

## Unit 22: Simulation

ActiveBPEL Fundamentals



This is Unit #22 of the BPEL Fundamentals course. In past Units we've looked at ActiveBPEL Designer, Workspaces and Projects, created the Process itself and then declared our Imports, PartnerLinks and Variables and then we created Interaction Activities in various ways. Next, we looked at the Sequence activity, Assignments and Copies and after that we studied Correlation, Scopes and Fault Handling. Then, we examined Compensation, Event Handling, Termination Handlers and the If activity and we also looked at the rest of the BPEL Basic activities. In the next four Units we looked at BPEL's Flow and Pick activities, the various types of Looping activities and at the Extension capabilities that are available in the BPEL v2.0 specification and how Active Endpoints has implemented them. In the last Unit we looked at three Advanced BPEL topics: Isolating Receive-Reply pairs, Additional Message definitions and Re-using BPEL snippets. In this Unit, we will examine the Simulation of a BPEL Process in the Designer.

## Unit Objectives

- At the conclusion of this unit, you will be familiar with:
  - Simulation overview
  - Sample Data
  - ActiveBPEL Debug Perspective
  - Simulating
  - Interacting with a simulation session

## Simulation Overview

- After designing but before deploying a BPEL process, you can simulate execution in ActiveBPEL Designer
  - Provides the ability to fully test a process's logic paths
    - Without the need for invoking the actual Web services
    - Sample data is used in place of the real data that a running process would normally receive and send
- Allows all paths of a process to be tested
  - Including faults, alarms, transitions, and join conditions
- Process execution is identical to that of deployed process
  - Except that it does not interact with any live Web services

3 Copyright © 2004-2007 Active Endpoints, Inc.



After you have completed designing your process, but before you deploy it, ActiveBPEL Designer gives you the option of *simulating* the process. A Simulation lets you execute the process and test all of the different execution paths and all of the processes activities, but without actually sending and receiving messages from your process partners. It does this by using Sample Data in place of the messages that are sent and received at run time. This means you can test the process without actually invoking any of the web services it interacts with. Execution control allows you to simulate all process actions, including Faults, Alarms and all of the various expression conditionals that are defined by the process, such as those for Joins, Ifs and Transitions.

## Prerequisites for Simulation

- Process must first be saved and contain no error messages
  - Warning and information messages are OK
- Ensure that the Abstract Process property is set to No for the process
  - The default is No
- Make sample data available for the input, fault, and output variables that a running process would receive and send.
  - For all execution paths you want to test
- Understand the ActiveBPEL Debug Perspective and Views

4 Copyright © 2004-2007 Active Endpoints, Inc.



Before we begin simulating a process, we have to have met all of the pre-requisites and setup requirements for doing so. First, the process must be saved and have no errors, though Warnings and Informational messages are fine. Second, an Abstract process cannot be simulated and the process property setting for this must be set to No. Third, there must be Sample Data defined for all places in the process that would be sending and/or receiving data from/to its partner services. (This only applies to those activities that are in the simulation's execution path. If the activity is not executed it need not have Sample Data defined.) Fourth, you must understand how the ActiveBPEL Debug Perspective and Views work.

## Unit Objectives

- At the conclusion of this unit, you will be familiar with:
  - ✓ Simulation overview
  - Sample Data
  - ActiveBPEL Debug Perspective
  - Simulating
  - Interacting with a simulation session

## Sample Data Overview

- Used to initialize certain input and output variables with sample data values
  - Basically to simulate the actual data that a running process would normally receive or send
- Sample data is needed for the following
  - Variable used in a receive or onMessage activity
  - Output variable in the invoke activity
  - Fault variable in the invoke activity
- Sample data files can be added via:
  - Web References
  - Process Variables
  - Simulation properties of an activity
  - XML Sample Data Generation Wizard

6 Copyright © 2004-2007 Active Endpoints, Inc.



The purpose of Sample Data, of course, is to act as stand-ins for the Messages that the process would receive from one or more of its partners during process execution. It is required for any variable that is used in a Receive or onMessage activity (including onEvents and Picks), and for any Output or Fault variable that comes in from an invoked partner service.

The Sample Data files can be added:

- by associating them with specific Schema/WSDL *files* in the Web Reference View
- by associating them with a specific *variable* through the Process Variables View or Outline View
- by adding them to the Simulation Properties of a specific *activity* (by using the activity's Properties View)

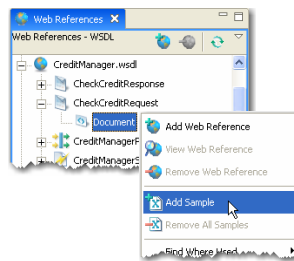
The other option is to use the Designer's XML Sample Data Generation Wizard. When using the wizard, which guides you through the generation of your samples, the Designer will associate the Sample Data based on where you are when you open it. Note that, because Sample Data can be added in multiple locations for the same process, you can have more than one set of Sample Data defined for a specific Variable or Message during a simulation. Because of this, there is an order of preference for which data set will be used during a simulation run. It is, in order:

- Simulation Properties Data
- Process Variables Data
- Web References Data

Now that we have an overview and we know how Sample Data works, let's take a look at each of the various ways we work with it in the Designer.

## Adding Sample Data via Web References

- Used to define sample data that is shared across all processes in the current workspace
- Expand a WSDL message and expand to select a Message Part
  - Right-click on the Message Part
    - From the pop-up menu, select Add Sample
- Multiple samples can be associated to each part



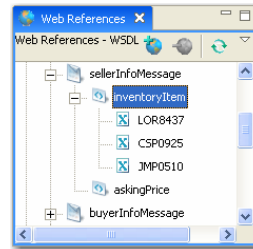
7 Copyright © 2004-2007 Active Endpoints, Inc.



First, we'll look at how we can add data through the Web References View. First, set the View's filter to show all of the WSDL messages, and then expand the one you want to associate data with. Use the Right Mouse context menu and select "Add Sample." Designer will allow more than one sample can be associated with a single message, and one of them must be chosen as the default. (Note that Sample Data associated with a Message in the Web References View is part of the *Workspace*, and is therefore visible and usable by any project or process in that *Workspace*.)

## Simple Type Sample Data via Web References

- For Message Parts which are defined using simple types
  - Specify the actual value directly using the Add Sample Data dialog
  - e.g., xs:string, xs:integer, xs:date, etc.
- Sample data for simple types
  - Once created, cannot be edited
  - Are not validated
  - Shown as the value entered



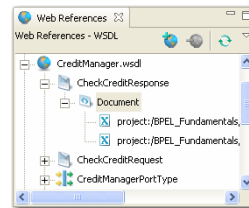
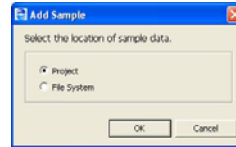
8 Copyright © 2004-2007 Active Endpoints, Inc.



When associating Sample Data through the Web References View, we must distinguish between Simple Types and Complex Types. Data for Simple Types can be specified by directly adding the value, and the treeview will then show the actual value in the display. Once created, these values cannot be edited and are not validated by the system.

## Complex Type Sample Data via Web References

- For Messages Parts which are defined using complex types
  - Added as an XML instance file representing the complex type
  - Can be added from
    - Any project in the workspace
    - Anywhere on the file system
  - Can be dynamically generated
- Sample data files are
  - Validated against the message part's schema definition
  - Viewable but not directly editable
  - Shown as the path to the sample data file



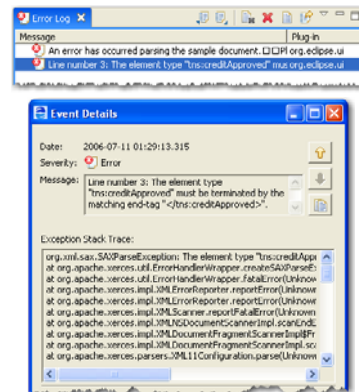
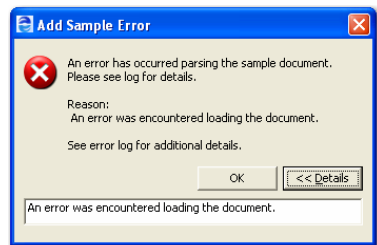
© Copyright © 2004-2007 Active Endpoints, Inc.



When adding Complex Types (i.e., those types not defined as one of the XMLSchema Standard Types) they are added in the form of a file, rather than as the actual value. They can be added from anywhere in the Workspace (meaning from any project within that Workspace) and they can be dynamically generated. When a Complex Type is used, it is validated against the Schema definition upon which it is based. The treeview will show the file path and name rather than the actual value.

## Complex Type Sample Data – Errors

- When adding Sample Data to Message Parts errors might occur
  - Usually due to invalid XML
  - Examine the Error Log view for the list of errors and their details



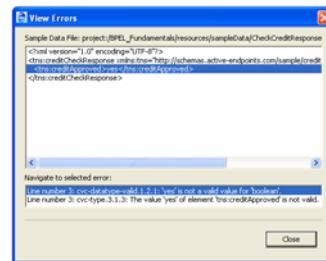
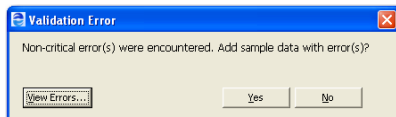
10 Copyright © 2004-2007 Active Endpoints, Inc.



Sample Data is generated in XML, and errors can occur. Designer will show you a list of these errors on the Error Log Tab, and they can be opened and examined by clicking on the Details button.

## Complex Type Sample Data – Validation Errors

- When adding Sample Data to Message Parts validation errors might occur
  - Can still be added as a sample
    - Represents case where invalid XML is passed into a BPEL process
  - View the details about the validation error(s) by selecting View Errors...
    - You can use the lower list box to navigate the specific validation errors found



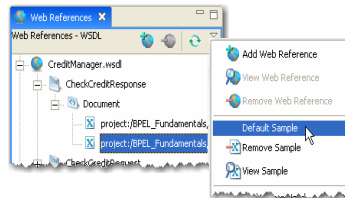
11 Copyright © 2004-2007 Active Endpoints, Inc.



If you have errors in a Complex Type's Data Sample, you have three options: change the data, remove the data, or let the bad data be used by the simulation. The purpose of using bad data in your simulation is to see how the process reacts when it receives such erroneous data, and to check on the Fault Handling, for example.

## Adding Sample Data via Web References - Default

- You can add several sample data files per message part in the Web References view
  - Select one sample file as the default file that is loaded into process variables
- In Web References view, expand a WSDL message and then expand one of the message's parts
  - Select a default sample data file, right-mouse click on a sample file and select Default Sample

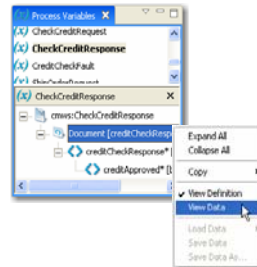


12 Copyright © 2004-2007 Active Endpoints, Inc.

You can associate more than one Sample Data file with a WSDL message, but note that one of them must be set as the default. To do this, change the Web References filter to "Messages," select the appropriate message and then use the Right Mouse context menu to set the file as the default Sample Data.

## Loading Sample Data via Process Variables View

- Used to load the sample data that will be used by the selected process definition during simulation
  - If present, the default sample data specified in Web References will be initially loaded
- From the Process Variables view
  - Double-click on the Variable you want to add sample data to
    - This opens the variable details within the view
  - Right-click anywhere in the variable details and select the View Data menu item



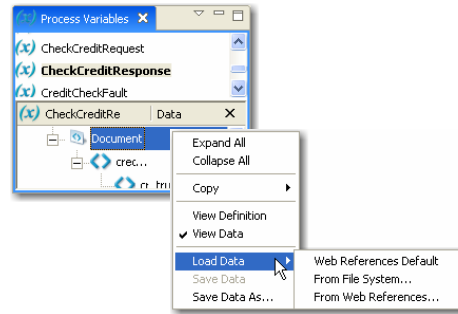
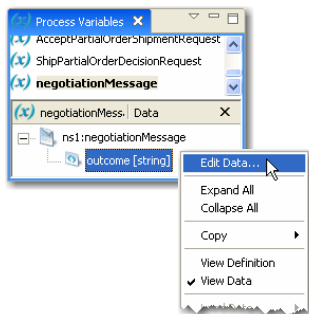
13 Copyright © 2004-2007 Active Endpoints, Inc.



You can also load Sample Data from the Process Variables View. If you have previously specified a Default Sample Data file for a particular variable using the Web References View, that data will be shown, and it will be used in the Simulation. If you have not added any data, just double-click on the variable or use the Right Mouse menu after selecting it and .

## Loading Sample Data via Process Variables View

- Right-click on the message part and select
  - Edit Data
    - for simple-type data
  - Load Data
    - for complex-type data



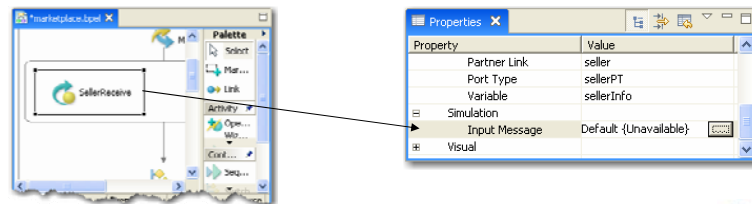
14 Copyright © 2004-2007 Active Endpoints, Inc.



When in the Process Variables View, there are two kinds of Sample Data that can be loaded, simple and complex, as we discussed before. For Simple Data types, you can view and edit the data directly by using the Right Mouse context menu and selecting "Edit Data." For Complex Data, you can load a data file from the File System or use one of the files associated in the Web References View. You also have the option to use the Web References default.

## Overriding Sample Data via Simulation Properties

- During simulation, used to override already loaded sample data for any activity that has not yet executed
- With the activity in focus, you can load sample data in the Properties view
  - For a Receive or onMessage, use the Input Message Simulation property
  - For an Invoke, use the Output Message or Fault Message Simulation property



15 Copyright © 2004-2007 Active Endpoints, Inc.



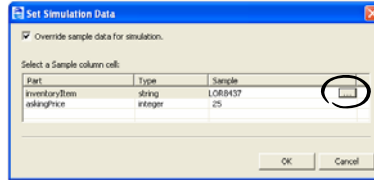
You can also associate Sample Data directly with the process activity by going to the Property View and selecting the ellipsis icon in the Simulation field, as seen above. For a Receive or onMessage activity, you set the Input Message and for an Invoke you set either the Output Message or the Fault Message. Now that we know that Sample Data can be associated with an activity in four different places, how do we know which one will be used when we run our simulation?

As we discussed earlier, the choice of which Sample Data the Designer is going to use is decided in this order:

- 1.) The activity's simulation data in the Property View
- 2.) The Process Variables View data
- 3.) The Web References' (default) associated data/file

## Overriding Sample Data via Simulation Properties

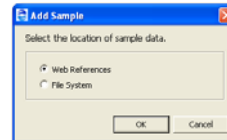
- Select the checkbox next to Override sample data for simulation then select a message part
  - Select the dialog button at the end of the selected row



- For simple-type data



- For complex-type data



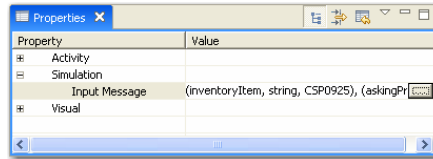
16 Copyright © 2004-2007 Active Endpoints, Inc.



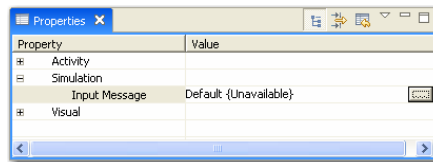
Once you've selected the Simulation field's ellipsis icon, you are shown the "Set Simulation Data" dialog and shown the sample data choices. There is a checkbox that enables or disables the use of this data, overriding any other associated data. For any sample data, you can click on the Sample data field's ellipsis icon to set the data. For Simple data types, you can fill in the value and for Complex data types you can associate a data file from the File System or from the files in the Web References View.

## Overriding Sample Data via Simulation Properties

- When Override is enabled, sample data value will be shown



- When Override is not enabled, Default {Unavailable} will be shown



Once you've done this and gone back to the Properties View, you will either see the data or a "default{unavailable}" notation.

## Selection Of Sample Data

- Sample data is loaded during Simulation in the following sequence
  1. Any Input, Output, or Fault Simulation properties specify an override
  2. The currently loaded sample data in the Process Variable view
  3. Otherwise, the Web References default sample data

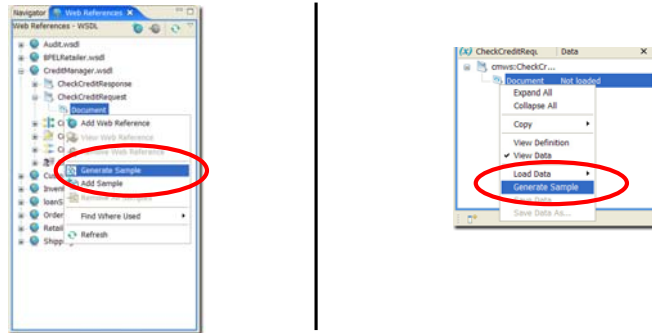
18 Copyright © 2004-2007 Active Endpoints, Inc.



Here is the Sample Data selection sequence, as we discussed earlier.

## Generating Complex Sample Data

- XML Sample Data Generation Wizard
  - Generates XML Instance Data files based on schema
  - Invoke from Web References view or Process Variables view



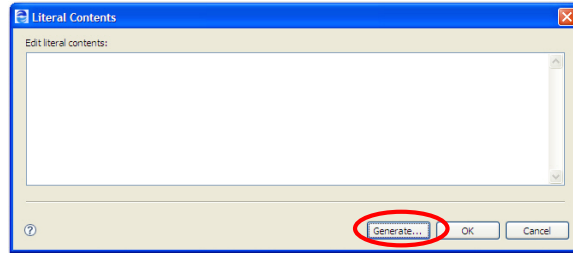
19 Copyright © 2004-2007 Active Endpoints, Inc.



ActiveBPEL Designer provides a Wizard that will generate an instance of the desired Complex Data Type. This can be done from the Web References View or from the Process Variables View.

## Generating Complex Sample Data

- XML Sample Data Generation Wizard
  - Available during Literal Copy operation construction
  - Available from Variable Initialization dialog



20 Copyright © 2004-2007 Active Endpoints, Inc.



The Sample Data Wizard can be used in the Variable Initialization dialog or in the Copy Operation's dialog when constructing a Literal for use in a Copy operation.

## Generating Complex Sample Data

### ■ XML Sample Data Generation Wizard

- Generate XML Instance from Complex Type or Element
- Optionally generates
  - Attributes / data
  - Elements / data
  - Repeating elements
- Abstract elements / types generated and/or commented
- XML Instance File saved
  - to user-selected project



21 Copyright © 2004-2007 Active Endpoints, Inc.



The Sample Data Generation Wizard allows you to create samples of Complex data types or elements. This wizard will optionally include attributes and elements in the sample it generates, and you have the option of creating repeating elements. (For example, if a Complex Type's nodes can repeat, this would allow you to generate a sample with more than one node.) Abstract elements/types can also be generated, with the option to create multiple elements that are all commented out except one. Once you've filled out this screen, you can click Next and save the generated file as part of the project.

## Unit Objectives

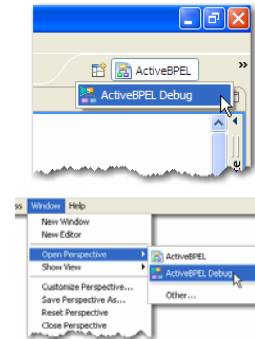
- At the conclusion of this unit, you will be familiar with:
  - ✓ Simulation overview
  - ✓ Sample Data
  - ActiveBPEL Debug Perspective
  - Simulating
  - Interacting with a simulation session

## ActiveBPEL Debug Perspective

- Contains the views, editors, menus, and toolbars that support simulated execution of BPEL processes
- Automatically switches to the Debug perspective when you begin simulation
  - Can also manually display the perspective

- From the Perspective bar, select the ActiveBPEL Debug Perspective icon

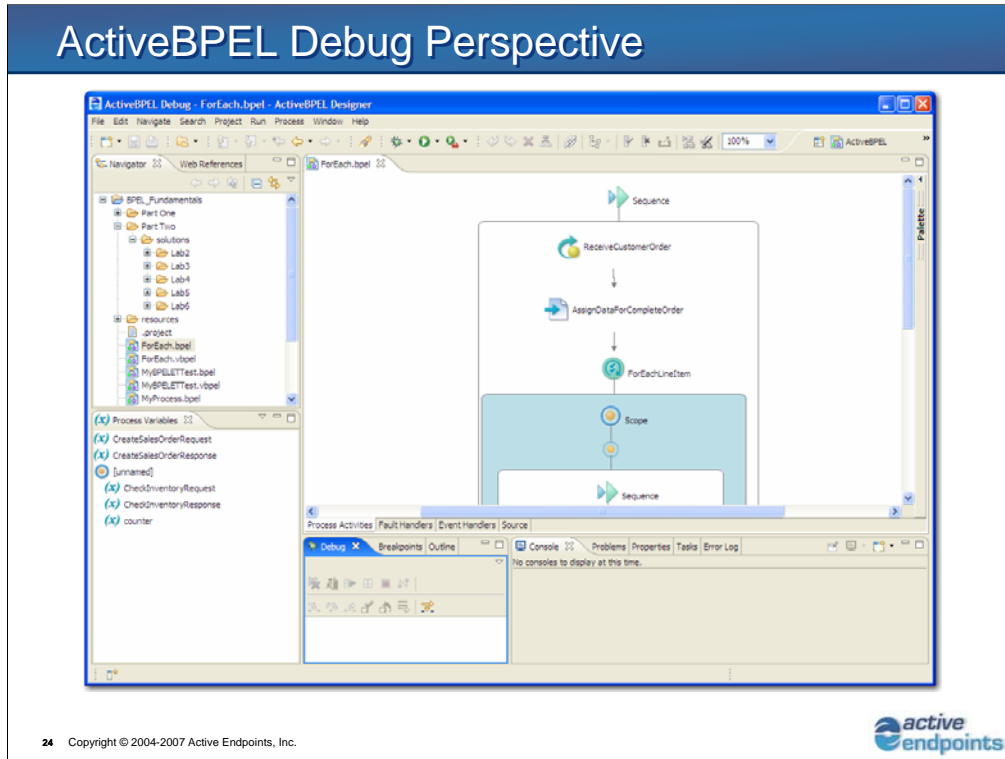
- From the Window menu, select Open Perspective>ActiveBPEL Debug



23 Copyright © 2004-2007 Active Endpoints, Inc.



Once you have associated Sample Data with each of the activities that will be executed in your simulation path, you are ready to start the simulation session. Switch to the Debug Perspective, from either by using the Window->Open Perspective->ActiveBPEL Debug menu choice or by selecting the Debug Perspective. Once open, the Debug Perspective will show you the Views, Editors, Menus, and Toolbars that are appropriate for your simulation session.



Here is a screenshot of the ActiveBPEL Designer's Debug Perspective. Note the Debug and Breakpoints Views and the Console Tab.

## ActiveBPEL Debug Perspective - Views

- This perspective introduces 3 new views
  - Debug
    - Controls the execution of the simulation
  - Console
    - Displays the execution log of the process
  - Breakpoints
    - Displays all breakpoints set in the workspace

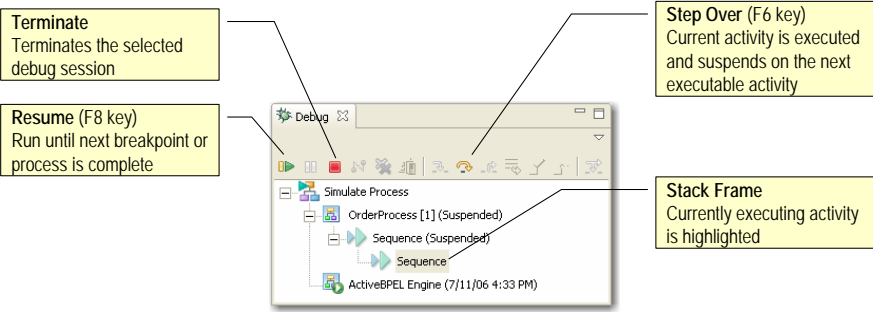
25 Copyright © 2004-2007 Active Endpoints, Inc.



The ActiveBPEL Designer's Debug Perspective introduces the three new views we saw on the previous slide. The Debug View is used to control the execution of the simulation. The Console View displays the log/history of the current execution, scrolling the view as the process executes. Finally, the Breakpoints View allows you to stop execution at any point desired. These breakpoints can be turned on and off as required.

## Debug View

- During simulation, allows you to manage the running or individually stepping through of a BPEL process




The screenshot shows the 'Debug' window with a toolbar and a tree view. The toolbar includes buttons for 'Resume', 'Step Over', and 'Terminate'. The tree view shows a 'Simulate Process' containing an 'OrderProcess [1] (Suspended)' which contains two 'Sequence (Suspended)' elements. The 'ActiveBPEL Engine' is shown at the bottom with a timestamp of 7/11/06 4:33 PM.

**Terminate**  
Terminates the selected debug session

**Resume (F8 key)**  
Run until next breakpoint or process is complete

**Step Over (F6 key)**  
Current activity is executed and suspends on the next executable activity

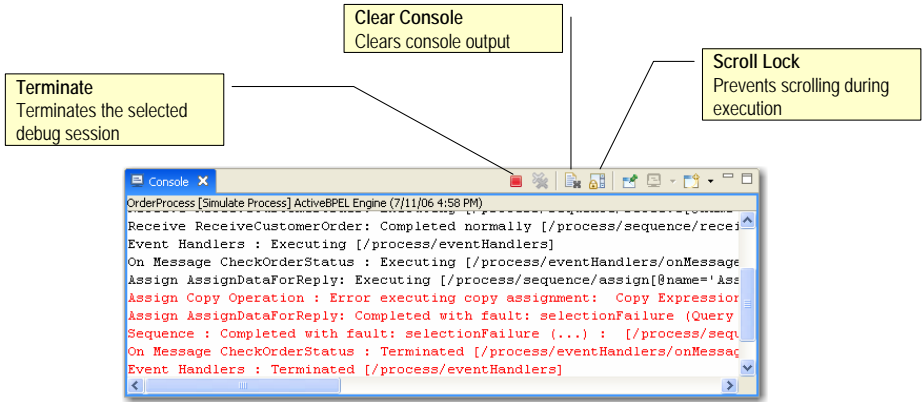
**Stack Frame**  
Currently executing activity is highlighted

26 Copyright © 2004-2007 Active Endpoints, Inc. 

Here, we look at the Debug View in detail. The Resume button will resume the current execution, running either to the next Breakpoint (if there is one) or running to the end of the process. The Step Over button allows you to step through the process activity by activity. The Stack Frame shows the process in the same way as the Outline View, building the treeview out as the process executes. And, finally, the Terminate button allows you to stop (kill) the current debug session immediately.

## Console View

- During simulation, shows execution events
  - Provides a view into the evaluation of expressions and results




The screenshot shows a 'Console' window with the following text:

```
OrderProcess [Simulate Process] ActiveBPEL Engine (7/11/06 4:58 PM)
Receive ReceiveCustomerOrder: Completed normally [/process/sequence/recei
Event Handlers : Executing [/process/eventHandlers]
On Message CheckOrderStatus : Executing [/process/eventHandlers/onMessage
Assign AssignDataForReply: Executing [/process/sequence/assign[@name='Ass
Assign Copy Operation : Error executing copy assignment: Copy Expression
Assign AssignDataForReply: Completed with fault: selectionFailure (Query
Sequence : Completed with fault: selectionFailure (...) : [/process/seq
On Message CheckOrderStatus : Terminated [/process/eventHandlers/onMessa
Event Handlers : Terminated [/process/eventHandlers]
```

Callouts in the image:

- Terminate**: Terminates the selected debug session
- Clear Console**: Clears console output
- Scroll Lock**: Prevents scrolling during execution

27 Copyright © 2004-2007 Active Endpoints, Inc. 

Here is a detailed look at the Console View. From here, we can Terminate the Debug session, clear the console (while still keeping the session active) and the Scroll Lock, which simply keeps the log view from scrolling as the process executes.

## Breakpoints View

- Lists all the breakpoints you have set in BPEL processes across all projects in your workspace
  - Can enable, disable, skip, or remove breakpoints

Removes selected breakpoints

Removes all breakpoints

Enable / Disable breakpoints

Opens the process and navigates to the selected breakpoint

28 Copyright © 2004-2007 Active Endpoints, Inc.

Finally, here is a detailed look at the Breakpoints View, which shows a list of all the breakpoints for the current process. You can remove all or just the selected breakpoints, Enable or Disable selected breakpoints or open the process and navigate to a specifically selected breakpoint.

## Unit Objectives

- At the conclusion of this unit, you will be familiar with:
  - ✓ Simulation overview
  - ✓ Sample Data
  - ✓ ActiveBPEL Debug Perspective
  - Simulating
  - Interacting with a simulation session

## Starting Simulation

- To start simulating your process
  1. Open and select the process you plan to simulate
  2. Click your mouse anywhere in the Process Editor to activate the toolbar
  3. Select the Simulate Process icon on the toolbar
    - If required, ActiveBPEL Debug Perspective will be opened
  4. Simulation stops at the first activity in the process



30 Copyright © 2004-2007 Active Endpoints, Inc.




To simulate a process, open the process and set focus on the Process Editor. You can either use the "Start Simulation of the Process" button or you can use the Run->Simulate Process menu item. If you have the process already in the Debug Perspective, you are fine, but if it is in another perspective it will shift, and open the Debug Perspective. Simulation will be initiated, but will stop just short of the first activity in the process.

## Initial Simulation State

The screenshot displays the ActiveBPEL Designer interface for a process named 'marketplace.bpel'. The process diagram shows a sequence of activities: 'MarketplaceSequence' (highlighted with a blue halo), 'MarketplaceFlow', 'SellerReceive', and 'BuyerReceive'. The console view at the bottom right shows the following log entries:

```
Process marketplace: Instance 1 created.  
Process marketplace: Executing [/process]  
Process Suspended [/process]
```

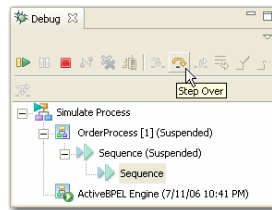
Process variables are listed on the left, including 'sellerInfo', 'negotiationOutcome', and 'buyerInfo'. The console view also shows the process instance has been created and is currently suspended.

31 Copyright © 2004-2007 Active Endpoints, Inc. 

Once the simulation is started (and automatically halted just before execution of the first activity) your process will be in the state shown on the slide. The first (i.e., next) activity is highlighted with a blue halo and the process variables are reset to their pre-initialization states. The console view shows that the process has been instantiated on the server.

## Stepping to the Next Activity in a BPEL Simulation

- You can step through the execution of a BPEL process one activity at a time
  - See the execution results in the Console view
  - Inspect the state of the Process Variables
- Click the Step Over button in the Debug view toolbar (or press the F6 key)
  - Currently selected activity is performed and suspends on the next executable activity



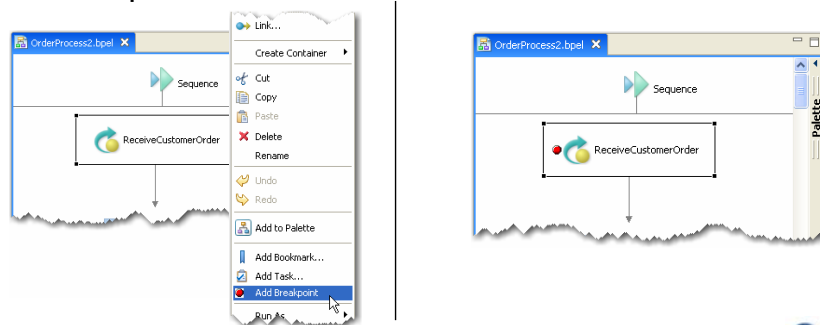
32 Copyright © 2004-2007 Active Endpoints, Inc.



Now you can step through the process activity by activity, as needed. The execution results show in the Console View, along with the process variable's current values.

## Adding a Breakpoint

- Right-click on an activity where you want to set a breakpoint
  - Select Add Breakpoint from the pop-up menu
    - Can select activity from the canvas or from the Outline view
  - Red circle appears next to the activity to indicate the breakpoint

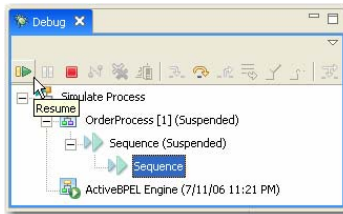


33 Copyright © 2004-2007 Active Endpoints, Inc.

To add a breakpoint to the process, select the activity under scrutiny (in the Outline View or in the Process Editor) and use the Right Mouse context menu to select "Add Breakpoint." A red circle appears next to the activity as a visual cue.

## Running to a Breakpoint

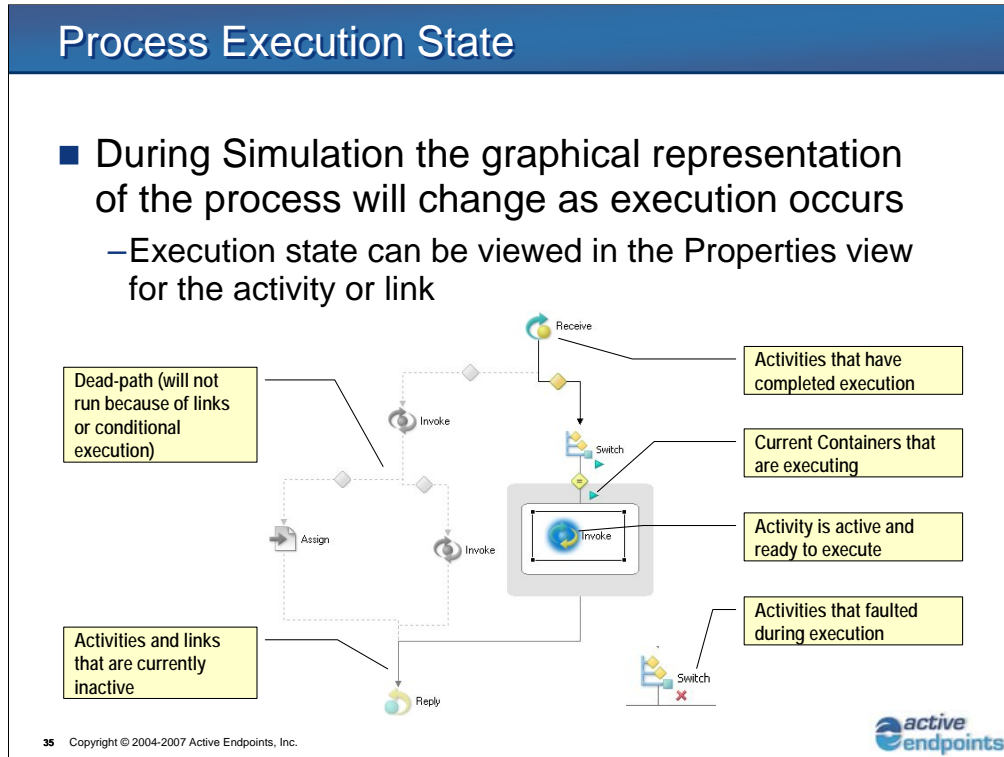
- You can resume execution and run to a breakpoint that you have set on an activity
- Click the Resume button in the Debug view toolbar (or press the F8 key)
  - Process runs to the next breakpoint



34 Copyright © 2004-2007 Active Endpoints, Inc.



During the simulation, you have the option of stepping through the process activity by activity, running to a breakpoint, or terminating the simulation.



While you are simulating, the process diagram will change to reflect the status of the execution. Activities that have completed execution are shown in full color. Currently executing activities and containers are shown with blue arrows, indicating that they have focus. Activities that have not executed yet (and might not, depending on the execution path) are shown in a faded palette. Activities that fault during the execution will be displayed with a red "X" next to them. Dead paths, i.e., those activities and containers that will never execute, are shown grayed out. (The system automatically evaluates the process during execution of the simulation and knows which paths will never be traversed.)

## Ending Simulation

- To end simulation of your process
  - Run or step to the end of the process
  - Explicitly terminate the simulation session

36 Copyright © 2004-2007 Active Endpoints, Inc.



Your simulation can finish by executing all of the process activities (using run or step) or by active termination of the session.

## Clearing Process Execution State

- When a process simulation ends, the graphical highlighting and the simulation properties state are preserved
- In order to clear the graphical highlighting and simulation properties state
  - Click the Clear Execution State icon on the toolbar



37 Copyright © 2004-2007 Active Endpoints, Inc.



Once the process simulation has finished (or halted) it will still be shown using the presentation elements we saw earlier (two slides back). The simulation state is preserved until it is cleared by the user. To restore the process diagram to the pre-simulation state, click on the "Clear Execution State" toolbar button.

## Unit Objectives

- At the conclusion of this unit, you will be familiar with:
  - ✓ Simulation overview
  - ✓ Sample Data
  - ✓ ActiveBPEL Debug Perspective
  - ✓ Simulating
  - Interacting with a simulation session

## Interacting with a Simulation session

- During Simulation you may need to interact with the executing process to
  - Add sample data if it is missing
  - View or modify process variable data
  - Test the handling of a fault returned from an Invoke activity
  - Set the execution path for a Pick Activity
    - Which On Message or On Alarm to execute
  - Execute an Event Handler
    - On Message or On Alarm

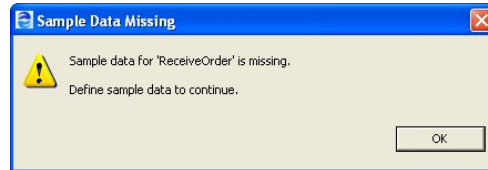
39 Copyright © 2004-2007 Active Endpoints, Inc.



During the process of simulation, it may be necessary to interact with the session in real time. For example, if you are simulating and the process comes to an activity that requires sample data to execute and there is none, the system will advise you of this and will give you the option to do add it on the fly. You can also modify the value of a process variable during the course of a simulation, and/or test the handling of a Fault that is returned from an Invoke activity. Finally, because ActiveBPEL has Pick activities and Event Handlers, which can receive one of many different inputs, you can manipulate the process to select which of the various options you would like to execute during the session. For example, a Pick activity may have three different onMessage activities. In that situation, you can pick which onMessage activity to execute.

## Variable Interaction

- Simulation will stop if a required sample file is not loaded
  - Load the sample data using either Web References, process variables, or simulation properties



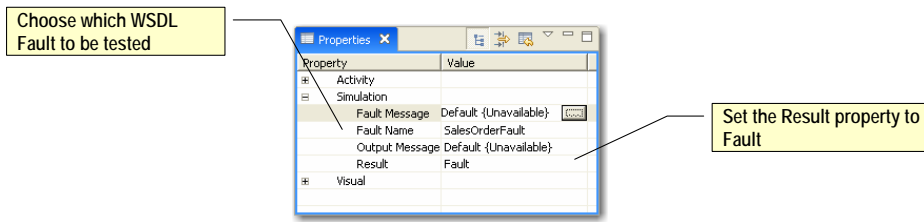
40 Copyright © 2004-2007 Active Endpoints, Inc.



If an activity requires that Sample Data be provided and you have not done so, the system will stop the execution at that point in the process and prompt you to add the necessary data.

## Testing Invoke Fault Handling

- Simulation properties can be set to mimic a WSDL Fault message being returned from an invoke activity
  - Sample data for the Fault message (if required) will need to be provided



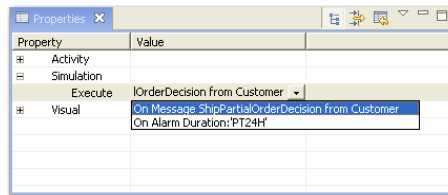
41 Copyright © 2004-2007 Active Endpoints, Inc.



When you are simulating a Fault for a particular activity, you must first set the Result value to "Fault" and then select the fault from the drop-down list. (Note that this list will only show the Fault messages that have been defined for that message.) If you are going to generate a fault, there must be sample data, as required.

## Testing Pick Activities

- A Simulation property is used to indicate what event is being tested
  - A dialog will be presented during simulation if no selection has been made



42 Copyright © 2004-2007 Active Endpoints, Inc.



As with an onMessage, with a Pick activity we have multiple options that can be fired, depending on the needs of the simulation. You can set focus on the activity and then go to the Properties View to select the onMessage or onAlarm event you'd like to test.

## Testing Event Handlers

- To test a specific event the Event Handler must be in an executing state
  - Recall that
    - Process Event Handlers are executing after the Create Instance activity is executed
    - Scope Event Handlers are executing after the Scope activity is executed

43 Copyright © 2004-2007 Active Endpoints, Inc.

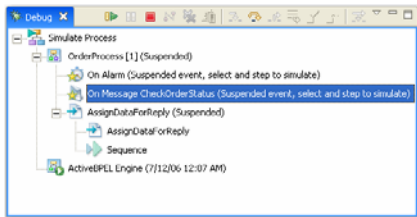


Remember that an Event Handler can be installed at the process level. In such a case, the instance must be created before any onEvent activity can fire. When an Event Handler is added at the scope level, the scope must be active before the onEvent can be fired.

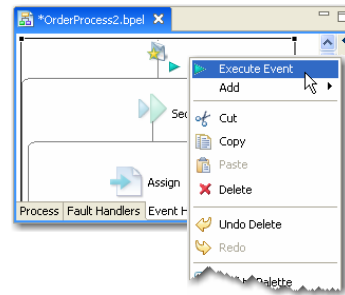
## Executing an Event Handler

- Once active, each event handler can be executed

- Debug View



- Event Handlers



Once the Event is active, it can be fired from either the Outline View or from the Process Editor. In either case, use the Right Mouse context menu to do so.

## Lab 3 & 4 – Process Simulation

- Overview of Lab Exercises
  - Add and load sample data
  - Simulate the Order Process

45 Copyright © 2004-2007 Active Endpoints, Inc.



The next Labs in the BPEL Fundamentals class are Labs #17 & #18. (Note: This are labs #3 & #4 if you are taking the BPEL Intermediate course.) In the first lab we'll load Sample Data from both the Web References View and from the Process Variables View. In the second lab we'll simulate the "Order Process" process.

## Unit Summary

- Now you are familiar with:
  - Simulation overview
  - Sample Data
  - ActiveBPEL Debug Perspective
  - Simulating
  - Interacting with a simulation session