FLEX - EVENT HANDLING

Flex uses concept of event to pass data from one object to another depend upon the state or user interaction within the application.

ActionScript has a generic **Event** class which defines much of the functionality needed to work with events. Every time an event occurs within a Flex application, three types of objects from the Event class hierarchy are created.

Event has the following three key properties

Property	Description
type	type states about what kind of event just happened. This may be click, initialize, mouseover, change, etc. The actual values will be represented by constants like MouseEvent.CLICK.
target	The target property of Event is an object reference to the component that generated the event.If you click a Button with an id of clickMeButton, the target of that click event will be clickMeButton
currentTarget	The currentTarget property varies container hierarchy. It mainly deals with flow of events.

Event Flow Phases

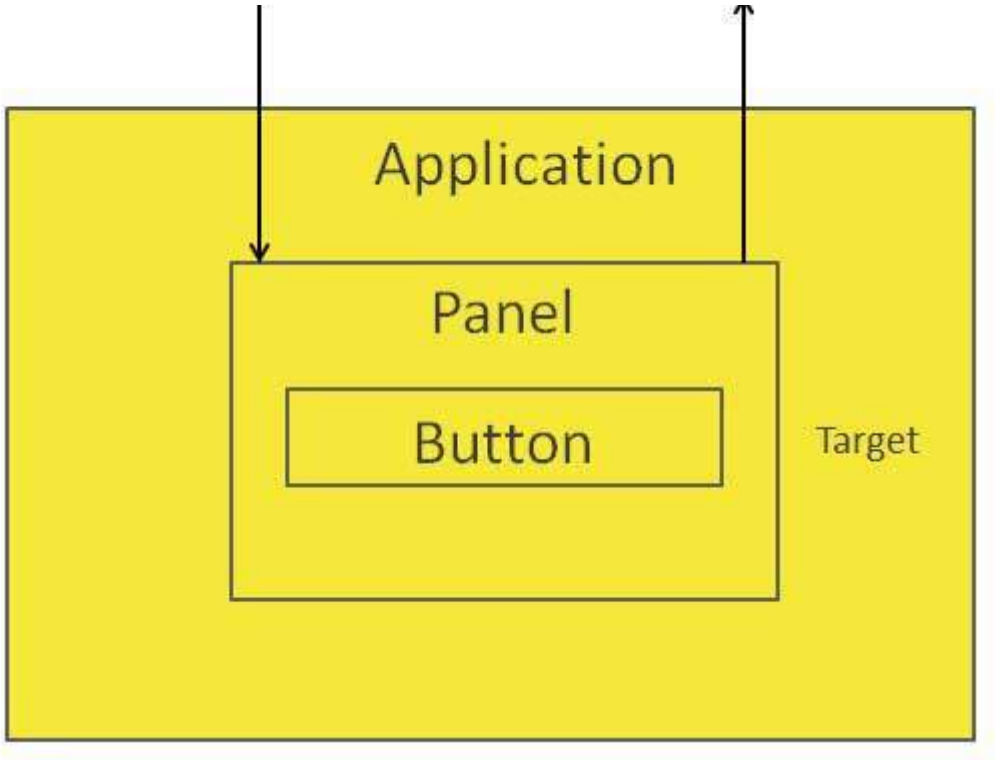
An event goes through three phases looking for event handlers.

Phase	Description
Capture	In the capture phase the program will start looking for event handlers from the outside *ortop* parent to the innermost one. The capture phase stops at the parent of the object that triggered the event.
Target	In the target phase ,the component that triggered the event, is checked for an event handler.
Bubble	The Bubble phase is reverse of capture phase, working back through the structure, from the target component's parent on up.

Consider the following application code

```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
   xmlns:s="library://ns.adobe.com/flex/spark"
   xmlns:mx="library://ns.adobe.com/flex/mx"
   width="100%" height="100%"
   minWidth="500" minHeight="500" >
   <s:Panel>
      <s:Button />
   </s:Panel>
</s:Application>
```

When the user clicks the Button, he or she has also clicked the Panel and the Application.The event goes through three phases looking for event-handler assignments.



Let us follow the following steps to test event handing in a Flex application:

Step	Description
1	Create a project with a name *HelloWorld* under a package *com.tutorialspoint.client* as explained in the *Flex - Create Application* chapter.
2	Modify *HelloWorld.mxml* as explained below. Keep rest of the files unchanged.
3	Compile and run the application to make sure business logic is working as per the requirements.

Following is the content of the modified mxml file **src/com.tutorialspoint/HelloWorld.mxml**.

```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
   xmlns:s="library://ns.adobe.com/flex/spark"
   xmlns:mx="library://ns.adobe.com/flex/mx"
   width="100%" height="100%" minWidth="500" minHeight="500"
   >
   <fx:Style source="/com/tutorialspoint/client/Style.css"/>
   <fx:Script>
      <![CDATA[
         protected function reportEvent(event:MouseEvent):void
         {
            var target:String = event.target.id;
            var currentTarget:String = event.target.id;
            var eventPhase: String;

            if(event.target is Button){
               var button:Button = event.target as Button;
               target = button.label + " Button";
            } else if(event.target is HGroup){
               var hGroup:HGroup = event.target as HGroup;
               target = hGroup.id + " HGroup";
            }else if(event.target is Panel){
               var panel:Panel = event.target as Panel;
               target = panel.id + " Panel";
            }
```

```
            if(event.currentTarget is Button){
               var button1:Button = event.currentTarget as Button;
               currentTarget = button1.label + " Button";
            }else if(event.currentTarget is HGroup){
               var hGroup1:HGroup = event.currentTarget as HGroup;
               currentTarget = hGroup1.id + " HGroup";
            }else if(event.currentTarget is Panel){
               var panel1:Panel = event.currentTarget as Panel;
               currentTarget = panel1.id + " Panel";
            }

            var eventPhaseInt:uint = event.eventPhase;

            if(eventPhaseInt == EventPhase.AT_TARGET){
               eventPhase = "Target";
            } else if(eventPhaseInt == EventPhase.BUBBLING_PHASE){
               eventPhase = "Bubbling";
            }else if(eventPhaseInt == EventPhase.CAPTURING_PHASE){
               eventPhase = "Capturing";
            }

            reports.text += " Target: " + target + "\n currentTarget: " +
            currentTarget + "\n Phase: " + eventPhase + "\n----------\n";
         }
      ]]>
   </fx:Script>
   <s:BorderContainer width="630" height="480"
      styleName="container">
      <s:VGroup width="100%" height="100%" gap="10"
         horizontalAlign="center" verticalAlign="middle">
         <s:Label
            fontSize="40" color="0x777777" styleName="heading"/>
         <s:Panel
            click="reportEvent(event)" width="500"
            height="100" includeInLayout="true" visible="true">
            <s:layout>
               <s:VerticalLayout  gap="10"
                  verticalAlign="middle" horizontalAlign="center"/>
            </s:layout>
            <s:HGroup >
               <s:Button label="Click Me"
                  click="reportEvent(event)"/>
            </s:HGroup>
         </s:Panel>
         <s:Panel
            title="Events" width="500" height="230">
            <mx:Text  />
         </s:Panel>
      </s:VGroup>
   </s:BorderContainer>
</s:Application>
```

Once you are ready with all the changes done, let us compile and run the application in normal mode as we did in Flex - Create Application chapter. If everything is fine with your application, this will produce following result: [Try it online]