

# Communicating Over Local Connections

---

## Import the Project

1. If not already created, create a directory named `adobeFlexTraining` on your **C** drive.
2. In Flex Builder, delete the **ReservationSystem** project if created previously. If you don't delete the contents the files will be overwritten with the new project files.
3. In Flex Builder, select **File > Import > Flex Project**.
4. In the dialog window, select **Archive File** and browse to where `Ex23_Starter.zip` is located in your local file system.  
**Note:** This starter file has the code you created through the end of Day 4 exercises. This exercise will only delve into a portion of that code. For more information, review Day 4 exercises.
5. Uncheck **Use default location**.
6. Enter `C:\adobeFlexTraining\ReservationSystem`.
7. Click **Finish**.

## Create the local connection sender

8. Using the **Flex Navigator** view, open the `RequestTicket.mxml` located in the **ReservationSystem > src > components** folder.
9. At the beginning of the code block within the `<mx:Script>` tags, import the `flash.net.LocalConnection` class.
10. Below the imported class, declare a private variable named `conn` with a datatype set to the `LocalConnection` class.
11. Before the function in the `Script` block, create a private function named `initConn` that takes no parameters and returns `void`.
12. Within the `initConn()` function, create a new `LocalConnection` assigned to the `conn` instance.

Your code should appear as follows.

```
private function initConn():void
{
    conn = new LocalConnection();
```

```
}
```

13. Below the `initConn()` function, create a private function named `sendNotification` that takes one parameter named `event` with a datatype set to the `MouseEvent` class. The function returns `void`.
14. Within the `sendNotification()` function declare a local variable named `message` with a datatype of `String`.
15. Set the variable equal to a string that displays "[Name] has requested a room reservation for [room name] on [date]." Replace the Name, room name and date values with the text property of the `fullname`, `roomName` and `dateNeeded` variables, respectively."
16. Use the `send` method of the `conn` instance and pass it the arguments: The connection name: `_myConnection`. The function in the receiver that handles the processing of the message: `localConnectionHandler`. The actual message that is passed as the string to the receiver: `message`.

Your code should appear as follows.

```
<mx:Script>
  <![CDATA[

      import flash.net.LocalConnection;

      private var conn:LocalConnection;

      private function initConn():void
      {
          conn = new LocalConnection;
      }

      private function
sendNotification(event:MouseEvent):void
      {
          var message:String;
          message = fullname.text + " has requested a room
reservation for " + roomName.text + " on " + dateNeeded.text;
          conn.send("_myConnection",
"localConnectionHandler", message);
      }

  ]]>
</mx:Script>
```

17. Using the **Outline** view locate the `{selectedRoom.name}` text property located inside the `FormItem` labeled `Room`.
18. Add an `id` of `roomName` to the `Text` tag.

Your code should appear as follows.

```
<mx:Text id="roomName"
  text="{selectedRoom.name}"
  fontWeight="bold" fontSize="12"
  x="0" y="110"/>
```

19. Add the `creationComplete="initConn()"` property to the beginning `Canvas` tag at the top of the file.

Your code should appear as follows.

```
<mx:Canvas xmlns:mx="http://www.adobe.com/2006/mxml"
  width="100%"
  xmlns:comp="components.*"
  creationComplete="initConn()"
  backgroundColor="#FFFFFF" >
```

20. Using the **Outline** view locate the `submit` button nested inside the `FormItem`.
21. In the `<mx:Button>` tag for the `submit` button add the `click` property that invokes the `sendNotification()` function.

Your code should appear as follows.

```
<mx:FormItem>

  <mx:Button id="button"
    label="{resourceManager.getString('localization','submit_button')}"
    click="sendNotification(event)"/>

</mx:FormItem>
```

22. Save the file.

## Create the local connection receiver

23. Create a new project in Flex Builder by selecting **File > New > Flex Project**.

24. Enter DeskTopNotification for the **Project name**.
25. Uncheck **Use default location**.
26. Enter C:\adobeFlexTraining\DeskTopNotification.
27. Choose **Desktop application** for the **Application Type**.
28. Click **Finish** to create the **Air** application.
29. Open the DeskTopNotification.mxml file.
30. In beginning the <mx:WindowedApplication> tag, add the creationComplete property having a value of initConn().

Your code should appear as follows.

```
<mx:WindowedApplication
xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute"
creationComplete="initConn()">
```

31. Between the WindowedApplication tags add a <mx:Script> block.
32. Import the flash.net.LocalConnection class.
33. Declare a private variable named conn with a datatype set to the LocalConnection class.
34. Create a private function named initConn that takes no parameters and returns void.
35. Within the initConn() function, instantiate the LocalConnection class and assign it to the conn variable.
36. Invoke the client method on the conn instance to register this local connection as the client.
37. Invoke the connect() method on the conn instance and pass it the connection name \_myConnection as the argument.

Your code should appear as follows.

```
private function initConn():void
{
    conn = new LocalConnection;
    conn.client = this;
    conn.connect("_myConnection");
}
```

38. Below the initConn() function, create a public function named LocalConnectionHandler taking one parameter named msg with a datatype set to the String class. The function returns void.

39. Within the `localConnectionHandler()` function, assign the string `msg` passed to the function to the `text` property of the `txtArea` component. Add a newline to the string as well.

Your complete code should appear as follows.

```
<mx:Script>
    <![CDATA[

        import flash.net.LocalConnection;

        private var conn:LocalConnection;

        private function initConn():void
        {
            conn = new LocalConnection;
            conn.client = this;
            conn.connect("_myConnection");
        }

        public function localConnectionHandler(msg:String):void
        {
            txtArea.text = txtArea.text + msg + "\n";
        }

    ]]>
</mx:Script>
```

40. Between the ending `<mx:Script>` tag and the ending `<mx:WindowedApplication>` tag, create an `<mx:TextArea>` component with the following properties.

- `id = txtArea`
- `x = 139`
- `y = 192`
- `width = 178`

41. Below the `<mx:TextArea>` add a `<mx:Label>` component with the following properties.

- `x = 139`
- `y = 139`
- `text = Room Request Notification`
- `width = 221`
- `height = 29`
- `fontWeight = bold`

- `fontSize = 12`

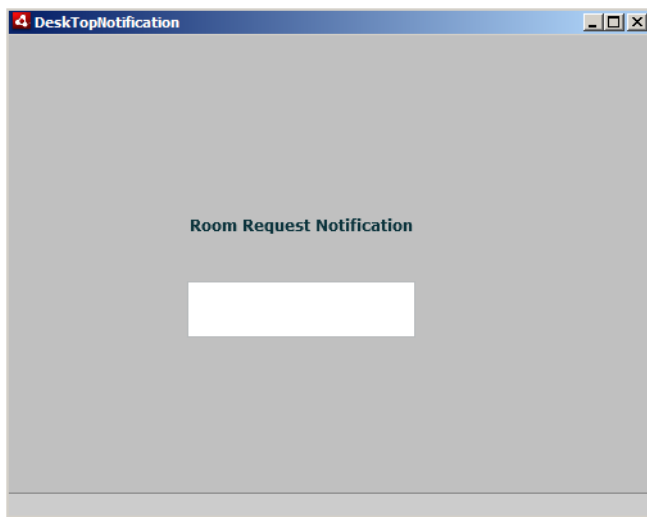
Your code should appear as follows.

```
<mx:TextArea id="txtArea"
  x="139" y="192" width="178"/>

<mx:Label x="139" y="139"
  text="Room Request Notification"
  width="221" height="29"
  fontWeight="bold" fontSize="12"/>
```

42. Save and run the file.

You should see a dialog box on your desktop with a label of Room Reservation Notification and a blank text area.



43. Return to Flex and run the `ReservationSystem.mxml` file.

After the login page, drag a room type into the Request Ticket form and fill out the form. Make sure to fill out the Name and Date for the reservation. If you click on the Submit button, a notification will be visible in the text area of the dialog box for Room Reservation Notification.

