3

Launching and Viewing Your Workflow Process

Once you have saved a valid workflow definition to the database, the next step is to launch it and see what the Workflow engine does. In this chapter, we will look at the different ways to launch a workflow process, and how to use the Workflow Monitor to see what the process did and also to administer the process.

Objectives

By the end of this chapter, you should be able to

- Launch a Workflow process
  - Via the Workflow Web Pages.
  - Via a PL/SQL API.
- Define the following terms
  - Item Key
  - User Key
  - Process Owner
- Use the Workflow web pages to search for and monitor your process.
- Describe the information shown on each of the information tabs in the Workflow Monitor.
- Describe the function of the administrations buttons in the Workflow Monitor.

Launching your Process

There are three main methods that you can use to launch your process

1. Via the Workflow Web pages;
2. Via an API call using PL/SQL or Java;
3. Via a message arriving on an Oracle Advanced Queue.

Of these mechanisms, the first should **NEVER** be used in a production environment, since it relies on administrator access to the workflow system. However, this is probably the most important at this stage in developing our workflow, since it is the easiest mechanism for starting a workflow in a development environment, and so I will cover this one first.

Launching via the Workflow Web Pages

I can't really stress this enough, so I will say it again – you should **never** use this mechanism to launch a process, except in a testing environment.

Whether you are using Oracle Workflow embedded within the eBusiness Suite, or Workflow as a standalone product, there are HTML pages which can be used to administer your workflow system. If you are running Workflow as a standalone product, then the URL for the pages will be `http://<server>:<port>/pls/<DAD>/wfa_html.home`; if you are using eBusiness Suite

then you should log into the application as a user who has been assigned the responsibility which is set to administer the workflow system, and from your internet browser session change the end of the URL from `/OracleMyPage.home` to `/wfa_html.home`. The URL that you navigate to is what I will refer to as the "Workflow Homepage".

If you have administrator privileges, then you should see a screen similar to that shown in Figure 3-1.

Figure 3-1. Workflow Homepage viewed as a Workflow Administrator

If the only options you see are as in the screenshot in Figure 3-2, then you do not have Workflow administrator privileges.

Figure 3-2. Workflow Homepage viewed as a Non-Workflow Administrator

You can find out which role has been assigned to the administrator role, you can run the following SQL code:

```
SELECT wrol.display_name
FROM   wf_resources         wres
,      wf_roles             wrol
,      FND_RESPONSIBILITY_TL  frs
WHERE  wres.name = 'WF_ADMIN_ROLE'
AND    wres.text = wrol.name
AND    wrol.orig_system like 'FND_RESP%'
```

```
AND    frs.responsibility_id =
SUBSTR(wrol.name,INSTR(wrol.name,':',1)+1)
UNION
SELECT text
FROM   wf_resources
WHERE  name = 'WF_ADMIN_ROLE'
AND    text not like 'FND_RESP%'
```

If you do not have administrator privileges, then you can either connect as the user or responsibility that has been designated as the Workflow administrator or you can update the value held in the WF_RESOURCES table to either:

- Your user name (either FND_USER name or a database user),
- A responsibility that you have been granted, or
- A '*' which will allow every user administrator access.

If you are using Workflow in and Oracle Applications environment, my recommendation would be to set the value to '*' and then click on the "Global Workflow Preferences" link on the Workflow web-page and correct the value to a more appropriate one if necessary. This allows you to easily select a valid responsibility without needing to manually determine the correct values that would need to be manually entered in the table.

Once you have the appropriate administrator rights to the system, you should click on the "Launch Processes" link from the Workflow homepage, which will open a page as shown in Figure 3-3.

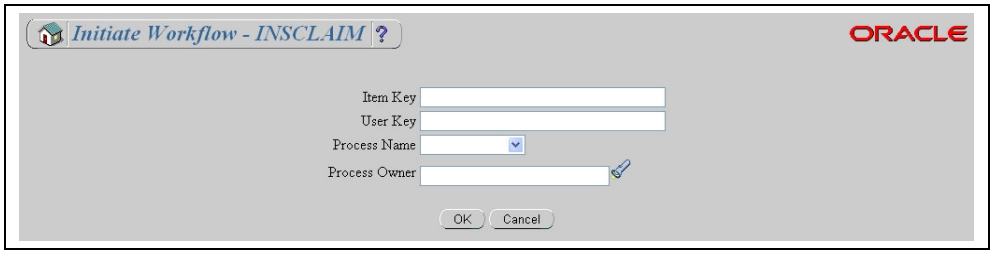

Figure 3-3. Launching a Process from the Workflow Web Pages

In order to manually launch your workflow via this page, you will need to provide some additional information via the form. Of the four fields which should be visible to you, only the "Item Key" field is mandatory in all circumstances, although in order to launch the process we have built so far, we will need to complete the "Process Name" field as well.

Before continuing to starting the process, let's add some more terms to the glossary.

Item Key

> An item key is a unique identifier for a particular instance of an item type. The combination of the Item Key and Item Type must be unique within the database, and Workflow will report an error if the combination already exists within the system. The easiest way to ensure that the item key is unique is to select it from a numeric sequence, or if the Workflow is based on a table with a primary key (for example the Applications Order Management Header Process uses the header ID from the OE_ORDER_HEADERS_ALL table as the item key for each process).

User Key

The User Key is an optional field, which is displayed in the Workflow monitor when the users search for their processes. If you are populating the user key (which is recommended for processes where users are likely to be enquiring about the state of the process), then this should be set to a user-friendly key so that users can easily find the appropriate process.

Process Owner

The process owner is, again, an optional field which should be set whenever you know the user who is initiating the process. As the above image shows, when initiating a Workflow process via the web page, the system will default to your current user name. If the Process Owner is not set, then only Workflow administrators will be able to view the process to examine the status of the flow. If the owner is set, then the user will be able to monitor the status of the processes that they are interested in more effectively.

In order to launch your newly created process, you now need to:

- Enter a new, unique Item Key. This can be anything you choose, up to 240 characters. My recommendation at this stage would be to use something that is easy enough to remember that you have used, since we will be launching a number of processes at different stages – I tend to use a number starting with 1 and incrementing with each launch.
- From the drop-down list for Process Name, select the only value that you should have there – "Main Process".
- Click on the "OK" button to launch the process.

Before we move on to looking at what actually happened when the process was launched, let's look at the different methods for launching processes, since this one should only be used for testing.

Launching a Workflow via a PL/SQL or Java API

No matter the complexity of the Oracle system in which you are working, you will always have the ability to call either a PL/SQL or Java API, and so this is the main method which can be used to launch a Workflow process.

The main ways that you can invoke a workflow using an API are

1. WF_ENGINE.LaunchProcess;
2. WF_ENGINE.CreateProcess and WF_ENGINE.StartProcess;
3. WF_EVENT.Raise;
4. WF_EVENT.Raise3.

Of these different calls, the calls to the WF_ENGINE package are also available in Java, whereas the WF_EVENT procedures can only be called via PL/SQL. If you are using any of these APIs to launch your workflow process, you need to ensure that you either commit or rollback once processing has completed, in order to view your process in any other database sessions such as via the Workflow Monitor.

WF_ENGINE.LaunchProcess

Table 3-1. Calling WF_ENGINE.LaunchProcess

PL/SQL Syntax	`Procedure LaunchProcess ( itemtype IN VARCHAR2`
	`, itemkey IN VARCHAR2`
	`, process IN VARCHAR2 DEFAULT ''`
	`, userkey IN VARCHAR2 DEFAULT ''`
	`, owner IN VARCHAR2 DEFAULT ''`
	`);`

PL/SQL Example	`WF_ENGINE.LaunchProcess ( itemtype => 'INCLAIM'`
	`, itemkey => 'Test1'`
	`, process => 'MAIN_PROCESS' );`

Java Syntax	`public static boolean launchProcess ( WFContext wCtx,`
	`String itemType,`
	`String itemKey,`
	`String process,`
	`String userKey,`
	`String owner )`

This API is the easiest method of launching a workflow process, as you only need to supply two mandatory parameters – the item type of the process and the item key to uniquely identify this instance of the process. If you know what process you wish to run, you should provide this as well – later we will look at writing code which allows a workflow to determine which process to run automatically. Once you complete the call, the workflow process will automatically start processing, and when the workflow process reaches a point when it can go no further (in our case so far the End point), then control of the database session will return to you. At this point, you should either commit the workflow transaction, or you can rollback completely and all workflow data will be removed from the database.

WF_ENGINE.CreateProcess and WF_ENGINE.StartProcess

Table 3-2. Calling WF_ENGINE.CreateProcess and StartProcess

PL/SQL Syntax	`Procedure CreateProcess ( itemtype IN VARCHAR2`
	`, itemkey IN VARCHAR2`
	`, process IN VARCHAR2 DEFAULT ''`
	`, user_key IN VARCHAR2 DEFAULT ''`
	`, owner_role IN VARCHAR2 DEFAULT ''`
	`);`
	
	`Procedure StartProcess ( itemtype IN VARCHAR2`
	`, itemkey IN VARCHAR2 );`

PL/SQL Example	`WF_ENGINE.CreateProcess ( itemtype => 'INCLAIM'`
	`, itemkey => 'Test2'`
	`, process => 'MAIN_PROCESS' );`
	
	`WF_ENGINE.StartProcess ( itemtype => 'INSCLAIM'`
	`, itemkey => 'Test2' );`

Java Syntax	`public static boolean createProcess ( WFContext wCtx,`
	`String itemType,`
	`String itemKey,`

	String process) public static boolean startProcess (WFContext wCtx, String itemType, String itemKey)

Using the LaunchProcess API to initiate your workflow will immediately start the process as well. However, there will be a number of times when you need to pass some information into the process before it starts, which is where these two APIs should be used instead. Firstly, the call to CreateProcess will create a new instance of the workflow. Again, you need to provide the item type and item key information, and since there is no way for the workflow engine to automatically determine which process to run, the process name should also be provided.

Once you have called the first API, then you can use subsequent APIs to setup data as required. Until you call the second API, your workflow will not begin processing. Whenever I am invoking a Workflow from PL/SQL using a WF_ENGINE API call, I always use this method rather than calling the LaunchProcess API above. The main reason for this is that if the process needs to have some additional attributes set within the workflow, then there is already a place in the calling code to do this. If you have used LaunchProcess instead, then you will have to re-code the call to invoke the workflow, as well as adding your new PL/SQL code.

WF_EVENT.Raise / WF_EVENT.Raise3

Table 3-3. Calling WF_EVENT.Raise or WF_EVENT.Raise3

PL/SQL Syntax	Procedure Raise (p_event_name IN VARCHAR2 , p_event_key IN VARCHAR2 , p_event_data IN CLOB DEFAULT NULL , p_parameters IN WF_PARAMETER_LIST_T DEFAULT NULL , p_send_date IN DATE DEFAULT NULL); Procedure Raise3 (p_event_name IN VARCHAR2 , p_event_key IN VARCHAR2 , p_event_data IN CLOB DEFAULT NULL , p_parameters IN OUT NOCOPY WF_PARAMETER_LIST_T , p_send_date IN DATE DEFAULT NULL);

The two different calls to the WF_EVENT package essentially do the same thing – the only difference is that the Raise3 procedure returns the parameter list back to the calling code, once processing has completed.

The PL/SQL calls raise a business event to the Business Event Manager, which executes any event subscriptions which are defined for that event. If there is a subscription which links to a workflow process, then this call can be used to restart an existing workflow or to initiate a new process.

We will revisit this API when we start to use Business Events within our process in Chapter **XX**.

Launching a Workflow via Oracle AQ

The final way that your process can be initiated is via a message arriving on an Oracle Advanced Queue. Either a custom piece of code can be written which dequeues the message and then processes it by calling one of the PL/SQL APIs above, or you can register the queue with the Business Event System (BES) and the Event Manager will manage the dequeue and raise the appropriate event automatically.

Starting workflows via messaging products such as Oracle AQ is a complex subject, which we will revisit when we look at event-driven workflows in Chapter **XX**.

Viewing your Process

Whichever mechanism you used to launch your process, that particular instance of the process will now have been executed in the database – let's see what it did!

If you launched your process via the Workflow homepage, then when you clicked on the "OK" button the screen will have moved on to show you the "Activities List" page, and you can skip to the next section. If not, then we need to get to that point manually.

Navigating to the Workflow Activity List

If you are using standalone Workflow, then you should navigate to your workflow homepage, which will be in the following format

```
http://<machine name>:<port>/pls/<DAD>/wfa_html.home
```

If you are using Workflow embedded in the eBusiness Suite, then you should log into a user with Workflow Administrator responsibility, and choose "Find Processes" from the menu. Alternatively, you can change the URL in your browser window and replace the OracleMyPage.home with wfa_html.home at the end.

The Find Processes page (Figure 3-4) allows you to search for processes which are persisted in the database. You can filter your search as necessary, or leave the default selection which will return all workflows in any state. If there have been a number of workflows running in your environment (particularly with eBusiness Suite), then I would advise filtering your search using the Item Type field at the least. If you enter the item key that you used to launch your process, then the system will perform a LIKE query automatically.

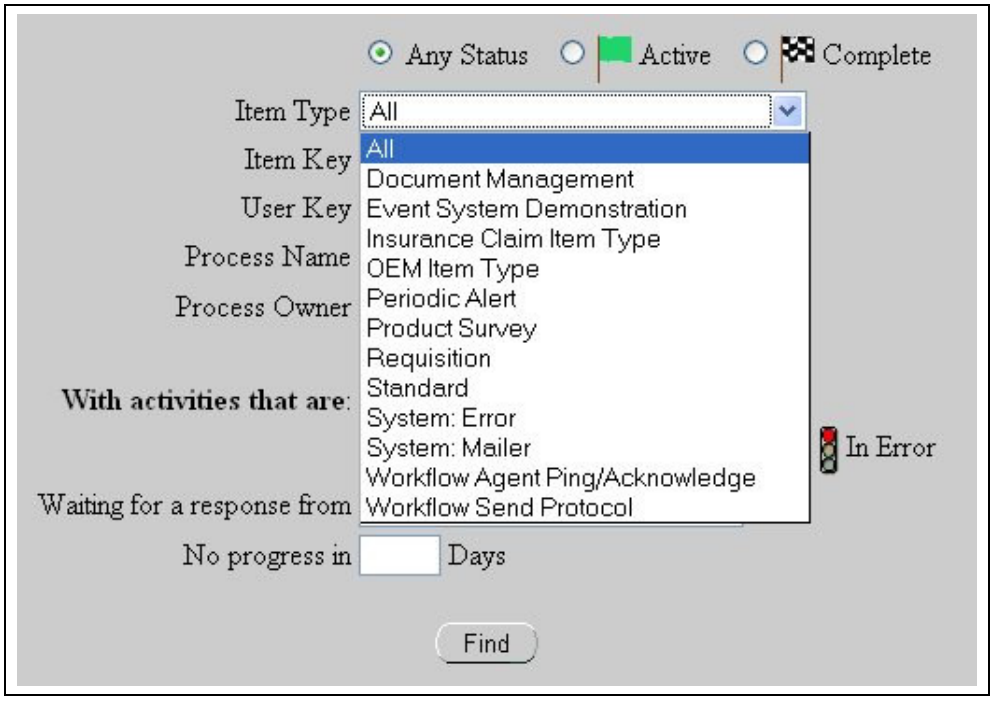

Figure 3-4. The Find Processes page

When you click on the Find button, you will be presented with a list of processes which match your search criteria, and you can then click on the process name to open the Notifications List.

The Workflow Notifications List

When you first open the screen for your process, it can seem a little disappointing – there isn't any indication of what happened to your process apart from a flag icon, as shown in Figure 3-5.

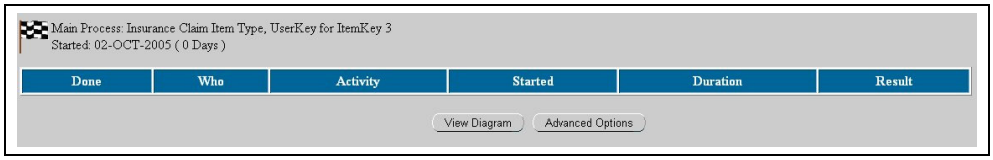

Figure 3-5. The Workflow Notifications List

The main reason that users will access this screen is to view why their workflow has not finished processing – generally because it is waiting for someone to perform an action with a notification. For this reason, the default screen that you see when you open a process in this manner is a list of any notifications which are active. Since we won't be adding any notifications to the process until the next chapter, the list is blank.

To view what happened to your process in a text format, you should click on the "Advanced Options" button and select every option in the row of checkboxes in the middle of the screen (Figure 3-6) before clicking the "Filter Activities" button, which will open the Workflow.

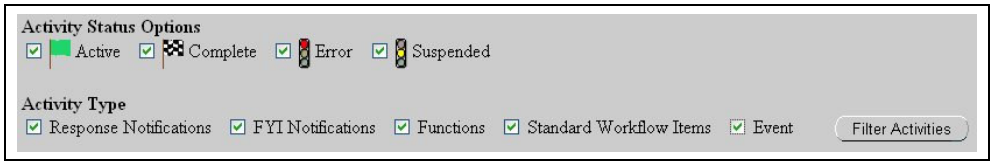

Figure 3-6. Filtering Activities

The Workflow Activities List

The process activity list that you are presented with after launching your process shows in a textual format what activities have been performed during the process. Since our process is particularly short at the moment, there are only these three steps shown, which took no time to execute, as shown in Figure 3-7.

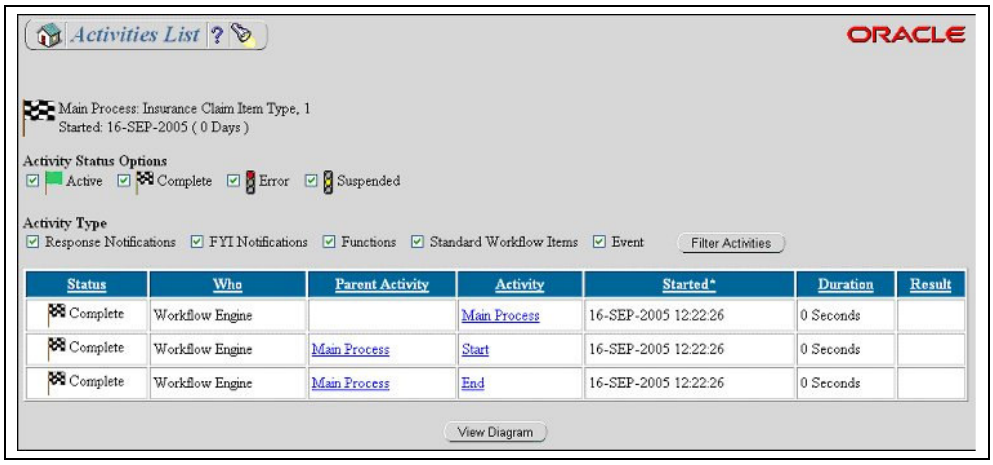

Figure 3-7. Workflow Activities List

When the process gets more complicated, the two different sets of check boxes on the screen will become more important, since they allow you to filter the display to your needs. If you search for a workflow process, you will be presented with this screen but instead of showing every activity that the process has executed you are only shown the status of open notifications which require a response. The reason behind this is that the vast majority of enquiries which require a user to search for a particular workflow tend to be on a theme of "who is holding the process up at the moment?" In these situations, showing a user what the process has already performed is of little or no benefit to them – their focus is on what is happening with the process now.

Every activity which the process has performed has a status icon. If the activity is currently being performed, then a green flag is shown – you should only see this when the process is waiting for a user to respond to a notification. In the process above, the Start and End activities completed have completed, and so the status shows a chequered flag. Once the process has completed, the status flag in the first row will change from a green flag to a chequered one, or in the worst-case scenario a red traffic light showing that the process has errored. The status of the process as a whole can be seen from both this first line in the activity list and also the flag icon in the top left hand corner.

HELP! *My process hasn't completed!*

If everything has gone right, then your workflow will be showing a chequered flag icon. If not, then don't worry – believe me this won't be the last problem you have in developing workflows!

At this stage, there are only a few things that you can check:

- Have you got both a "true" start and end?
- Are the start and end nodes connected?
- Is the WF_STANDARD package compiled and valid in the database?

If the package is not valid, then you should check the installation completed successfully, since this is installed by default with every Workflow installation. Whilst the other possibilities may seem unlikely, you will need to go back to your process definition in the Workflow builder and review the activities to ensure that the start and end nodes are set correctly, and that the transition from start to end point connects the two activities correctly. I once ran a training course where some of the delegates attempting moving an icon on top of the transition rather than creating a link between each of the steps.

If your process has not completed successfully, and you cannot easily see what is wrong with the workflow, at this stage it may be easier to start again from the beginning of chapter two.

Since our process is particularly short at the moment, it is easy to determine what happened within it – it started and then ended. However, once the process starts to have more complexity, then this list becomes increasingly difficult to understand – particularly if the process has branched and is executing two separate threads in parallel. In that case, the Workflow Monitor becomes an invaluable tool in determining what your process did – click on the "View Diagram" button now to open it.

The Workflow Monitor

When you open the Workflow Monitor by clicking on the "View Diagram" button, a screen similar to that shown in Figure 3-8 will open.

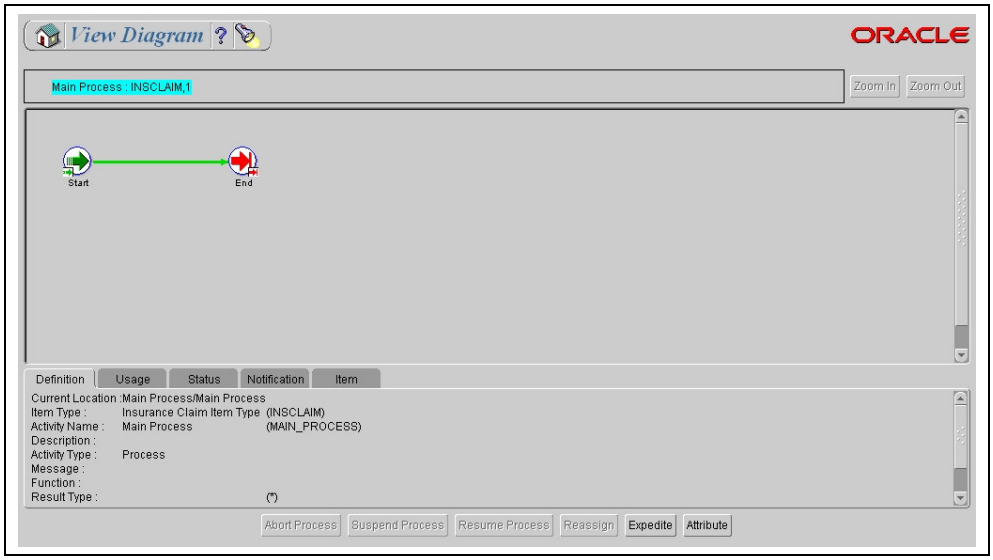

Figure 3-8. Viewing your process in the Workflow Monitor

The workflow monitor is a Java Applet which shows to the end-user the process *exactly* as it was designed in the builder. This is one reason that when you are building the processes, you should try to ensure that the flow of the process on screen is fairly logical and clear to see. Additionally, since the users will see activities shown by their icons that you choose and the display name, you should try to make sure that these are appropriate and understandable to the end-user.

Workflow Information Tabs

Immediately below the process diagram, there are five tabs which provide information about the process and the activities within that process.

Definition Tab

The definition tab shows the key information about whatever activity you have selected in the process diagram. If no activities are selected, then the tab will show you the definition of the whole process, as in Figure 3-9.

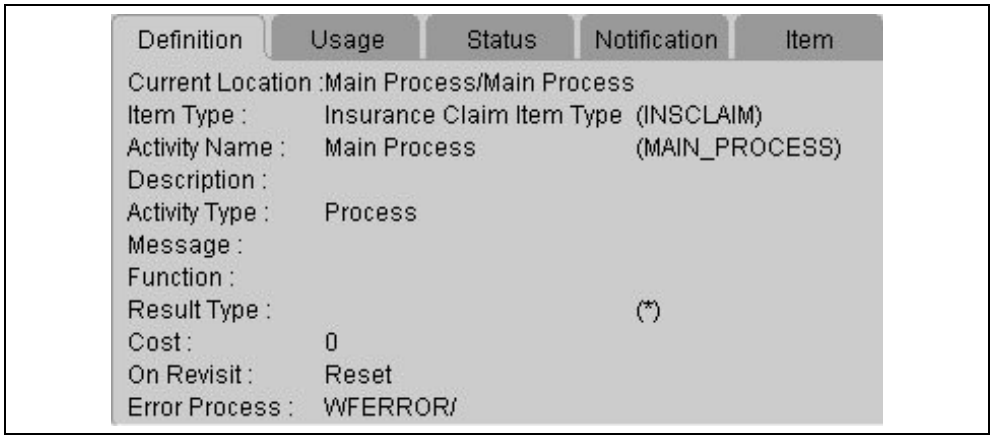

Figure 3-9. The Definition Tab for the Main Process

If you select an activity, then the details will change accordingly. One of the most important things that you can check here is to see what piece of code is being called for each activity – when you select a function, the "Function" field on the tab will change to show you what was called (or will be called) for that activity. If you have changed the workflow definition a number of times and altered which code is executed for the activity, then this can be of great use since you can look at your process instance in the database and ensure that it is calling the code that you expect it to.

Usage Tab

The Usage tab (Figure 3-10) provides details on how the process or activity that you select in the diagram is used within the process.

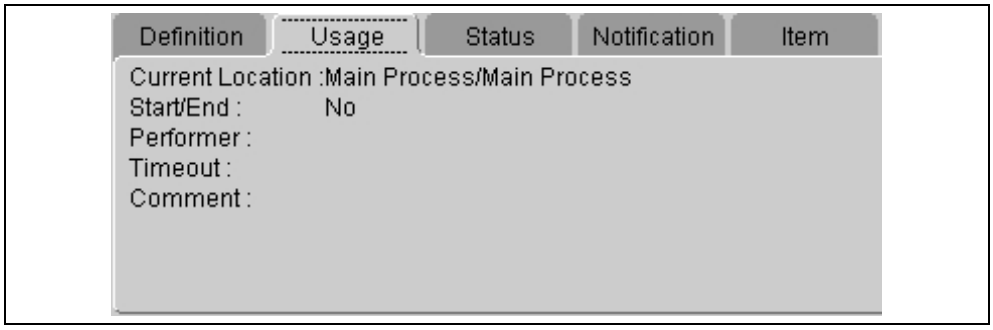

Figure 3-10. The Usage Tab for the Main Process

When you highlight an activity, the tab will indicate whether the node has been set as a Start or an End point. If the activity that you have selected has any attributes associated with it, then these will be displayed on this tab as well – since we have not defined any yet, the space where these appear (after the Comment field) remains blank.

Status Tab

The Status tab (Figure 3-11) indicates what the activity or process is currently doing, and for completed processes or activities what the result was. The process that we have built

so far does not return any result to a calling process, and so the result shown in the tab is "Null".

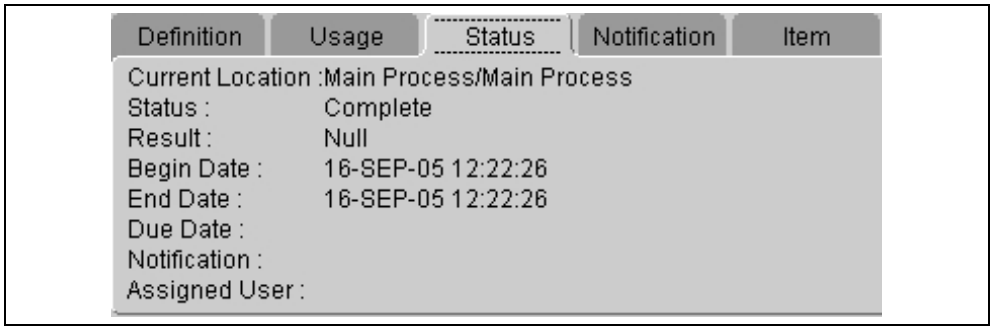

Figure 3-11. The Status Tab for the Main Process

The diagram should give you a clear indication of what the result was, but this is not always the case. If an activity has multiple possible results, it is sometimes easier to model the results that you are interested in and then use a "default" transition to handle any other results, as in the example process in Figure 3-12.

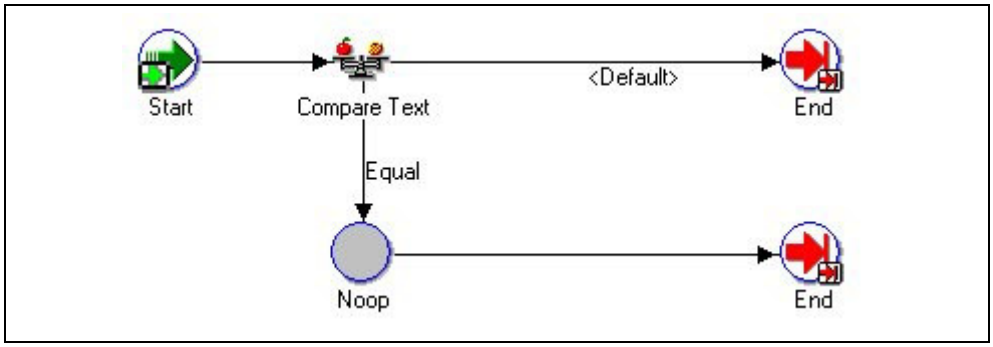

Figure 3-12. Using a "Default" Transition

In this example, the only response that I am interested in is "Equal" – any other responses (Greater Than, Less Than or Null) should all transition directly to the End node, rather than to the Noop activity. Looking just at the workflow diagram merely indicates that the process took the default route, not what the actual result was. Using the Usage tab, the actual result that the "Compare Text" activity returned was "Greater Than".

The usage tab also indicates how long a process or activity took to perform, which can then be used in looking at tuning your workflow process.

Notification Tab

The Notification tab (Figure 3-13) is used to indicate which user, users or role has been sent a notification, and what their response was.

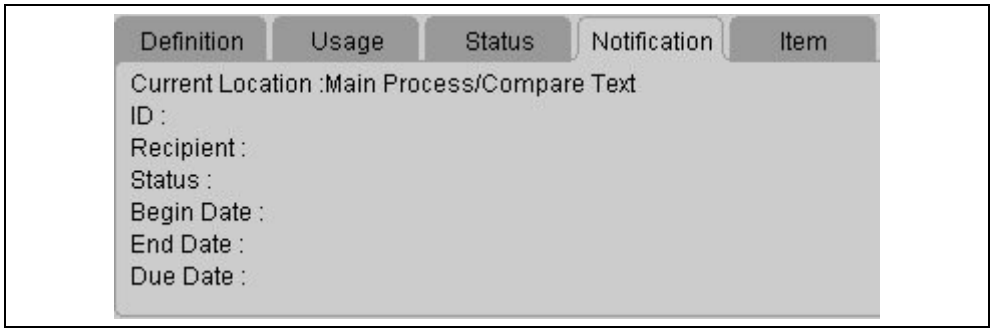

Figure 3-13. The Notification Tab for the Main Process

The tab also displays the notification ID, which can then be used for searching. Chapter 4 introduces notifications and communicating with people, and so we will cover this in more detail in that chapter.

Item Tab

Figure 3-14 shows the final tab, which displays information about the process, including the current values of all the attributes which have been defined. The information is in a display only format, and so users can view the key information about the process but not have any impact on how the process runs. A Workflow administrator can update this information using the Administration buttons described in the next section.

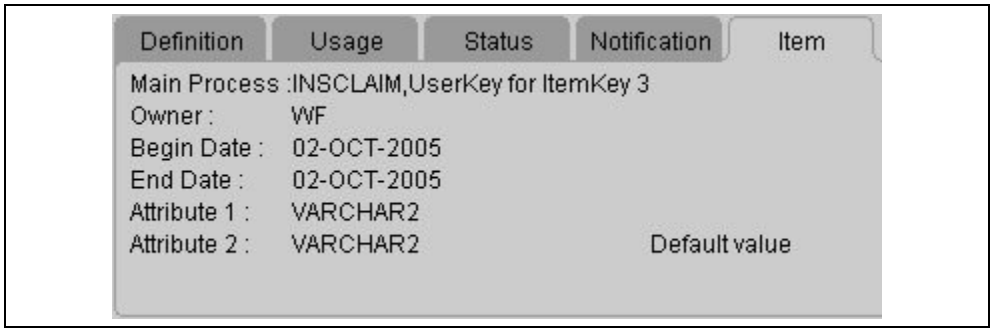

Figure 3-14. The Item Tab for the Main Process

Workflow Administration Buttons

As well as displaying the process diagram and the information tabs, there are buttons at which provide extra control for workflow administrators. Since the tasks that a workflow administrator can perform through this page can have dire consequences if misused, it is important that you understand both the function that each tab performs, and also the importance of restricting how many workflow administrators have access to your system.

Workflow Monitor Buttons

There are two different collections of buttons which are visible in the workflow monitor. In the top right hand corner of the screen, there are two "Zoom" buttons. If your process contains sub-processes, then you can use these buttons to drill into each sub-process to see what activities were performed.

At the bottom of the screen there is a group of six buttons, of which only the last two buttons should be enabled (Figure 3-15).

Figure 3-15. Administration Buttons in the Workflow Monitor

- Abort Process

  If the process is still running, and you are a workflow administrator, then this button will be enabled. It allows the user to stop the process running completely, and will allow the workflow data to be purged from the database.

- Suspend Process

  If the process is still running, then this button will also be enabled. This allows a workflow administrator to pause the workflow and stop it running until told otherwise. This is particularly useful if it is highlighted to the administrator that there is a bug in the code – if the process has not yet reached the section of code which is incorrect, the process can be paused to enable a developer to correct the problem. Once the problem has been fixed, then the "Resume Process" button can be used to continue processing. If the process is not suspended, and so is allowed to continue through problematic code, then the process might error or (even worse!) complete "successfully" and impact on other parts of the business. If a process is suspended, then the list of processes will show an amber traffic light () to show that it is paused.

- Resume Process

  The "Resume Process" button will only be active if you are viewing a process which has been suspended.

- Reassign

  If you highlight a notification step within the workflow, then the "Reassign" button will activate. This allows an administrator to either delegate responsibility or transfer ownership of a notification completely to a different user or role. This is useful if the system has incorrectly identified the person who should respond to the notification, for example if they are away on holiday or on temporary leave and have not set their holiday settings to automatically route their notifications to other users.

- Expedite

  The "Expedite" button is, in my opinion, the **BEST** reason for ensuring that users do not have administrator privileges to your workflow system. This button allows you to force a workflow process down a particular route, to re-execute steps already performed, or to skip some parts of the process and continue from there – in short it allows you to jump to **ANY** point in the process and execute that step. In order to do this, you simply highlight the step in the diagram, and then click on the button, which will prompt you to Retry the step or Skip it completely, providing a response if any is required. The important thing to note here is that the process does not need to have executed that step before – you can expedite a workflow from any point in the process.

- Attribute

  The final button allows you to see and change any attributes that are defined within the process. If you click on the button now, the only information that you will see is the item type name, either the item key or the user key if you provided one and the date at which the process started and ended (Figure 3-16). If the process contained any item attributes, then you will see the list of attributes and can change any if necessary (Figure 3-17).

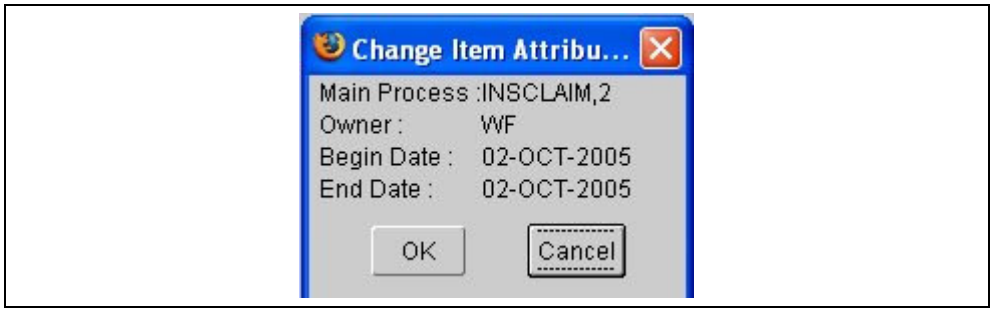

Figure 3-16. Change Attributes Window for a Process without Attributes

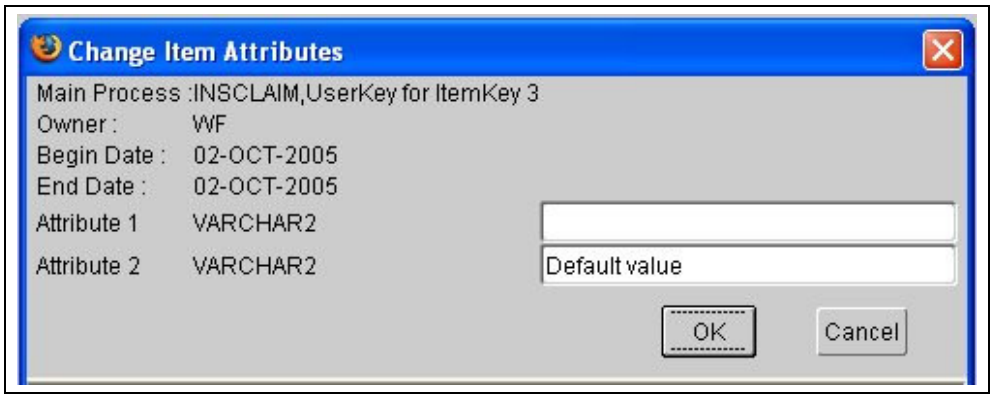

Figure 3-17. Change Attributes Window for a Process with Attributes

Figure 3-16 also shows a process where the user key was not set. The top line, therefore, shows the internal name of the item type (INCLAIM) and the item key (2). The process in Figure 3-17 has had the user key set to the value 'UserKey for ItemKey 3' and so this is displayed to the user instead of the item key, since the user key should be more meaningful.