



This is Unit #7 of the BPEL Fundamentals course. In past Units we looked at ActiveVOS Designer, Workspaces and Projects and then we created the Process itself. In the last two units we looked at global declarations and how to create Interaction Activities like Receives, Replies and Invokes. So, now let's take a look at a new process activity, the Sequence.

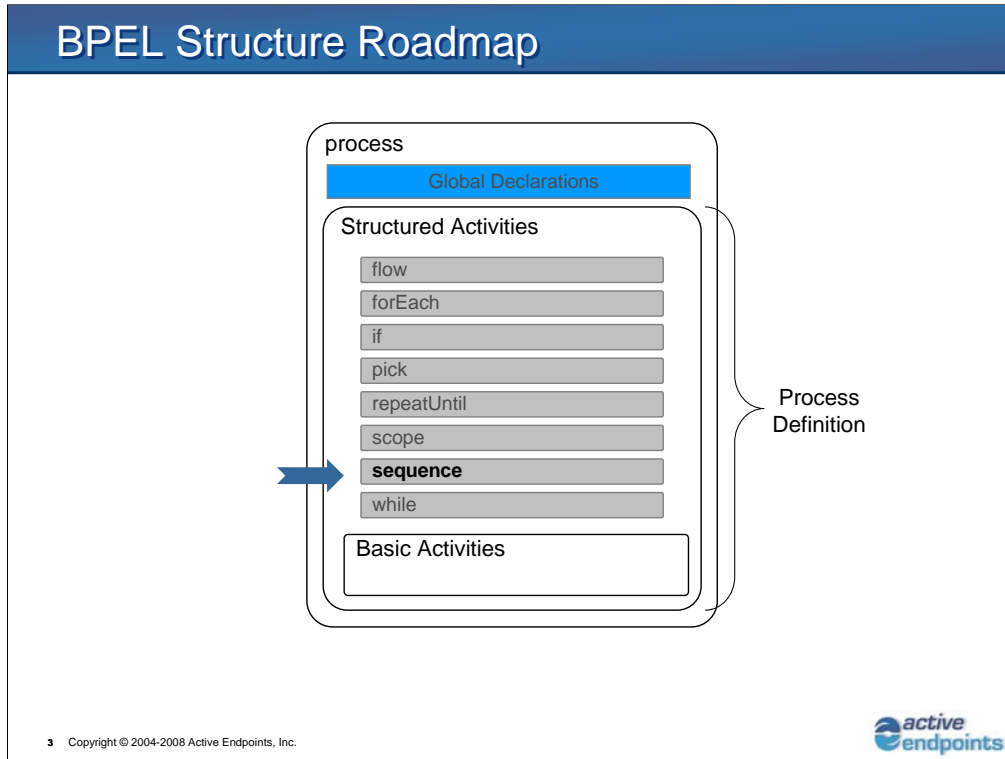
Unit Objectives

- At the conclusion of this unit, you will be familiar with:
 - sequence activity
 - Working with activities which can contain other activities

2 Copyright © 2004-2008 Active Endpoints, Inc.



In this Unit we'll look at the Sequence activity, which is our first container activity, meaning that it is an activity that can contain other activities.



Looking at our BPEL Structure Roadmap, we see that the Sequence activity is the first of our structured activities and is part of the process definition. So, how do we use them?

sequence Activity Overview and Syntax

- Used to define a set of activities to be executed in a strict ordered sequence
 - Can contain both basic and structured activities nested to arbitrary depth

```

<sequence standard-attributes>
  standard-elements
  activity+
</sequence>
```

4 Copyright © 2004-2008 Active Endpoints, Inc.



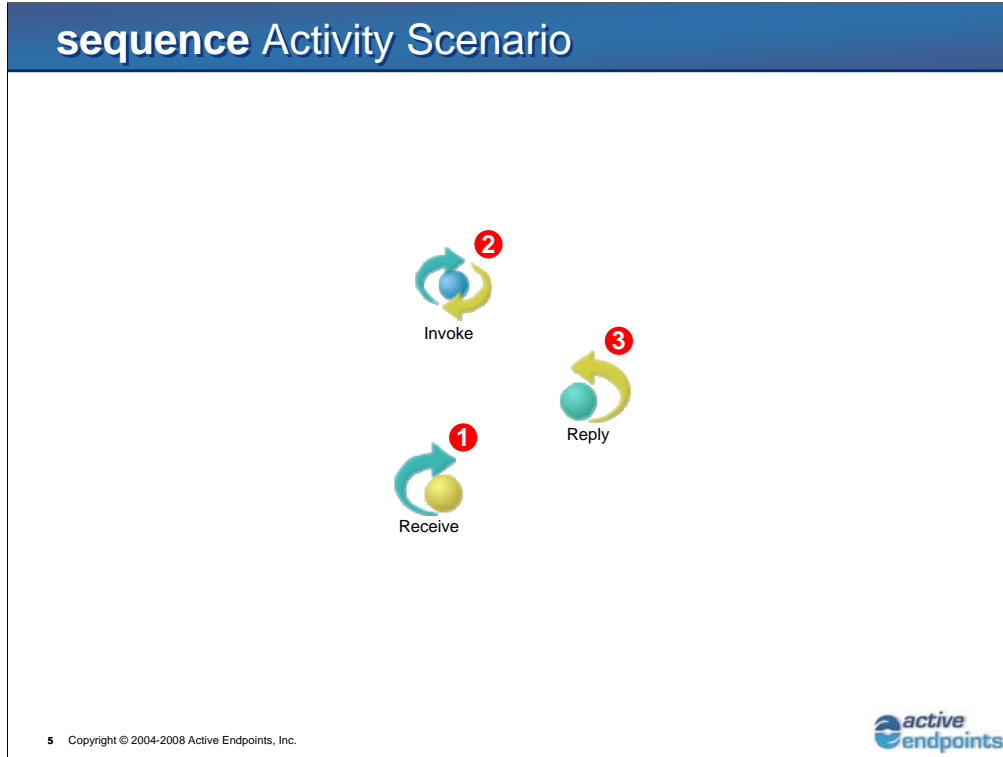
A **Sequence** is a structured activity which can contain other activities, all of which will be executed in a specifically defined order. The purpose of a Sequence, therefore, is to define the execution order for a group of activities. BPEL Sequences can contain other Sequences and can be nested to an arbitrary depth. Sequences have all the standard attributes and elements and they must contain at least one or more activities (note the “+” symbol.)

For reference purposes, here are some of the element delcaration symbols used to define how many times (if at all) that an item appears.

? = 0 or 1 (i.e., use is optional)

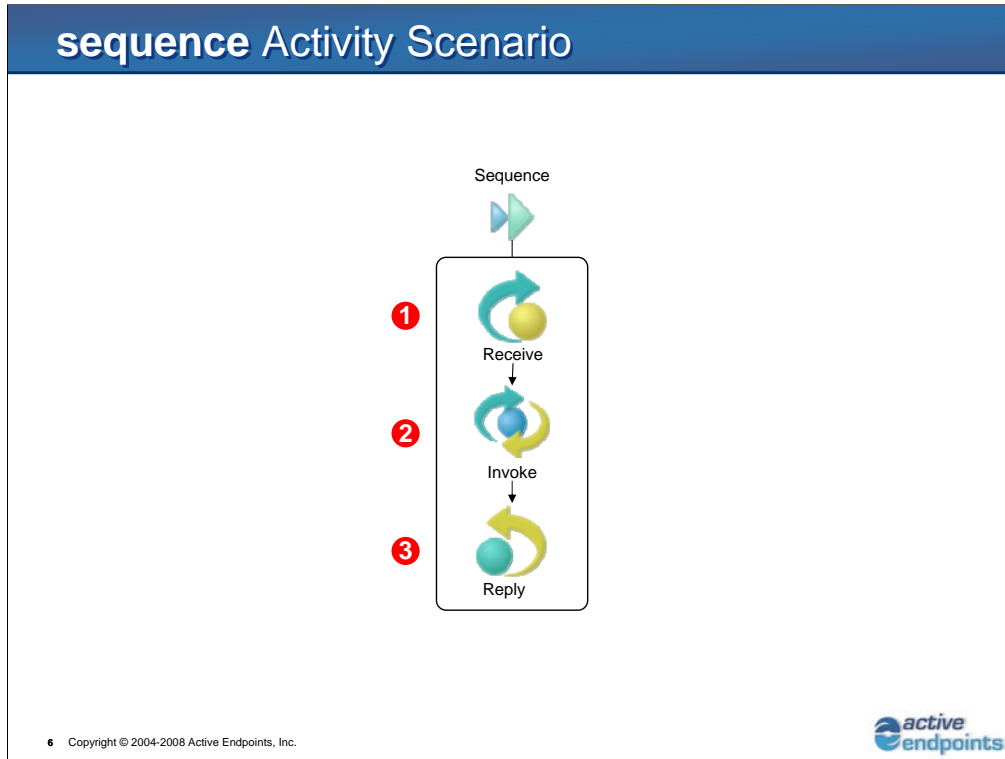
* = 0 or 1 or more (i.e., use is optional)

+ = 1 or more (i.e., use is required)



Here we have three interaction activities without a container. So, which is first? Which one is next? And, which one is last? If we want to see the Receive execute first, then the Invoke and finally the Reply, then we must put them inside a container. We will use a Sequence container to set the order of execution.

Note: If you drop two or more unordered activities onto the canvas, ActiveVOS Designer will automatically create a Flow activity in the underlying code in order to make the process legal and executable. If you want to see this, go to your project, open the BPEL process file and switch to the Source tab of the Process Editor. At this point there are two process activities, a Receive and a Reply. Notice that these two activities have been wrapped inside a Flow activity. We'll be discussing the Flow activity in a later unit.



Here the three activities have been put into a Sequence and they will be executed in the order shown: Receive, Invoke and Reply.

sequence Activity Example

```
<sequence>  
  <receive ... />  
  <invoke ... />  
  <reply ... />  
</sequence>
```

7 Copyright © 2004-2008 Active Endpoints, Inc.



Here is the syntax for the previous Sequence.

sequence Activity Semantics

- Contained activities are executed in the order in which they are listed
 - One of the few BPEL constructs where element order matters
- Completes when the last activity has completed
- Links between the activities are implicit and cannot be configured

© Copyright © 2004-2008 Active Endpoints, Inc.



Activities contained in a Sequence are executed in the precise order in which they are listed, one of the few BPEL constructs where order matters. When the last activity in the Sequence is complete, then the Sequence itself is considered to be complete. Note that the Links between a Sequence's activities are implicit, and are not configurable. Some other BPEL links are configurable, as we'll see later on.

Unit Objectives

- At the conclusion of this unit, you will be familiar with:
 - ✓ *sequence activity*
 - Working with activities which can contain other activities

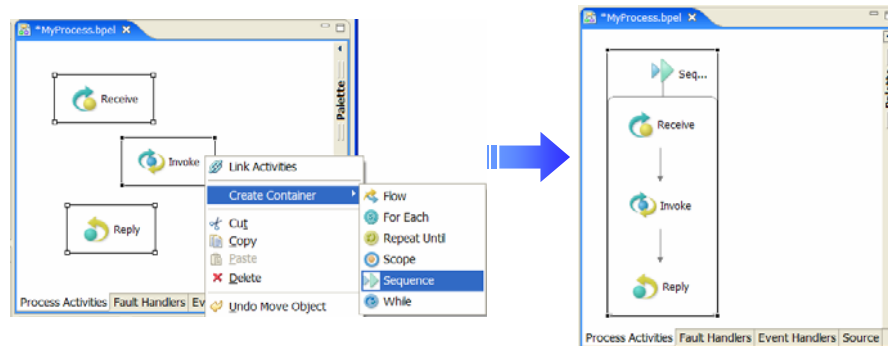
© Copyright © 2004-2008 Active Endpoints, Inc.



Now that we know what a Sequence is, let's look at how we work with our Sequences in the Designer's GUI.

Adding Activities for Containment

- Allows for a convenient way to automatically put a selection of activities within certain containers
 - Works with `sequence`, `scope`, `while`, `repeatUntil`, `forEach` activities



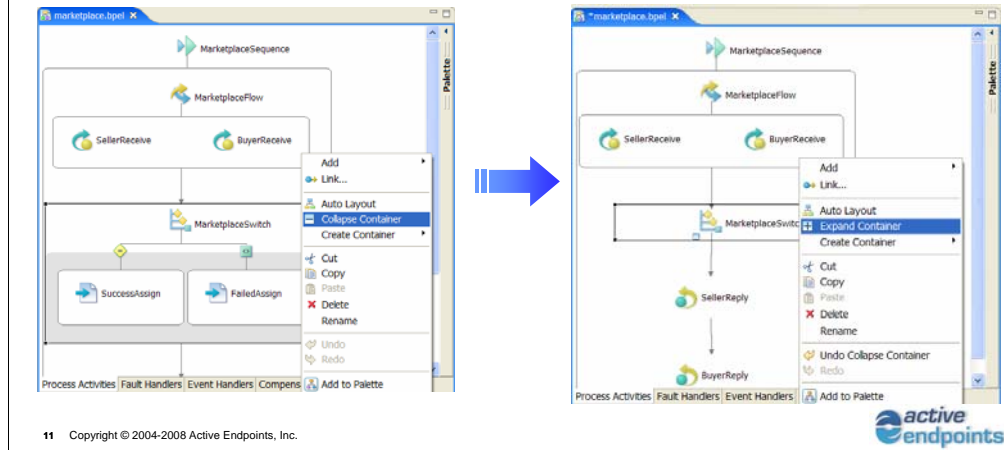
10 Copyright © 2004-2008 Active Endpoints, Inc.



Starting with the diagram on the left, we have three activities on our canvas (Receive, Invoke & Reply) but with no set order. You can select them as an *ordered group* by holding down the Control key and then selecting each of them in the desired execution order. Then, use the Right Mouse context menu and select Create Container->Sequence. Now they are inside the Sequence container and will execute in the proper order. Once a container and its activities are defined, you can drag the activities up or down to change their order within the Sequence. You can also use the Auto Layout button on the toolbar to clean up the display when you are done. The Sequence activity can be combined with other containers and activities, as we will see in later units.

Expanding and Collapsing Container Activities

- Provides better management of screen real estate by collapsing container activities
 - e.g., sequence, flow, scope, while

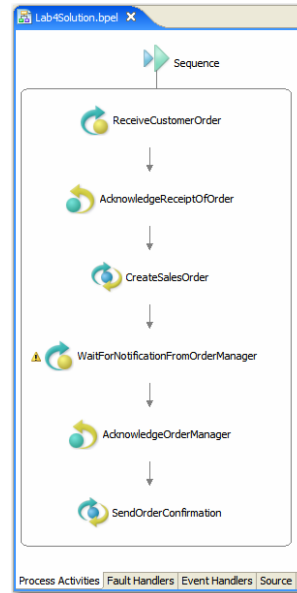


Containers can contain an arbitrary number of activities and can take up lots of room on the canvas. If we *collapse* them we will see the top-level container only, rendered as a simple box. Here on the left side of the slide we have a “MarketplaceSwitch” IF statement, which is fully expanded. On the right side of the slide, the “MarketplaceSwitch” IF statement has been collapsed and only the top-level IF icon is seen. We can collapse any BPEL container, or *expand* them as needed, using the Right Mouse context menu. (This menu item works as a toggle.) Note also that Sequences, just like all of our BPEL activities, can be renamed to something more meaningful, which can help make the process diagram more easily understood.

Lab 4 – Interaction Activities

■ Overview of Lab Exercises

- Add sequence activity
- Add invoke
 - create a sales order (manually)
 - send order confirmation (Interfaces view using the portType)
- Add receive-reply activities to wait for order confirmation
 - (Interfaces view using the partnerLinkType)



12 Copyright © 2004-2008 Active Endpoints, Inc.



The next Lab in the BPEL Fundamentals class is Lab #4. In this lab we will enhance our business process by invoking a service that will enter the customer's order into our internal order management system. The process will wait until that service completes, at which time it will resume execution and send an order confirmation back to the customer. Recall that in Lab #3 we added two activities to our new process: `ReceiveCustomerOrder` and `AcknowledgeReceiptOfOrder`. Today we will start Lab #4 by putting those activities in a `Sequence` container that will control their order of execution. Then we'll add some process variables to hold our data and finally, we'll add activities that will `CreateSalesOrder`, `WaitForNotification` of that Order from the Order manager and then `SendOrderConfirmation` back to the Customer. When you are finished your process diagram will look like the one on the right hand side of the slide. One tip that will let you see the process interaction more clearly is to go into the `Process` menu and select "View Swimlanes" and choose one or more `partnerLinks`. This will show the back and forth of the process-partner conversation and might help you better understand the information flow.

Unit Summary

- Now you are familiar with:
 - sequence activity
 - Working with activities which can contain other activities