are found during deployment of a process, double-clicking the error will navigate directly to the offending line in the source file referenced.

BPEL Server

After a process is completed it can be compiled and deployed to the BPEL Server. Deployment sends a set of files in a JAR file with a directory structure similar to the project directory structure to the Server.

BPEL Control

Oracle BPEL Control provides an interface to run, monitor, and administer BPEL processes.



The screenshot shows the BPEL Control Dashboard with four tabs: Dashboard, BPEL Processes, Instances, and Activities. The Dashboard tab is active, displaying a table of deployed BPEL processes and in-flight instances. The table is divided into two sections: 'In-Flight BPEL Process Instances 1 - 3' and 'Recently Completed BPEL Process Instances (More...)'. The 'In-Flight' section lists three instances: Instance #34 of OrderBooking, Instance #30 of OrderBooking, and Instance #32 of SelectManufacturing. The 'Recently Completed' section lists three instances of ReportsDemo, each with a green checkmark indicating completion.

Deployed BPEL Processes		In-Flight BPEL Process Instances 1 - 3		
Name	Instance	BPEL Process	Last Modified ↑	
BatchOrderProcessing	34 : Instance #34 of OrderBooking	OrderBooking (v. 1.5)	7/23/06 11:30:56 AM	
CreditRatingService	30 : Instance #30 of OrderBooking	OrderBooking (v. 1.4)	7/23/06 11:06:07 AM	
FulfillOrder	32 : Instance #32 of SelectManufacturing	SelectManufacturing (v. 1.0)	7/23/06 11:05:29 AM	
OrderApproval (v. 1.0)				
OrderApproval (v. POAcknowledge				
RapidDistributors				
ReportsDemo (v. 1.0)				
ReportsDemo (v. 2.0)				
ReportsDemo (v. 3.0) ★				
SelectManufacturing				
TaskActionHandler				
TaskManager				
Recently Completed BPEL Process Instances (More...)				
	✓ 149 : Instance #149 of ReportsDemo	ReportsDemo (v. 3.0)	7/24/06 7:11:14 PM	
	✓ 148 : Instance #148 of ReportsDemo	ReportsDemo (v. 3.0)	7/24/06 7:05:50 PM	
	✓ 147 : Instance #147 of ReportsDemo	ReportsDemo (v. 3.0)	7/24/06 7:05:44 PM	

Figure 4: BPEL Control - Dashboard

Dashboard

The figure above shows the Dashboard tab that appears when BPEL Control is started. It shows the currently deployed BPEL processes and instances of these processes that are currently running and that have recently completed. Click on a process in the Name column to access a page for creating an instance to test the process. Any currently running process can also be viewed.

BPEL Processes

The BPEL Processes tab displays BPEL process life cycles and states (Shown below). Notice that different versions of the same process can be running at the same time. The current default version is identified by an asterisk.

Dashboard		BPEL Processes	Instances	Activities																																																																											
Search Processes Process Name <input type="text"/> State Any <input type="button" value="v"/> Lifecycle Any <input type="button" value="v"/> <input type="button" value="Go"/>		Deployed Processes <table border="1"> <thead> <tr> <th>BPEL Process ↓</th> <th>Lifecycle</th> <th>State</th> <th>Open Instances</th> <th>Closed Instances</th> </tr> </thead> <tbody> <tr><td><input type="checkbox"/> BatchOrderProcessing (v. 1.0)</td><td>Active</td><td>On</td><td>0</td><td>2</td></tr> <tr><td><input type="checkbox"/> CreditRatingService (v. 1.0)</td><td>Active</td><td>On</td><td>0</td><td>9</td></tr> <tr><td><input type="checkbox"/> FulfillOrder (v. 1.0)</td><td>Active</td><td>On</td><td>0</td><td>0</td></tr> <tr><td><input type="checkbox"/> OrderApproval (v. 1.0)</td><td>Active</td><td>On</td><td>0</td><td>1</td></tr> <tr><td><input type="checkbox"/> OrderApproval (v. v2006_07_25__37316) ★</td><td>Active</td><td>On</td><td>0</td><td>0</td></tr> <tr><td><input type="checkbox"/> OrderBooking (v. 1.0)</td><td>Active</td><td>On</td><td>0</td><td>1</td></tr> <tr><td><input type="checkbox"/> OrderBooking (v. 1.1)</td><td>Active</td><td>On</td><td>0</td><td>2</td></tr> <tr><td><input type="checkbox"/> OrderBooking (v. 1.2)</td><td>Active</td><td>On</td><td>0</td><td>1</td></tr> <tr><td><input type="checkbox"/> OrderBooking (v. 1.3)</td><td>Active</td><td>On</td><td>0</td><td>1</td></tr> <tr><td><input type="checkbox"/> OrderBooking (v. 1.4)</td><td>Active</td><td>On</td><td>1</td><td>2</td></tr> <tr><td><input type="checkbox"/> OrderBooking (v. 1.5) ★</td><td>Active</td><td>On</td><td>1</td><td>0</td></tr> <tr><td><input type="checkbox"/> POAcknowledge (v. 1.0)</td><td>Active</td><td>On</td><td>0</td><td>4</td></tr> <tr><td><input type="checkbox"/> RapidDistributors (v. 1.0)</td><td>Active</td><td>On</td><td>0</td><td>8</td></tr> <tr><td><input type="checkbox"/> ReportsDemo (v. 1.0)</td><td>Active</td><td>On</td><td>0</td><td>3</td></tr> </tbody> </table>			BPEL Process ↓	Lifecycle	State	Open Instances	Closed Instances	<input type="checkbox"/> BatchOrderProcessing (v. 1.0)	Active	On	0	2	<input type="checkbox"/> CreditRatingService (v. 1.0)	Active	On	0	9	<input type="checkbox"/> FulfillOrder (v. 1.0)	Active	On	0	0	<input type="checkbox"/> OrderApproval (v. 1.0)	Active	On	0	1	<input type="checkbox"/> OrderApproval (v. v2006_07_25__37316) ★	Active	On	0	0	<input type="checkbox"/> OrderBooking (v. 1.0)	Active	On	0	1	<input type="checkbox"/> OrderBooking (v. 1.1)	Active	On	0	2	<input type="checkbox"/> OrderBooking (v. 1.2)	Active	On	0	1	<input type="checkbox"/> OrderBooking (v. 1.3)	Active	On	0	1	<input type="checkbox"/> OrderBooking (v. 1.4)	Active	On	1	2	<input type="checkbox"/> OrderBooking (v. 1.5) ★	Active	On	1	0	<input type="checkbox"/> POAcknowledge (v. 1.0)	Active	On	0	4	<input type="checkbox"/> RapidDistributors (v. 1.0)	Active	On	0	8	<input type="checkbox"/> ReportsDemo (v. 1.0)	Active	On	0	3
BPEL Process ↓	Lifecycle	State	Open Instances	Closed Instances																																																																											
<input type="checkbox"/> BatchOrderProcessing (v. 1.0)	Active	On	0	2																																																																											
<input type="checkbox"/> CreditRatingService (v. 1.0)	Active	On	0	9																																																																											
<input type="checkbox"/> FulfillOrder (v. 1.0)	Active	On	0	0																																																																											
<input type="checkbox"/> OrderApproval (v. 1.0)	Active	On	0	1																																																																											
<input type="checkbox"/> OrderApproval (v. v2006_07_25__37316) ★	Active	On	0	0																																																																											
<input type="checkbox"/> OrderBooking (v. 1.0)	Active	On	0	1																																																																											
<input type="checkbox"/> OrderBooking (v. 1.1)	Active	On	0	2																																																																											
<input type="checkbox"/> OrderBooking (v. 1.2)	Active	On	0	1																																																																											
<input type="checkbox"/> OrderBooking (v. 1.3)	Active	On	0	1																																																																											
<input type="checkbox"/> OrderBooking (v. 1.4)	Active	On	1	2																																																																											
<input type="checkbox"/> OrderBooking (v. 1.5) ★	Active	On	1	0																																																																											
<input type="checkbox"/> POAcknowledge (v. 1.0)	Active	On	0	4																																																																											
<input type="checkbox"/> RapidDistributors (v. 1.0)	Active	On	0	8																																																																											
<input type="checkbox"/> ReportsDemo (v. 1.0)	Active	On	0	3																																																																											
Related Tasks <ul style="list-style-type: none"> Clear WSDL Cache Deploy New Process Perform Manual Recovery Refresh Alarm Table View Process Log 																																																																															

Figure 5: BPEL Control – BPEL Processes

The list on the bottom left provide additional management capabilities for processes.

Instances

The third tab displays each individual instance of a process as shown below:

Dashboard		BPEL Processes	Instances	Activities																																							
Locate Instances Instance Id# <input type="text"/> Title <input type="text"/> Priority <input type="text"/> BPEL Process All Processes <input type="button" value="v"/> Creation Date All Times <input type="button" value="v"/> State All States <input type="button" value="v"/> Test Filter Show All <input type="button" value="v"/> <input type="button" value="Go"/>		List of BPEL Process Instances 1 - 20 <table border="1"> <thead> <tr> <th>Instance</th> <th>BPEL Process</th> <th>Last Modified ↑</th> </tr> </thead> <tbody> <tr><td>✓ 30 : Instance #30 of OrderBooking</td><td>OrderBooking (v. 1.4)</td><td>7/26/06 6:47:29 AM</td></tr> <tr><td>✓ 32 : Instance #32 of SelectManufacturing</td><td>SelectManufacturing (v. 1.0)</td><td>7/26/06 6:47:28 AM</td></tr> <tr><td>✓ 104 : Instance #104 of ReportsDemo</td><td>ReportsDemo (v. 2.0)</td><td>7/24/06 6:44:38 PM</td></tr> <tr><td>⚠ 38 : Instance #38 of POAcknowledge</td><td>POAcknowledge (v. 1.0)</td><td>7/23/06 11:30:56 AM</td></tr> <tr><td>34 : Instance #34 of OrderBooking</td><td>OrderBooking (v. 1.5)</td><td>7/23/06 11:30:56 AM</td></tr> <tr><td>✓ 36 : Instance #36 of SelectManufacturing</td><td>SelectManufacturing (v. 1.0)</td><td>7/23/06 11:30:55 AM</td></tr> <tr><td>✓ 37 : Instance #37 of RapidDistributors</td><td>RapidDistributors (v. 1.0)</td><td>7/23/06 11:30:54 AM</td></tr> <tr><td>✓ 35 : Instance #35 of CreditRatingService</td><td>CreditRatingService (v. 1.0)</td><td>7/23/06 11:30:15 AM</td></tr> <tr><td>✓ 33 : Instance #33 of RapidDistributors</td><td>RapidDistributors (v. 1.0)</td><td>7/23/06 11:06:06 AM</td></tr> <tr><td>✓ 31 : Instance #31 of CreditRatingService</td><td>CreditRatingService (v. 1.0)</td><td>7/23/06 11:05:27 AM</td></tr> <tr><td>✓ 29 : Instance #29 of BatchOrderProcessing</td><td>BatchOrderProcessing (v. 1.0)</td><td>7/23/06 11:05:27 AM</td></tr> <tr><td>⚠ 28 : Instance #28 of POAcknowledge</td><td>POAcknowledge (v. 1.0)</td><td>7/23/06 10:51:57 AM</td></tr> </tbody> </table>			Instance	BPEL Process	Last Modified ↑	✓ 30 : Instance #30 of OrderBooking	OrderBooking (v. 1.4)	7/26/06 6:47:29 AM	✓ 32 : Instance #32 of SelectManufacturing	SelectManufacturing (v. 1.0)	7/26/06 6:47:28 AM	✓ 104 : Instance #104 of ReportsDemo	ReportsDemo (v. 2.0)	7/24/06 6:44:38 PM	⚠ 38 : Instance #38 of POAcknowledge	POAcknowledge (v. 1.0)	7/23/06 11:30:56 AM	34 : Instance #34 of OrderBooking	OrderBooking (v. 1.5)	7/23/06 11:30:56 AM	✓ 36 : Instance #36 of SelectManufacturing	SelectManufacturing (v. 1.0)	7/23/06 11:30:55 AM	✓ 37 : Instance #37 of RapidDistributors	RapidDistributors (v. 1.0)	7/23/06 11:30:54 AM	✓ 35 : Instance #35 of CreditRatingService	CreditRatingService (v. 1.0)	7/23/06 11:30:15 AM	✓ 33 : Instance #33 of RapidDistributors	RapidDistributors (v. 1.0)	7/23/06 11:06:06 AM	✓ 31 : Instance #31 of CreditRatingService	CreditRatingService (v. 1.0)	7/23/06 11:05:27 AM	✓ 29 : Instance #29 of BatchOrderProcessing	BatchOrderProcessing (v. 1.0)	7/23/06 11:05:27 AM	⚠ 28 : Instance #28 of POAcknowledge	POAcknowledge (v. 1.0)	7/23/06 10:51:57 AM
Instance	BPEL Process	Last Modified ↑																																									
✓ 30 : Instance #30 of OrderBooking	OrderBooking (v. 1.4)	7/26/06 6:47:29 AM																																									
✓ 32 : Instance #32 of SelectManufacturing	SelectManufacturing (v. 1.0)	7/26/06 6:47:28 AM																																									
✓ 104 : Instance #104 of ReportsDemo	ReportsDemo (v. 2.0)	7/24/06 6:44:38 PM																																									
⚠ 38 : Instance #38 of POAcknowledge	POAcknowledge (v. 1.0)	7/23/06 11:30:56 AM																																									
34 : Instance #34 of OrderBooking	OrderBooking (v. 1.5)	7/23/06 11:30:56 AM																																									
✓ 36 : Instance #36 of SelectManufacturing	SelectManufacturing (v. 1.0)	7/23/06 11:30:55 AM																																									
✓ 37 : Instance #37 of RapidDistributors	RapidDistributors (v. 1.0)	7/23/06 11:30:54 AM																																									
✓ 35 : Instance #35 of CreditRatingService	CreditRatingService (v. 1.0)	7/23/06 11:30:15 AM																																									
✓ 33 : Instance #33 of RapidDistributors	RapidDistributors (v. 1.0)	7/23/06 11:06:06 AM																																									
✓ 31 : Instance #31 of CreditRatingService	CreditRatingService (v. 1.0)	7/23/06 11:05:27 AM																																									
✓ 29 : Instance #29 of BatchOrderProcessing	BatchOrderProcessing (v. 1.0)	7/23/06 11:05:27 AM																																									
⚠ 28 : Instance #28 of POAcknowledge	POAcknowledge (v. 1.0)	7/23/06 10:51:57 AM																																									

Figure 6: BPEL Control – Instances

By clicking on one of the instances a page will be displayed that will provide access to:

- Flow – visual representation of activity history showing the actual path taken by the current instance..

- Audit – view of the audit trail showing the actual activities that have been performed.
- Debug – matches source code with the current state in order to help to pinpoint the current place in the code.
- Interactions – displays the information from the Activities tab for this instance.
- Sensor Values – view of any results of activity, fault, or variable sensors.
- Test – save the current instance as a test case for import into JDeveloper.

Activities

The final tab shows the current state of the individual activities within the instances.

Dashboard BPEL Processes Instances Activities						
Locate Activities Activity Label <input type="text"/> Activity State All States v <input type="button" value="Go"/>	Activities 1 - 20					
	Activity Label	Instance	Process	Created	Due	
✓	assign (78)	2	CreditRatingService (v. 1.0)	7/22/06 7:18:48 PM	no date	
✓	receiveInput	2	CreditRatingService (v. 1.0)	7/22/06 7:18:48 PM	no date	
✓	replyOutput	2	CreditRatingService (v. 1.0)	7/22/06 7:18:48 PM	no date	
✓	assignCR	1	OrderBooking (v. 1.0)	7/22/06 7:18:48 PM	no date	
✓	assignRating	1	OrderBooking (v. 1.0)	7/22/06 7:18:48 PM	no date	
✓	assignOutput	1	OrderBooking (v. 1.0)	7/22/06 7:18:48 PM	no date	

Figure 7: BPEL Control – Activities

Process Overview

In order to show the differences in the two Process Modelers I will contrast the implementation of a single flow in the two systems. The flow comes from the E-business Suite world that I have used in the past to demonstrate how to build workflows. When a new employee is entered it is helpful to have a process validate that everything is set up correctly for an employee and notify the proper personnel if they are not. The BPEL version does not do everything it would need to do in order to interface with the E-Business Suite but provides the same basic steps. This flow actually highlights some of the strengths of workflow so it is a somewhat unfair comparison. I do this because this is the type of transition that will have to be made from existing Workflows to BPEL. But try getting workflow to communicate with a web service on a non-oracle system.

Workflow

The workflow version is a little more concise and readable, mostly because there is both an internal and a display name and the flows are easily labeled. The process below first goes to the database to look up and initialize some variables. It then checks to see if everything is in order for the employee. If everything is OK, the process ends, just waiting for the system administrator to be notified. If things are not OK, email is sent to HR to repair the setup and the process waits for the HR person to respond. If HR does not respond in a timely manner an email is sent to a supervisor with no wait for a response. A wait is introduced to prevent looping before the supervisor has had a chance to light a fire under someone. The loop at the bottom is just a way to limit the number of times the process sent something to the system administrator.

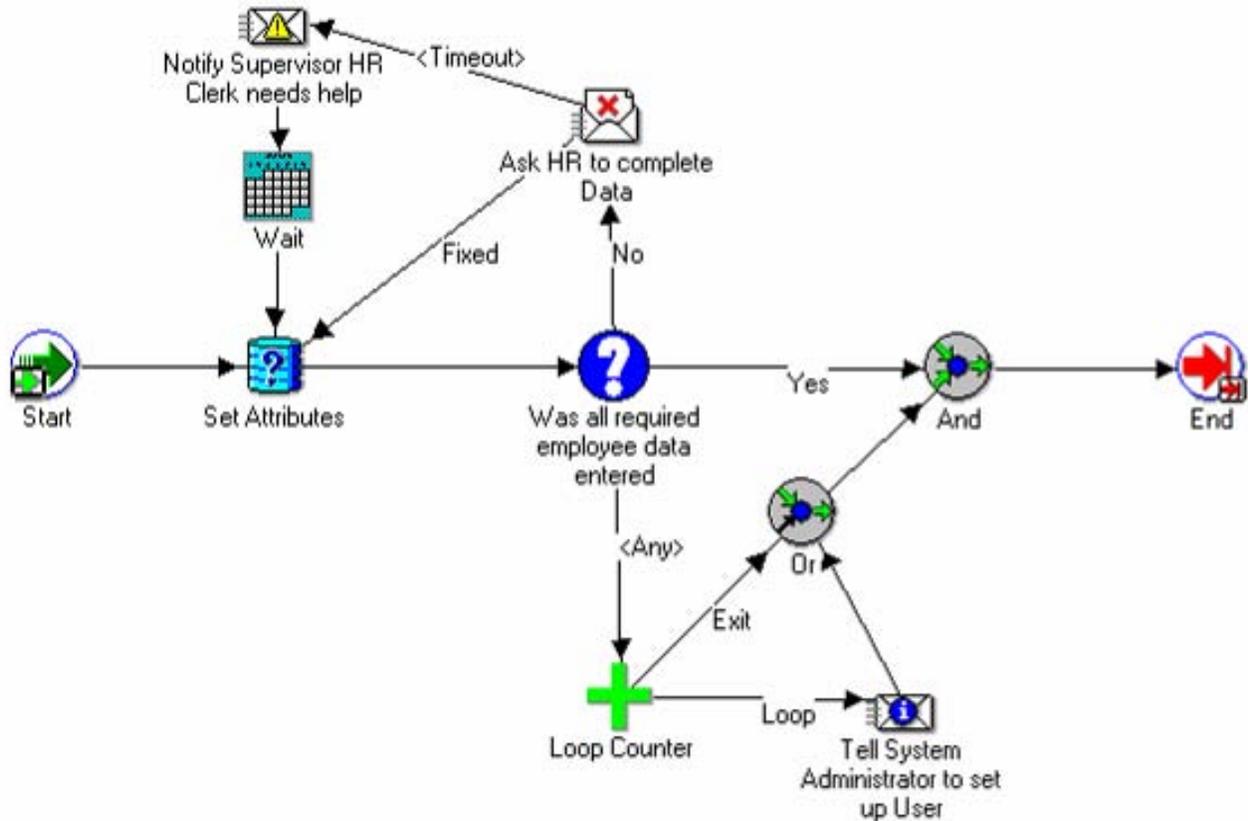


Figure 8: Workflow Diagram

BPEL

The BPEL process is laid out a little differently and we have to look at it in pieces. The swim lanes on either side contain services that are called by the process. On the left are a TaskService to do “Human Workflow” and a NotificationService to just send mail. On the right are two Services that interact with the database, one for initializing variables and the other for checking to see if everything is set up for the employee. The actual flow shows the basic steps. There is an initializeEmp scope that does some initial setup (see Figure 11 below for details). The next step is to send an email to the system administrator. The final step is a while loop that does most of the major work invoking the CheckEmp database adaptor and then performing a switch activity based on the return (See Figure 10 below for details).

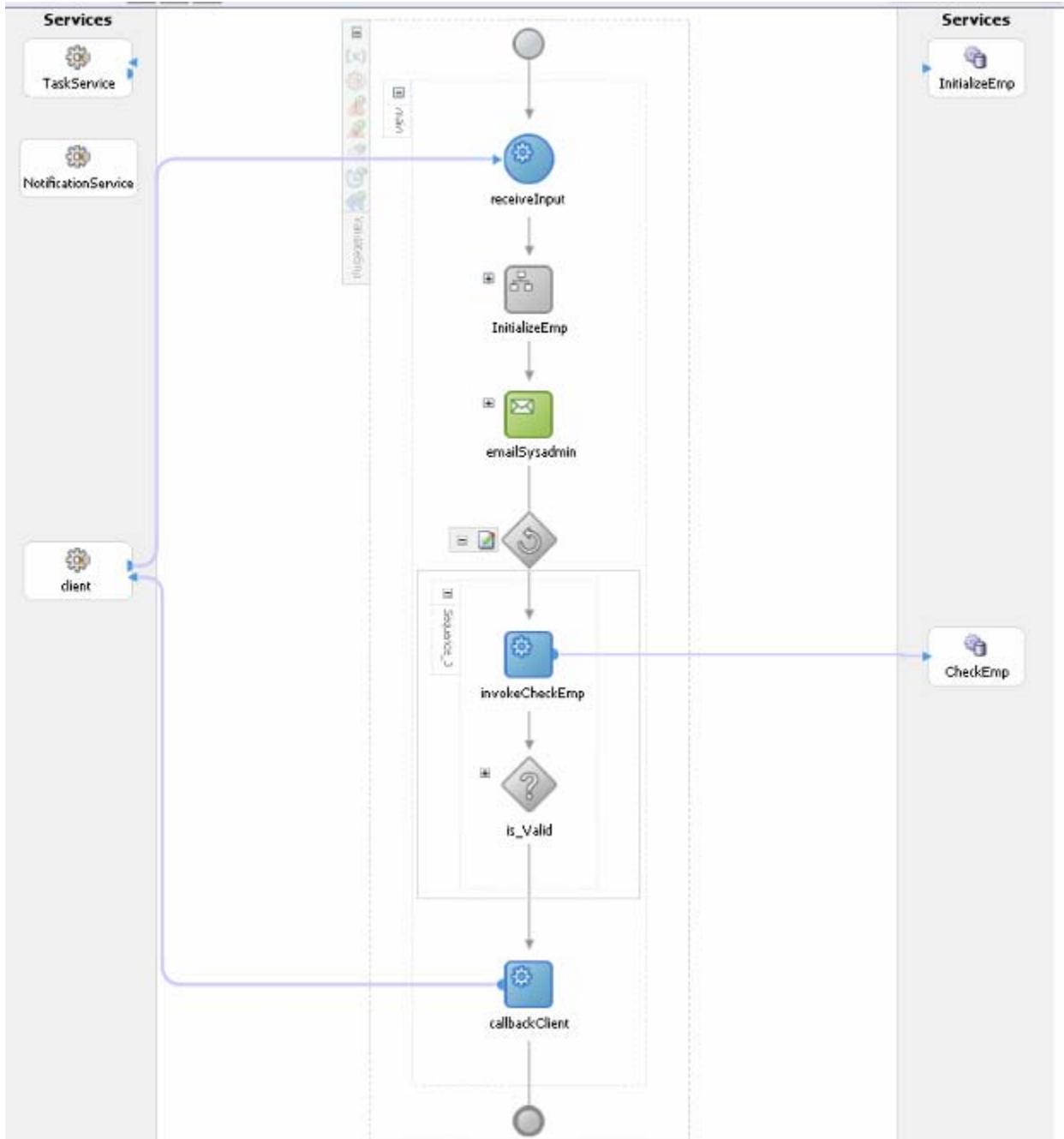


Figure 9: Main BPEL Flow

The expanded switch activity splits the flow based on a return from the invokeCheckEmp activity. If the return is VALID, an appropriate return value for the process is assigned and the process ends. If the employee is not valid, a notification is sent to HR that there is work to be done. This notification expects a response of OK when everything is ready to check again. If for some reason a response is not received, the other branch of the second switch activity is taken and a notification without response is sent to a manager and like the Workflow example a wait is introduced to allow work to be done before checking again.

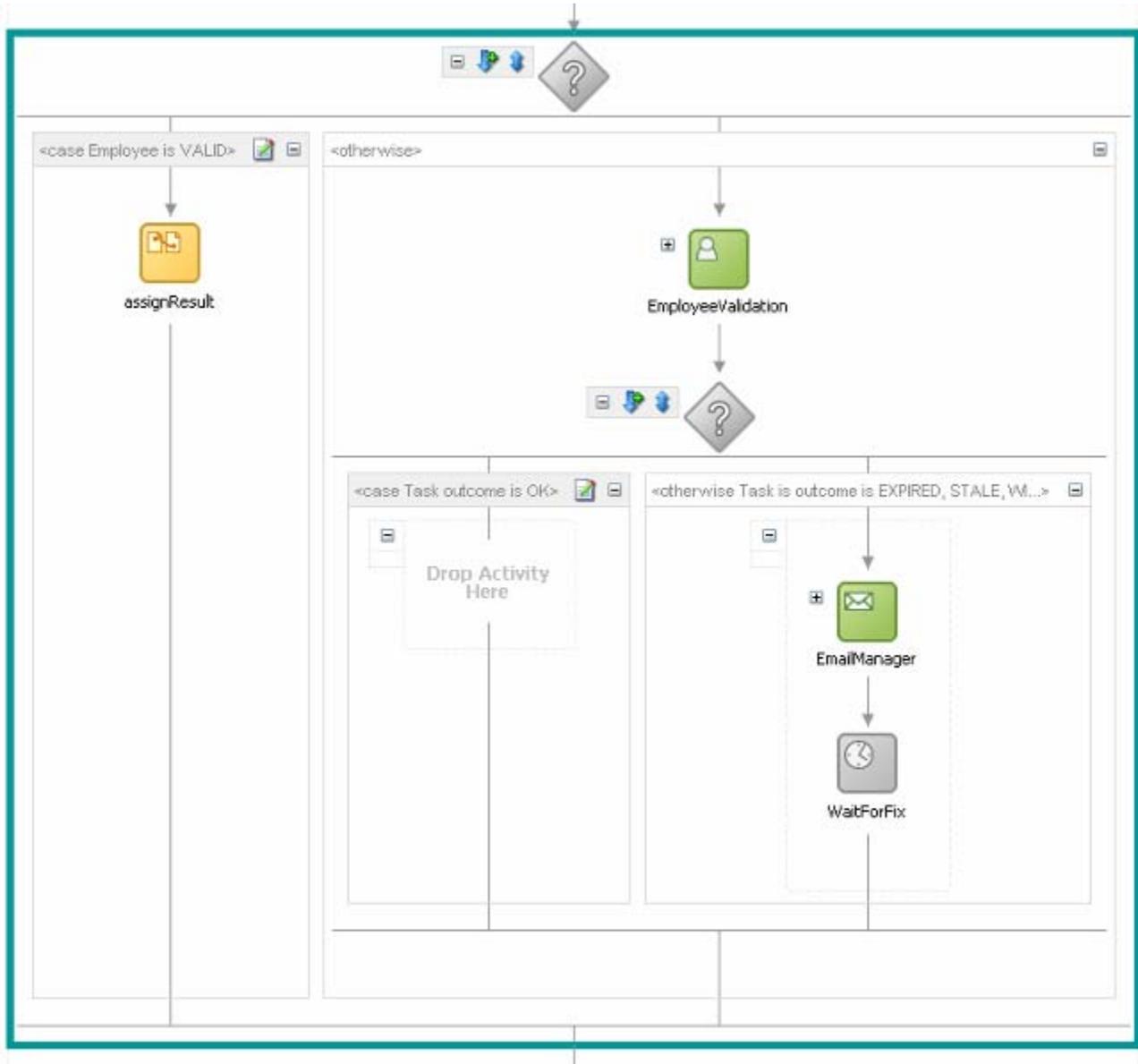


Figure 10: Expanded Switch Activity

The initializeEmp scope from above assigns some initial values to some internal variables and then invokes the InitializeEmp database adapter to retrieve additional information about the employee.

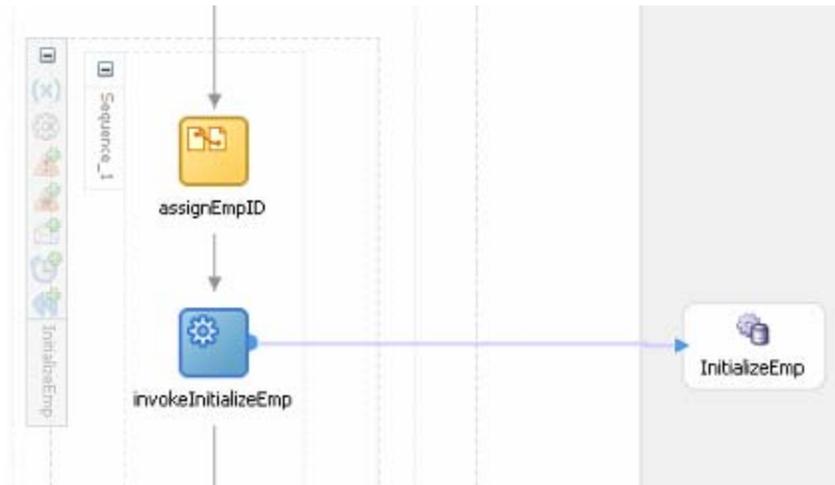


Figure 11: Expanded initializeEmp Scope

Process Details

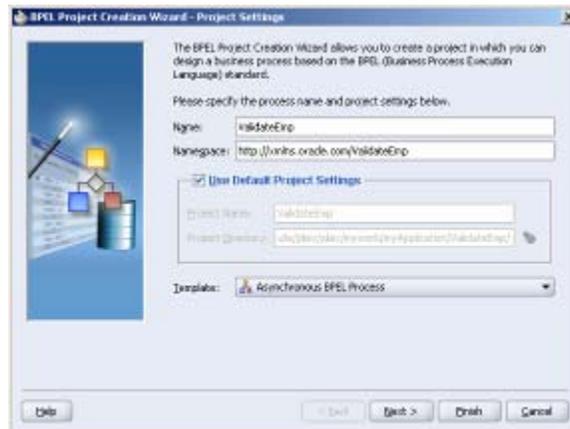
To get started with a BPEL process you need an application and a process to hold all the pieces.

Create an Application to contain the process from the Application Navigator.

1. Right-click **Applications** and choose **New Application**.
2. Type **myApplication** for a **Name**, accept defaults for the rest and click **OK**.
3. Click **Cancel** on the projects pop-up.

Create a BPEL project for a process.

1. Right-click **myApplication** and choose **New Project**.
2. Double-click **BPEL Process Project**.
3. Type **validateEmp** as the **Name** and click **Finish**.



- Your screen should now look like this after you expand everything in the **Application Navigator** and click on the diagram:

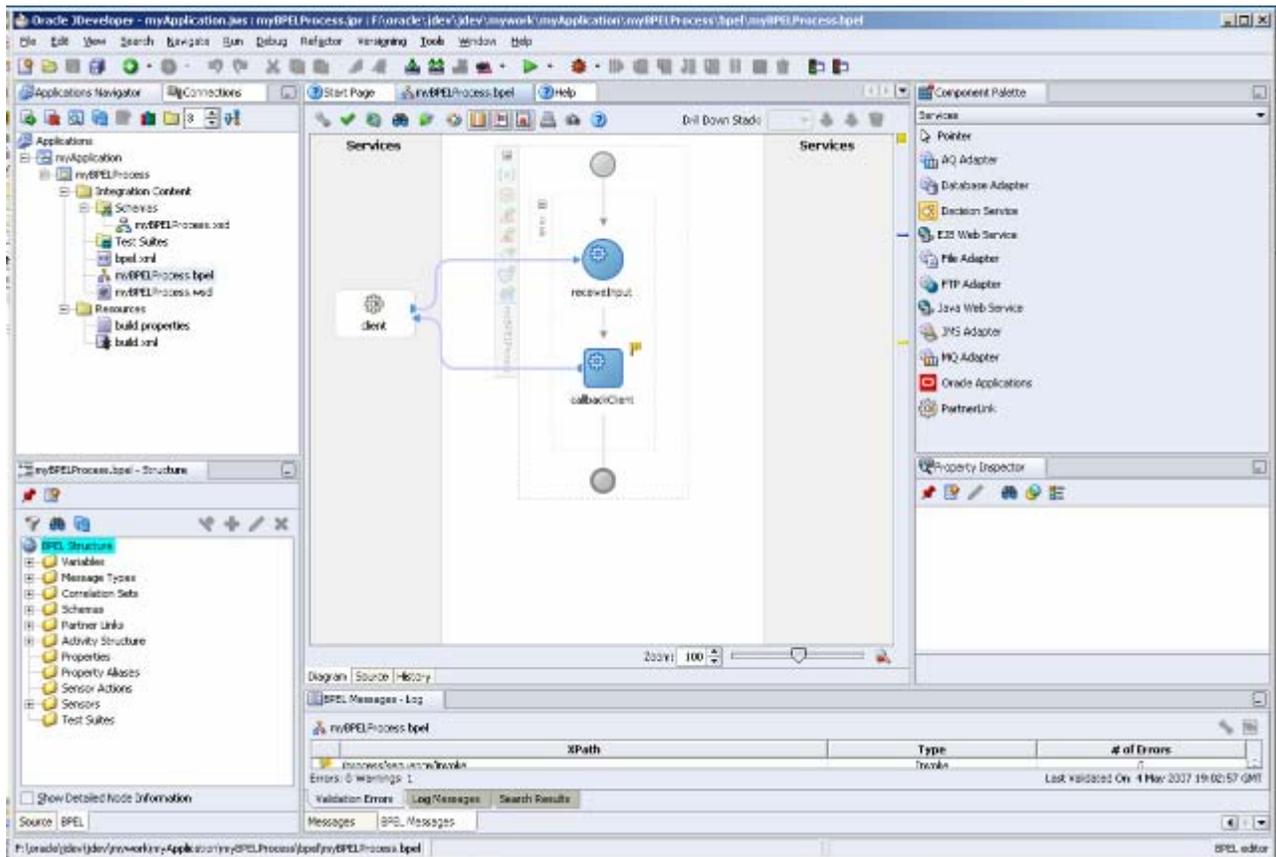


Figure 12: Initial BPEL Process

To get started with Workflow you need an Item Type and Process.

The screenshot shows the 'Workflow Quick Start Wizard' dialog box. It contains two sections: 'New Item Type' and 'New Process'. Each section has input fields for 'Internal Name' and 'Display Name', a dropdown for 'Persistence Type', and a text field for 'Number of Days'. The 'Internal Name' for the item type is 'XXEMP' and for the process is 'XXVALEMP'. The 'Display Name' for the item type is 'Validate New Employee' and for the process is 'Validate New Employee Data'. There are 'OK' and 'Cancel' buttons at the bottom.

Create Item Type and Process

- From the **File** menu choose **Quick Start Wizard**.
- Enter **XXEMP** as the **Internal Name**.
- Enter **Validate New Employee** as **Display Name**.
- Enter **XXVALEMP** as New Process **Internal Name**.
- Enter **Validate New Employee Data** as **Display Name**.

6. Your screen should look like this.

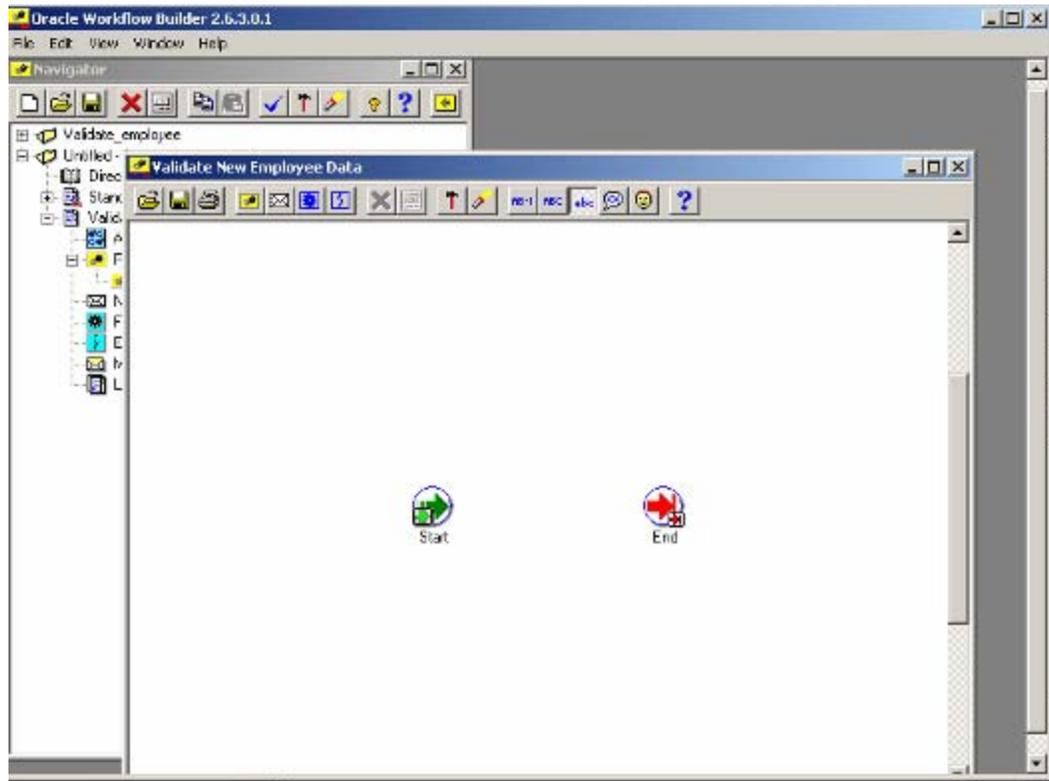


Figure 13: Initial Workflow Process

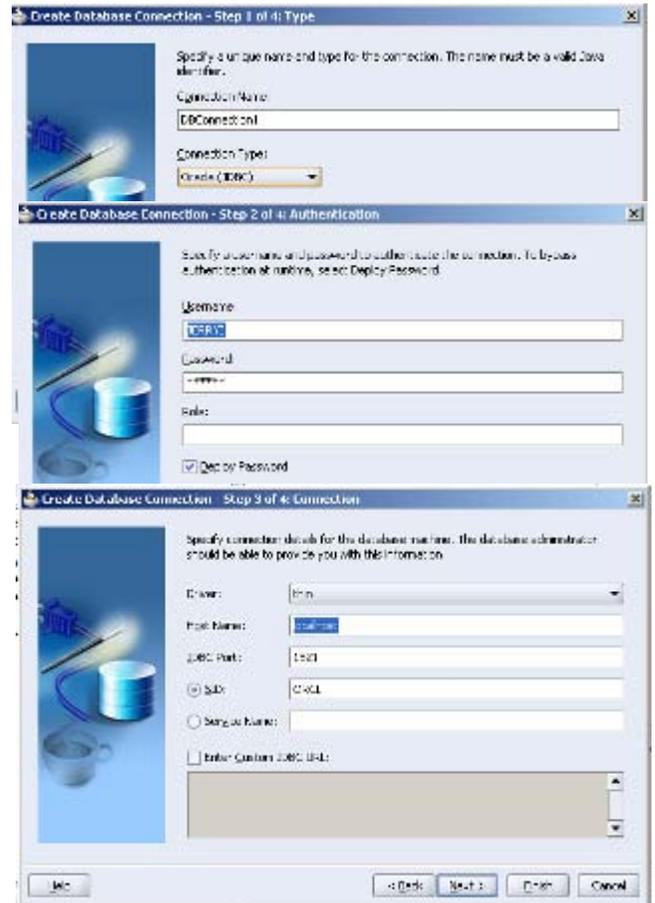
So far things are similar, but let's look at a couple of activities. We will look in more detail at the check employee call to the database and the step that notifies HR with an expected response.

Check the Employee for Valid Setup for BPEL

Add a Database Connection

BPEL needs to have a database connection defined in order to interact with a database.

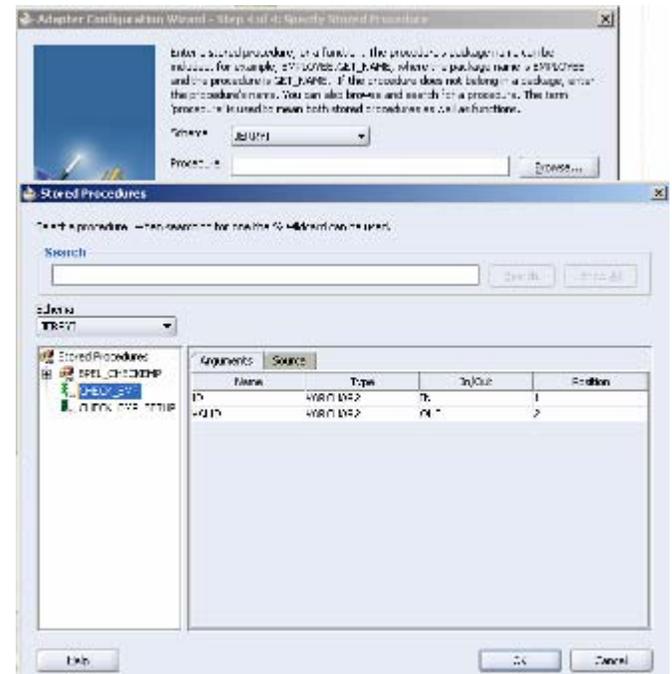
1. Switch to the **Connections** tab in the upper left window.
2. Right-Click on **Database** and choose **New Database Connection**.
3. Click **Next** on Welcome Page.
4. Enter **DBConnection1** for **Connection Name** and click **Next**.
5. Enter **Username** and **Password**, check **Deploy Password** and click **Next**.
6. Enter the appropriate **Host Name**, **JDBC Port** and **SID** for the database and click **Next**.
7. Click **Test Connection** to be sure everything is set up correctly and then click **Finish**.



Add Database Adaptor Service

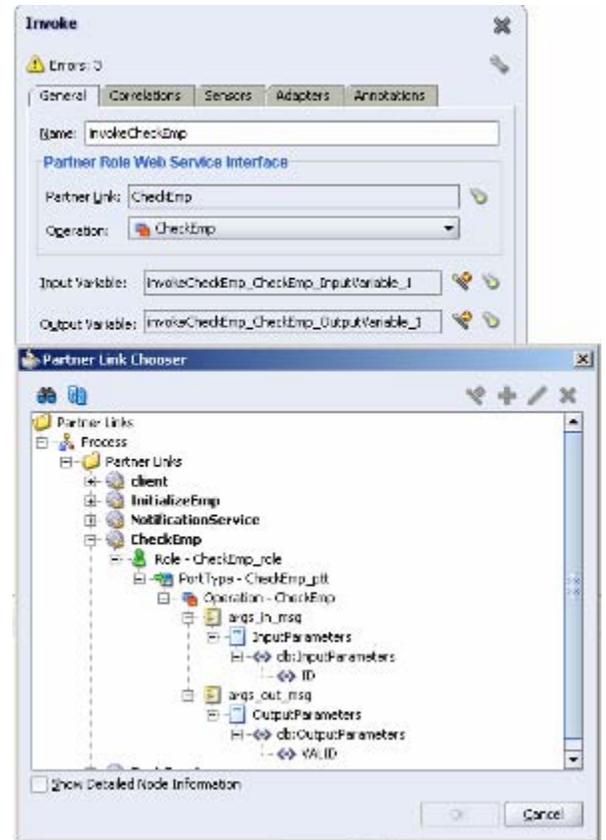
Define the interaction with the database.

1. In the **Component Palette**, switch to the **Services** selection.
2. Drag and drop the **Database Adaptor** into the right swim lane of the Diagram.
3. Click **Next** on the Welcome page.
4. Enter a **Service Name** of **CheckEmp** and click **Next**.
5. Select the **Database Connection** you just created above.
6. Leave the default of **Call a Stored Procedure or Function** and click **Next**.
7. Select **Schema JERRY1** from dropdown and click **Browse**.
8. Choose **CHECK_EMP** to display the in and out parameters and Click **OK**.
9. Click **Next** and then **Finish** to complete.



Add Invoke Activity to Diagram

1. In the **Component Palette**, switch to the **Process Activities** selection.
2. Drag and drop the **Invoke** activity onto the diagram.
3. Double-click the new activity.
4. Enter **invokeCheckEmp** as the **Name** and click the **flashlight** icon at the end of **Partner Link**.
5. Choose **CheckEmp** that you just created. It is expanded to show the details in the example.
6. Click on the **Create Variable** icon to the right of **Input** and **Output Variable** and accept the default variables.
7. Click **OK**.

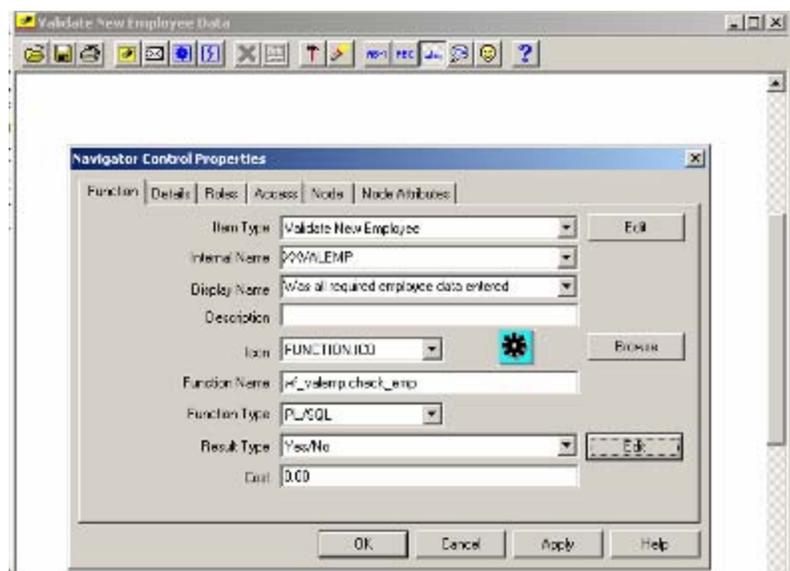


Check the Employee for Valid Setup for Workflow

Things are a little simpler for workflow. It is already connected to the database because that is where it runs. We just need to tell it what procedure to call. The procedure would have to be written differently to conform to Workflow.

Add Function to Call Procedure

1. Click the **Function** Icon at top of Diagram.
2. Click the diagram where you want the function to be.
3. Enter the following data in the pop-up:
 - a. Internal Name – XXVALEMP
 - b. Display Name - Was all required employee data entered
 - c. Function Name – wf_valemp.check_emp
 - d. Result Type – Yes/No



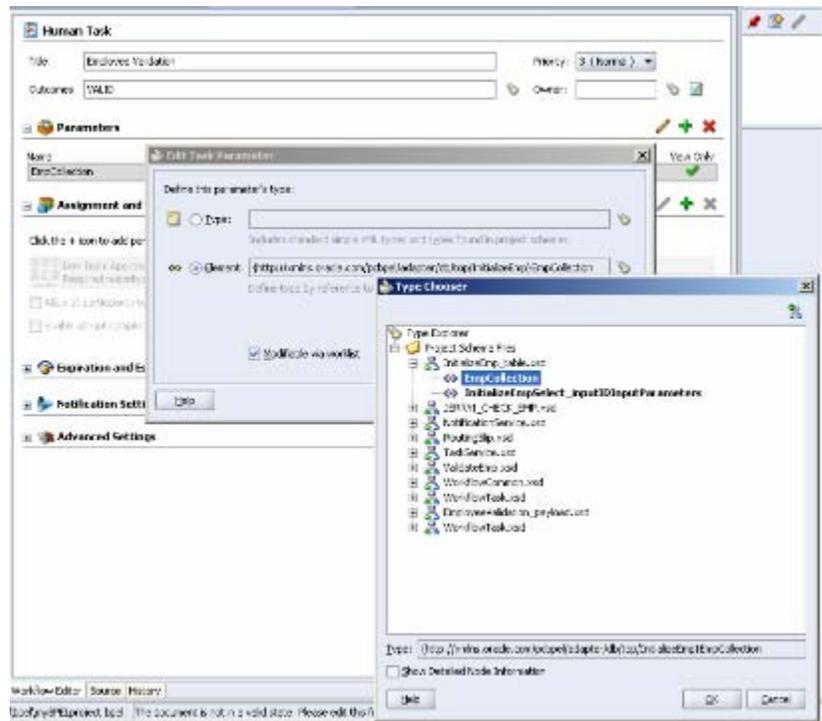
While the Workflow step is simpler to set up, the code in the package is more ridged in its format and is not clear what variables are being returned to the workflow process because it returns values by calling APIs that set the variables defined in the flow. BPEL on the other hand is very clear about what is returned and also accepts functions, custom SQL and allows for polling for changes in a table.

HR Validation Notification for BPEL

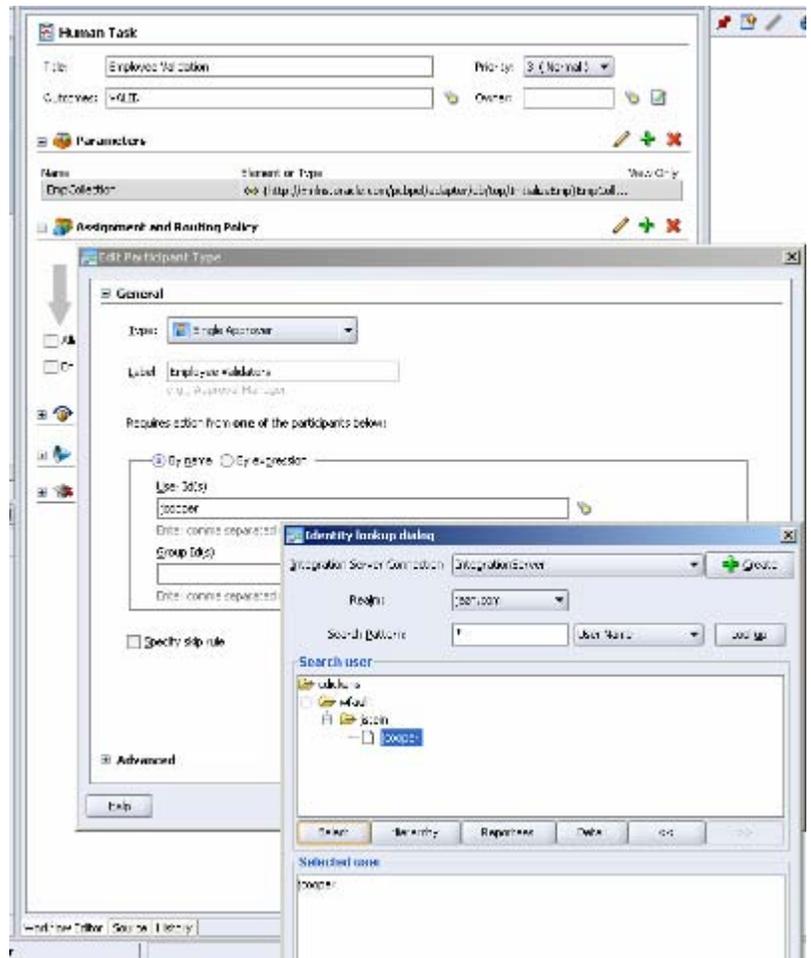
Adding a Human Task to the Employee Validation Process

Because we are going to use different outcomes than the standard default outcomes of APPROVE and REJECT, it is best to create the Human Task Definition first. This way the designer creates everything with the right values. If you try it in one step, some of the pieces get set up with the original default values and you have to go through a number of additional steps.

1. Right click on the project **ValidateEmp** and choose **Create Human Task Definition**.
2. Enter **EmployeeValidation** and click **OK**.
3. In the **Human Task Editor** enter **Employee Validation** for **Title**.
4. Click the **flashlight** icon next to the **Outcomes** field.
5. Uncheck **APPROVE** and **REJECT** and check **OK** and click **OK** button.
6. For **Parameters** click the + icon on the right side.
7. In the **Add Task Parameter** window click **Element** and then the **flashlight** icon.
8. In the **Type Chooser** window expand and select **Project Schema Files > InitializeEmp_table.xsd > EmpCollection**, and click **OK**. This is the schema that was defined as part of the **InitializeEmp** database adaptor.

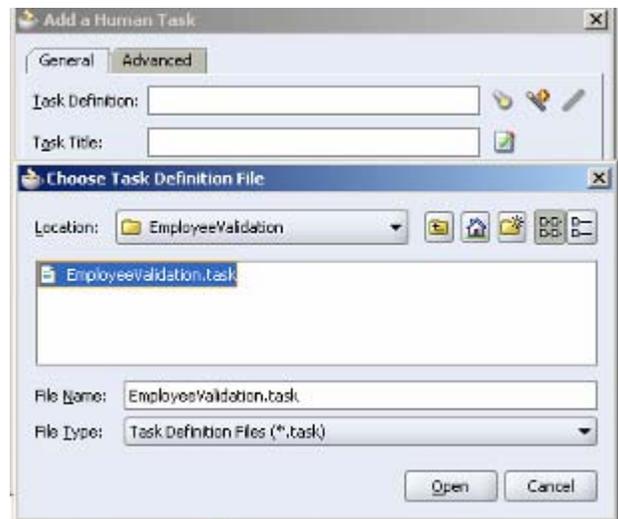


9. In the **Assignment and Routing Policy** section, click the + icon on the right side of the window.
10. In the **Add Participant Type** window, for **Type**, select **Single Approver**.
11. For **Label**, enter **Employee Validators**.
12. Click **By name** and then the **flashlight** icon to the right of **User Id(s)** to display the **Identity lookup dialog** window. Ensure that your integration server connection is selected.
13. Ensure that Oracle BPEL Server is running and click **Lookup** to see a list of all users currently in the system.
14. Select **jcooper** and click the hierarchy just to see what that looks like.
15. Click **jcooper** again to highlight it and click **Select**, and click **OK**.
16. Click the + icon next to **Expiration and Escalation Policy**.
17. Select **Expire after** from the drop down.
18. Enter a **1** in **Day** to have the notification expire after a day.
19. Select **Save** from the **File** main menu.
20. Click the **X** next to **EmployeeValidation.task** to close the **Human Task** editor.



Add Human Task to Diagram

1. Select the **Process Activities** from the **Component Palette** section.
2. Drag and drop a **Human Task** activity from the **Component Palette** on to the diagram.
3. Click the **flashlight** icon to the right of the **Task Definition** field.
4. Double-click the **bpel** folder
5. Double-click on the **EmployeeValidation** folder that was just created above.
6. Click on the **EmployeeValidation.task** entry.
7. Click **Open**.
8. Add the word **for** to the end of **Employee Validation** in **Task Title** and click the **XPath Expression Builder** icon to the right.



9. In the **BPEL Variables** section, expand and select **invokeInitialize_InitializeEmpSelect_inputID_OutputVariable > EmpCollection > ns2.EmpCollection > ns2.Emp > ns2:EmpName** and click **Insert Into Expression**.
10. Click **OK**.
11. Click the **flashlight** icon in the **BPEL Variable** column.
12. In the **Task Parameters** window, expand and select Variables > **invokeInitialize_InitializeEmpSelect_inputID_OutputVariable > EmpCollection > ns2.EmpCollection** and click **OK**.
13. Click **OK** in the **Human Task** window.

Creating a Task Form for the Worklist

1. In the **Application Navigator**, right-click the **EmployeeValidation** folder that contains the **EmployeeValidation.task** file and select **Auto Generate Simple Task Form**.
2. Once the autogenerated task form appears close **payload-body.jsp** by clicking the **X** sign on the tab.

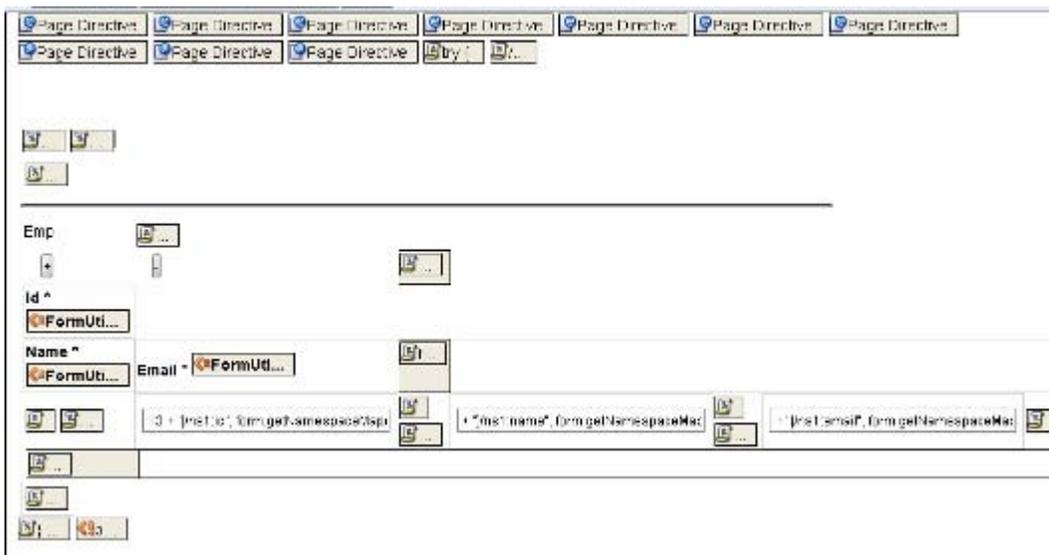


Figure 14: Tasklist Autogenerated Form

The result of all this is the creation of many objects within the process. Just take a look at the number of additional objects in the Application Navigator. Also expand the EmployeeValidation task and the Switch activity following it on the diagram. Fortunately, most of these objects can be manipulated using the Human Task Editor. The one thing that does not edit gracefully is the Outcomes field. The switch activity that follows the email task does not get changed with the new values.

HR Validation Notification for Workflow

Create a Lookup Type

The notification is going to return a value that is not currently defined as a Lookup Type so a new one has to be created.

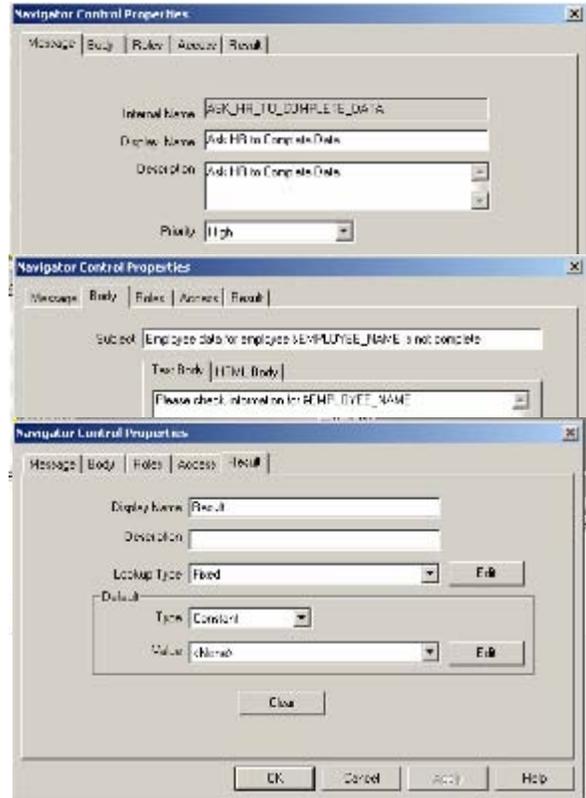
1. Right-click the **Lookup Type** entry in the tree and select **New Lookup Type**.
2. Enter an **Internal Name** of **FIXED** and a **Display Name** of **Fixed**.
3. Click **OK**.

4. Right-click the resulting **Fixed Lookup Type** and select **New Lookup Code**.
5. Enter an **Internal Name** of **FIXED** and a **Display Name** of **Fixed**.
6. Click **OK**.

Create Message

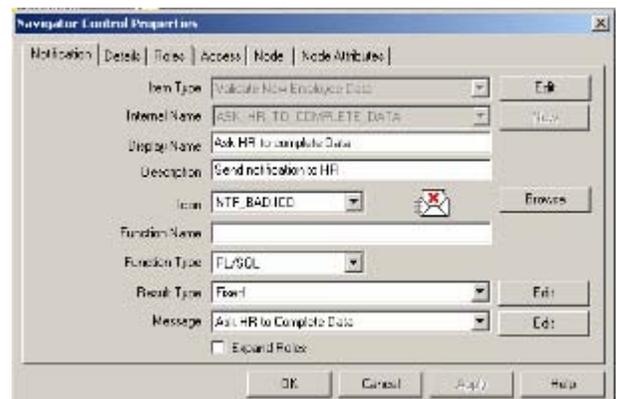
A message for the Notification has to be created separately.

1. Right-click the **Message** entry in the tree and select **New Message**.
2. Enter **HRALERT** as the **Internal Name** and **Ask HR to Complete Data** as the **Display Name**.
3. Click the **Body** tab
4. Enter Employee **Data for employee &EMPLOYEE_NAME is not complete** in the **Subject** line. The &attribute_name substitutes the value of the attribute.
5. Enter **Please check information for &EMPLOYEE_NAME** in the **Text Body**.
6. Click the **Result** tab.
7. Enter **Result** in the **Display Name**.
8. Choose **Fixed** from the drop-down list in the **Lookup Type** field.
9. Click **OK**.



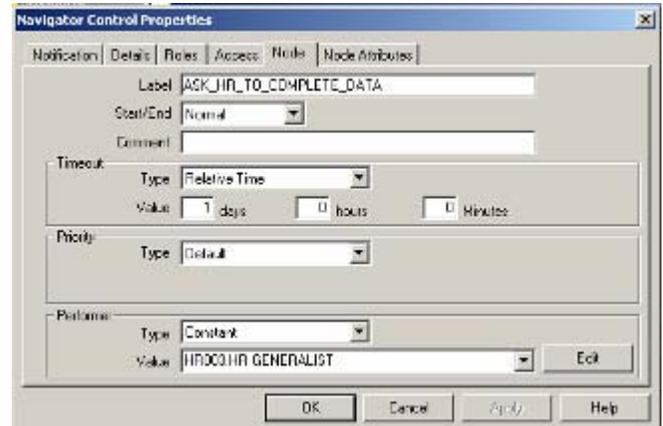
Create the Notification on the Diagram

1. Click the **New Notification** icon at the top of the diagram.
2. Click on the diagram where you want it placed.
3. Enter the following data into the form:
 - a. Internal Name - ASK_HR_TO_COMPLETE_DATA
 - b. Display Name - Ask HR to complete Data
 - c. Description - Send notification to HR
 - d. Result Type – Fixed
 - e. Message – Ask HR to Complete Data
4. Click **OK**.



5. Click on **Node** tab.
6. For **Performer Value** enter HR003.HR GENERALIST
7. Choose Relative Time from Timeout Type drop-down list.
8. Enter 1 in days field.

Again Workflow provides a much simpler set up for a notification with response. A notification without a response is actually simpler in BPEL because the Notification and Message are not separate pieces. But BPEL provides many more automatic capabilities that would have to be built into the logic of a Workflow process. It can automatically escalate a notification up a management chain based on a time out setting. It can automatically send reminders if an item hasn't been responded to yet. And there are many more options.



Validating, Compiling, and Deploying the Employee Validation Process in BPEL

1. Right-click the **ValidateEmployee** process in the **Application Navigator**
2. Select **Deploy > BPEL Process Deployer** and click **OK**.
3. Check for and repair errors.

Running the Employee Validation Process

1. Log into the **Oracle BPEL Control** by selecting it from the Start menu.
2. When the **Dashboard** tab appears, click on the **ValidateEmployee** from the **Deployed BPEL Process** list.
3. Enter an **ID** for an Employee and click **Post XML Message**.

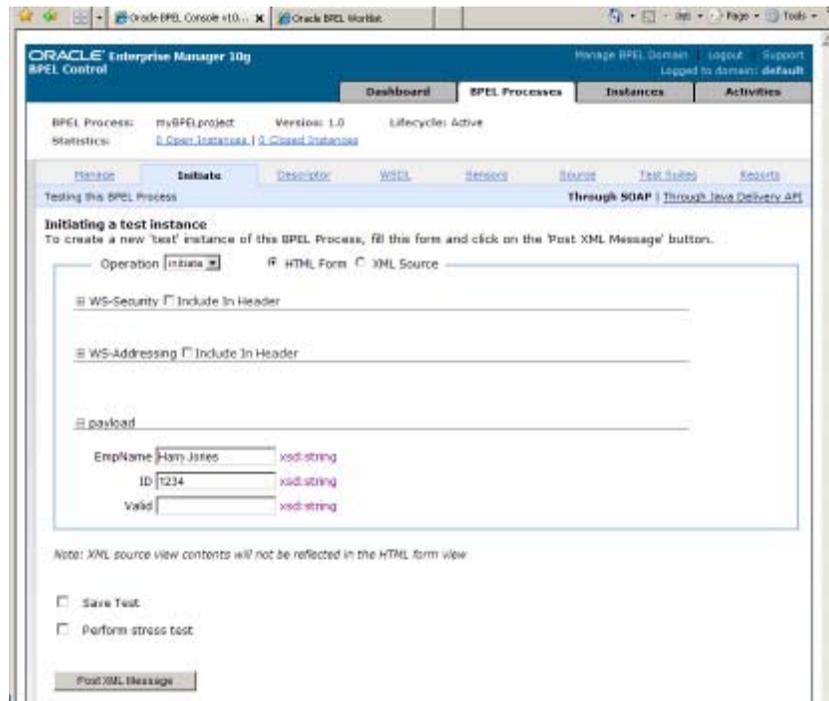


Figure 15: Run Employee Validation

- When the **Instance Initiated** page is displayed click on the **Flow** icon to see where the process is.
- Expand the **EmployeeValidation** activity to see the detail.

You can see that the process has progressed to the point of waiting for a response from the validators. You can click on any of the nodes to view the underlying code. Clicking on the Audit tab provides a clearer picture of where the process is.

Figure 16: Process Progress

- In order to respond to the request start the **Oracle BPEL Worklist** and log in as jcooper/welcome1.
- The worklist is displayed. Select **Claim** in the actions list of the task to validate and click **Go**. This is the way that someone can claim an item on the worklist.

Task Number	Title	Priority	Assigned Users	Assigned Groups	State	Created Date	Expiration Date	Actions
10001	Employee Validation for Harry Jones	3		Supervisor	Assigned	May 5, 2007 1:00 AM		Claim

Figure 17: Task List

8. The details of the item are displayed along with a short history of what has happened to the item.
9. Select the response **OK** in the **Task Action Field** and click **Go**.
10. The task list is redisplayed to act on the next task.
11. Go back to the BPEL Control page and re-query the process instance. It has now progressed.

Validating, Compiling, and Deploying the Employee Validation Process in Workflow

In order to deploy a workflow process it has to be loaded into the database.

1. Select **Save as** from the **File** menu.
2. Click **Database**.
3. Enter the APPS user, password and SID.
4. Click **OK**.
5. Any errors in logic will be reported and must be corrected before the process is loaded.
6. The workflow process is then launched from a program via the Workflow API.

Additional Observations

Both tools have a detailed log of activities that are processed stored in a database. Both require vigilant purging of the tables because they can grow very quickly and effect performance. Both have extensive error logging capabilities and methods for reversing earlier steps when a process is not completed. In both cases this reversing or compensating can be a very tricky things to implement. JDeveloper has a nice capability to switch back and forth between source and diagrams. This is a great learning tool and if you are handy with XML you can sometimes be more efficient working directly with the code. Workflow Builder has no underlying language to look at. Both tools have been known to get themselves in a confused state where they report a general error with no real clue as to where it exists and how to fix it. BPEL has a much greater facility for creating test scenarios and storing them with the application.

Conclusions

These two tools both have their strengths and weaknesses. Not surprisingly, they are best in the areas they were designed for and look a little awkward in areas they were not designed for but have been forced to supply. It will be interesting to see how the two play out and how the vast number of Workflows in E-Business Suite can be transitioned from Workflow to BPEL.