

Implementing Drag and Drop between Components

Note: The starter and solution files for this project do not include the LiveCycle Data Services technology that you implemented in Exercise 9. Removing that code simplifies the setup for this and the following exercises. However, this means that the form submission functionality does not work. To implement the form submission with remote object, follow the steps in Exercise 9 and watch the associated video.

Create drag action on a List control

1. Import **Ex10_Starter.zip** into the **adobeFlexTraining** workspace to see the **AdobeODT** project.

Note: You can alternatively start with the files you completed in Exercise 9.

2. In **AdobeODT.mxml** locate the `Script` block.
3. After the last import statement, import the `IUIComponent`, `DragManager` and `DragEvent` classes.

```
import mx.core.IUIComponent;
import mx.managers.DragManager;
import mx.events.DragEvent;
```

4. Using Outline view, locate the **List dg** control.
5. Add the `dragEnabled` property with a value of `true`.

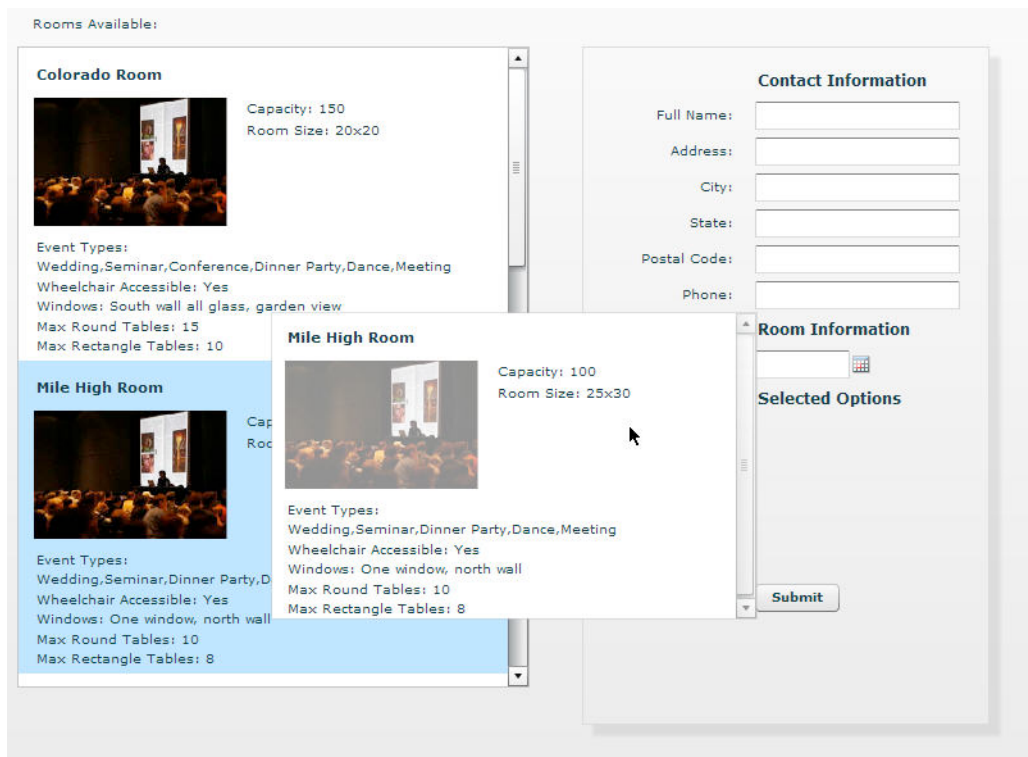
```
<mx:List id="dg"
  x="0" y="142"
  dataProvider="{roomList}"
  itemRenderer="components.RoomRenderer"
  labelField="name"
  width="385" height="501"
  dragEnabled="true"></mx:List>
```

6. Save the file and run the application.

Note: If you receive an error message stating that there is a security setting violation or that the application can't access a resource or asset, then you will need to add a compiler setting that will allow you to access resources from the local file system. Use the following steps to add the compiler setting.

- a. Right click on the **AdobeODT** project and select **Properties** from the menu.

- b. Select the **Flex Compiler** menu.
 - c. Add a space and then `-use-network=false` to the compiler arguments.
 - d. Click OK.
 - e. Run `AdobeODT.mxml` again.
7. Click and drag a room from the `List` control to the empty section of the form on the right and release your mouse.
- You should see the drag data return to the `List` control because the form is currently not set up to accept the data.



Create a property in the custom component

1. In the `ReservationForm.mxml` file in the `components` directory, locate the `Script` block.
2. If not already created, create a `Bindable` public variable named `selectedRoom` datatyped as `Object`.

```
[Bindable] public var selectedRoom:Object;
```

3. Save the file.

Create drop action on custom component

1. In the `AdobeODT.mxml` file, locate the `ReservationForm` component instance.

2. Add an `id` property with a value of `rf`.
3. Change the `width` property value to 338 and the `height` property value to 557.
4. Add the `dragEnter` event with a value of `doDragEnter(event)`.
5. Add the `dragDrop` event with a value of `doDragDrop(event)`.

```
<comp:ReservationForm id="rf"
    x="441" y="142"
    width="338" height="557"
    selectedOptions="{selectedOptions}"
    dragEnter="doDragEnter(event)"
    dragDrop="doDragDrop(event)" />
```

6. Locate the `Script` block.
7. Before the end of the `Script` block, create a private function named `doDragEnter` that takes one parameter named `event` and data type it to the `DragEvent` class. The function returns `void`.

```
private function doDragEnter(event:DragEvent):void {
}
```

8. Within the function, invoke the `acceptDragDrop()` method of the `DragManager` class. Cast the `event.target` as `IUIComponent` and pass the parameter to the method.

```
DragManager.acceptDragDrop(event.target as IUIComponent);
```

9. Before the end of the `Script` block, create a private function named `doDragDrop` that takes one parameter named `event` and data type it to the `DragEvent` class. The function returns `void`.

```
private function doDragDrop(event:DragEvent):void {
}
```

10. Within the function create and instantiate a local variable named `dropData` to the `Object` class.

```
var dropData:Object = new Object();
```

11. Use the `dataForFormat('items')` method of the `event.dragSource` and assign the data to `dropData`. The `dragSource` contains an `Array` with an object named `items` at the zero index.

```
dropData = event.dragSource.dataForFormat('items');
```

12. Assign the zero index of the `dropData` `Array` to the `selectedRoom` property of `rf`.

```
rf.selectedRoom = dropData[0];
```

13. Save the file and run the application.
14. Click and drag a room from the `List` control to the form and release your mouse.

You should see that the form now accepts the drag data but the data disappears.

Modify the Form to display data from the drag event

1. Return to the **ReservationForm.mxml** file.
2. Locate the `FormItem` tag block that contain the `dateNeeded DateField` instance.
3. After the closing `FormItem` tag add a `FormItem` tag block with a `label` property set to a value of `Event Type:`.
4. Within the `FormItem` tag block create a `ComboBox` control with an `id` property of `eventtype`. Add the `dataProvider` property with a value bound to `selectedRoom.eventtypes.event`. Add a `prompt` property with a value of `Select`.

```
<mx:FormItem label="Event Type:">
    <mx:ComboBox id="eventtype"
        dataProvider="{selectedRoom.eventtypes.event}"
        prompt="Select"/>
</mx:FormItem>
```

5. Add another `FormItem` tag block with a `label` property set to a value of `Room Name:`.
6. Within the `FormItem` tag block add a `Text` control with an `id` property set to a value of `roomname`. Add a `text` property with a value bound to `selectedRoom.roomname`.

```
<mx:FormItem label="Room Name:">
    <mx:Text id="roomname" text="{selectedRoom.name}"/>
</mx:FormItem>
```

7. Add a `FormItem` tag block with a `label` property set to a value of `Capacity:`.
8. Within the `FormItem` tag block, create a `Text` control with an `id` of `capacity` and a `text` property bound to `selectedRoom.capacity`.

```
<mx:FormItem label="Capacity:">
    <mx:Text id="capacity" text="{selectedRoom.capacity}"/>
</mx:FormItem>
```

9. Add a `FormItem` tag block with a `label` property set to a value of `Room Size:`.
10. Within the `FormItem` tag block create a `Text` control with an `id` of `roomsized` and a `text` property bound to `selectedRoom.roomsized`.

```
<mx:FormItem label="Room Size:">
    <mx:Text id="roomsized" text="{selectedRoom.roomsized}"/>
```

</mx:FormItem>

11. Save the file and run the application.

Click and drag a room from the List control to the form and release. You should see the form display the room name, capacity and room size from the drag event.

XYZ Convention Center

111 49th Ave, Denver, CO 80634 Phone: 303-555-1212 Fax: 303-555-1234


Room Reservation System

Please select any additional amenities needed for your room. Our staff will contact you to finalize your reservation.

Microphone Dance Floor Food Station Security Extra Trash Receptacals
 Projector Special Lighting Color Table Linens Waitstaff Additional Electrical Outlets

Rooms Available:


Colorado Room



Capacity: 150
Room Size: 20x20

Event Types:
Wedding, Seminar, Conference, Dinner Party, Dance, Meeting
Wheelchair Accessible: Yes
Windows: South wall all glass, garden view
Max Round Tables: 15
Max Rectangle Tables: 10

Mile High Room



Capacity: 100
Room Size: 25x30

Event Types:
Wedding, Seminar, Dinner Party, Dance, Meeting
Wheelchair Accessible: Yes
Windows: One window, north wall
Max Round Tables: 10
Max Rectangle Tables: 8

Contact Information

Full Name: *

Address: *

City: *

State: *

Postal Code: *

Phone: *

Room Information

Date Needed: *

Event Type: ▼

Room Name: Colorado Room

Capacity: 150

Room Size: 20x20

Selected Options: Dance Floor
Special Lighting