

Using Shared Objects

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 `Ex24_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**.

Import the Air application

8. If created, delete the **DeskTopNotification** project. You may keep the files. They will be overwritten.
9. In Flex Builder, select **File > Import > Flex Project**.
10. In the dialog window, select **Archive File** and browse to where `Ex24AIR_Starter.zip` is located in your local file system.
11. Uncheck **Use default location**.
12. Enter `C:\adobeFlexTraining\DeskTopNotification`.
13. Click **Finish**.

Add the save and delete functionality to the receiver

14. Open the **DeskTopNotification** project in Flex Builder.
15. Open the `DeskTopNotification.mxml` file.
16. In the **Script** block, import the `flash.net.SharedObject` class.

Your code should appear as follows:

```
import flash.net.SharedObject;
```

17. Declare a private variable named `sharedObj` with a datatype set to the `SharedObject` class.

Your code should appear as follows:

```
private var sharedObj:SharedObject;
```

18. Locate the `initConn()` function.
19. Within the `initConn()` function assign the `sharedObj` variable to the `getLocal()` method of the `SharedObject` class that takes a shared object named `myNotification` as an argument.
20. Create a conditional that sets the `txtArea.text` value equal to the `sharedObj.data.contacts` property if the `sharedObj.size` property is greater than 0. The `contacts` property within the `data` property stores the actual value of the shared object.

Your code should appear as follows.

```
private function initConn():void
{
    conn = new LocalConnection;
    conn.client = this;
    conn.connect("_myConnection2");
    sharedObj =
SharedObject.getLocal("myNotification");
    if(sharedObj.size > 0)
    {
        txtArea.text = sharedObj.data.contacts;
    }
}
```

21. Below the last function between the `<mx:Script>` tags, create a private function named `saveData` that accepts one parameter named `event` with a datatype set to the `MouseEvent` class. The function returns `void`.
22. Within the `saveData()` function assign the `txtArea.text` to `sharedObj.data.contacts`. Use the `flush()` method of the `sharedObj` instance to immediately write the locally persistent shared object to a local file. If you don't use this method, Flash writes the shared object to a file when the shared object session ends--that is, when the SWF file is closed, when

the shared object is garbage-collected because it no longer has any references to it, or when you call `SharedObject.close()`.
Your code should appear as follows.

```
private function saveData(event:MouseEvent):void
{
    sharedObj.data.contacts = txtArea.text;
    sharedObj.flush();
}
```

23. Below the `saveData()` function create a private function named `deleteData` that takes one parameter named `event` with a data type set to the `MouseEvent` class. This function returns `void`.
24. Within the `deleteData()` function use the `clear` method to remove the `sharedObj` data and delete the persistent copy of the object.

Your code should appear as follows.

```
private function deleteData(event:MouseEvent):void
{
    sharedObj.clear();
}
```

25. Below the `<mx:Label>` tag locate create an `<mx:HBox>` component with the following properties.
 - `x = 81`
 - `y = 318`
26. Between the `<mx:HBox>` tags create a `<mx:Button>` component with a `label` property of `Save Shared Object` and a `click` property that invokes the `saveData()` function.
27. Add another `<mx:Button>` component with a `label` property of `Delete Shared Object` and a `click` property that invokes the `deleteData()` function.

Your code should appear as follows.

```
<mx:HBox x="81" y="318">

    <mx:Button label="Save Shared Object"
    click="saveData(event)" />

    <mx:Button label="Delete Shared Object"
    click="deleteData(event)" />
```

```
</mx:HBox>
```

Your complete code should appear as follows.

```
<?xml version="1.0" encoding="utf-8"?>
<mx:WindowedApplication xmlns:mx="http://www.adobe.com/2006/mxml"
    layout="absolute"
    creationComplete="initConn()">

    <mx:Script>
        <![CDATA[

            import flash.net.SharedObject;

            import flash.net.LocalConnection;

            private var conn:LocalConnection;

            private var sharedObj:SharedObject;

            private function initConn():void
            {
                conn = new LocalConnection;
                conn.client = this;
                conn.connect("_myConnection2");

                sharedObj =
                SharedObject.getLocal("myNotification");

                if(sharedObj.size > 0)
                {
                    txtArea.text = sharedObj.data.contacts;
                }

            }

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

            private function saveData(event:MouseEvent):void
            {
```

```

        sharedObj.data.contacts = txtArea.text;
        sharedObj.flush();
    }

    private function deleteData(event:MouseEvent):void
    {
        sharedObj.clear();
    }

]]>
</mx:Script>

<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"/>

<mx:HBox x="81" y="318">

    <mx:Button label="Save Shared Object"
        click="saveData(event)" />

    <mx:Button label="Delete Shared Object"
        click="deleteData(event)" />

</mx:HBox>

</mx:WindowedApplication>

```

28. 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 with two buttons at the bottom. We had created the “sender” piece of the local connection in exercise 23.

29. 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. Now click on the `Save Shared Object` button to save the notification. If you close the window and run the application again, you will see the notification saved and available in the new window. Use the `Delete Shared Object` button to delete any notifications present in the window. If you close the window and run the application again, you will see the notification has been deleted.

The link below has a lot of useful information about using shared objects.

http://livedocs.adobe.com/flex/3/html/help.html?content=lsos_3.html