Creating SWC files

Import the Project

1. If not already created, create a directory named `adobeFlexTraining` on your **C** drive.
2. In Flex Builder, select **File > Import > Flex Project**.
3. In the dialog window, select **Archive File** and browse to where `Ex21a_Starter.zip` is located in your local file system.
4. Uncheck **Use default location**.
5. Enter `C:\adobeFlexTraining\EmployeePortal`.
6. Click **Finish**.

Run the imported project

7. Using the **Flex Navigator** view, open the **EmployeePortal > src** folder.
8. Select and run the `EmployeePortal.mxml` file.

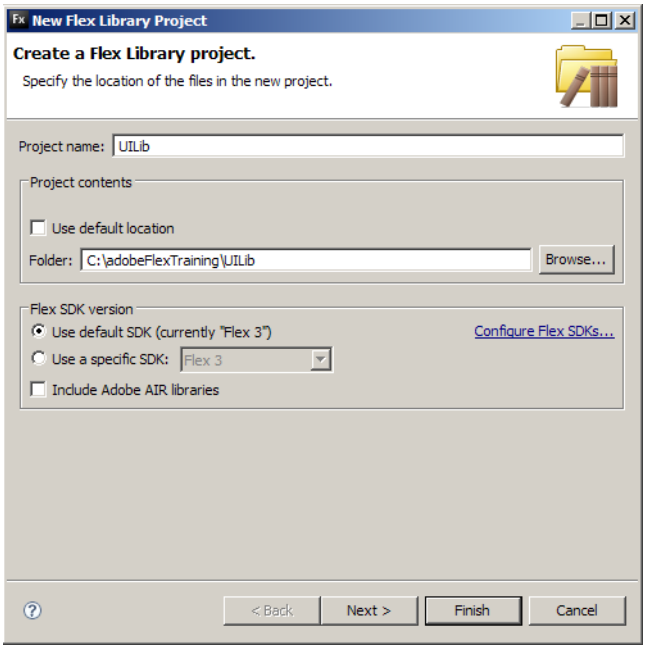
   Notice the application and its layout when it gets launched. This is a simple one page application with a text field for Username and Password. We are only interested in the look and feel of this application for our learning purposes, not its functionality.

Creating a Flex Library to reuse across applications

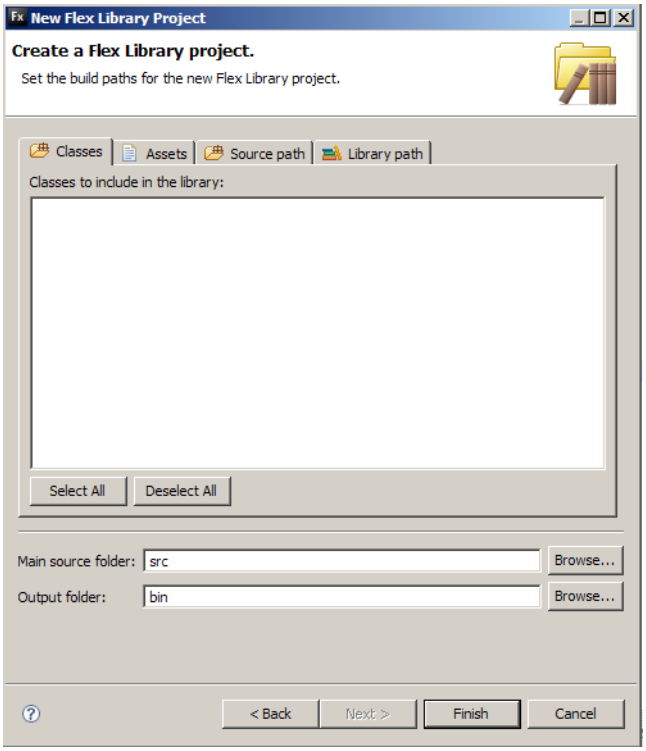
9. Using the **Flex Navigator** view, open the **EmployeePortal > src** folder.

   Notice the folders **assets, components** and **skins** located within the **src** folder. We will move these folders to a separate Flex Library project to reuse across application.

10. In Flex Builder, select **File > New > Flex Library Project.**
11. In the dialog window, enter `UILib` as the **Project** name.
12. Uncheck **Use default location**.
13. Enter `C:\adobeFlexTraining\UILib`.



14. Click the **Next** button.
15. Enter `src` for the **Main source folder**.

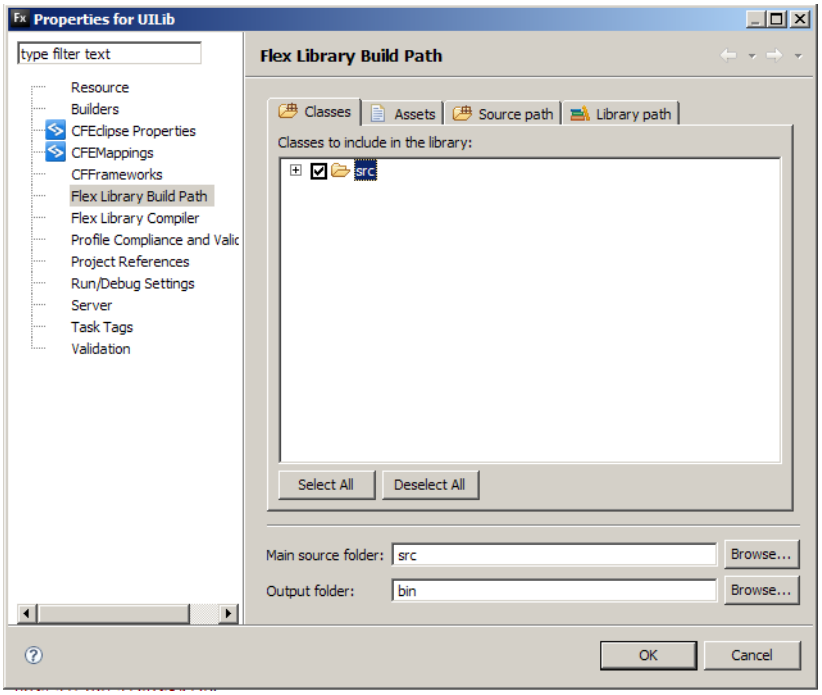


16. Click **Finish**.

    You will notice that the **UILib** project shows an error icon.

17. Select the folders **assets, components and skins** under **EmployeePortal > src** folder and move them to the **UILib > src** folder.
    You will notice that the **EmployeePortal** project is now showing an error icon.

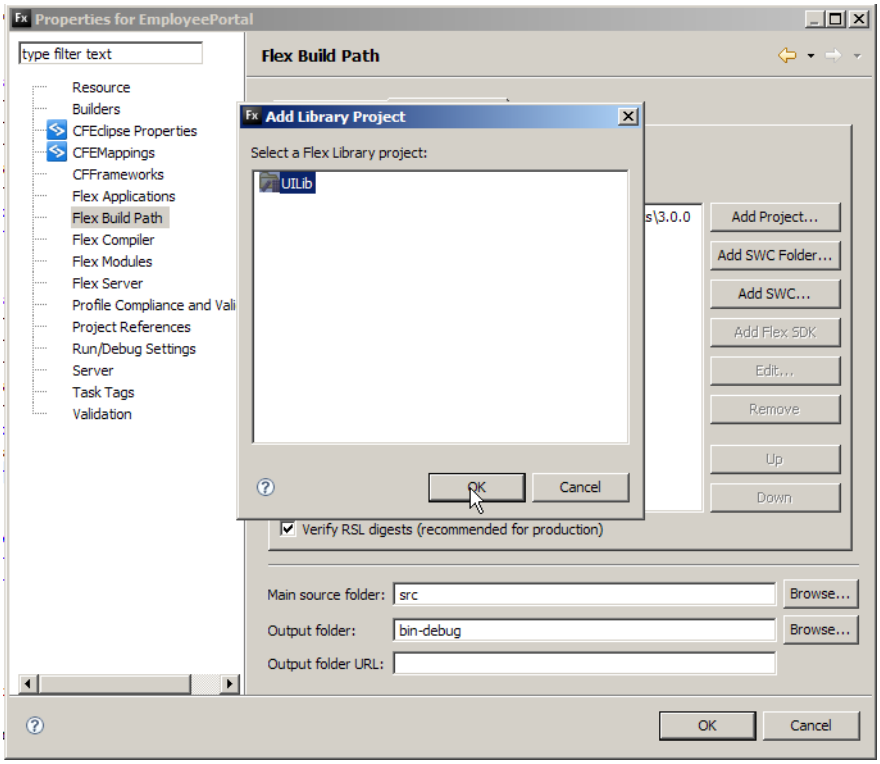
18. Right-click on the **UILib** project and select **Properties**.
19. Select the **Flex Library Build Path** from the left hand menu.
20. In the **Classes** tab, check the checkbox for the **src** folder. This tells the project which folders to include in the compilation of the library.



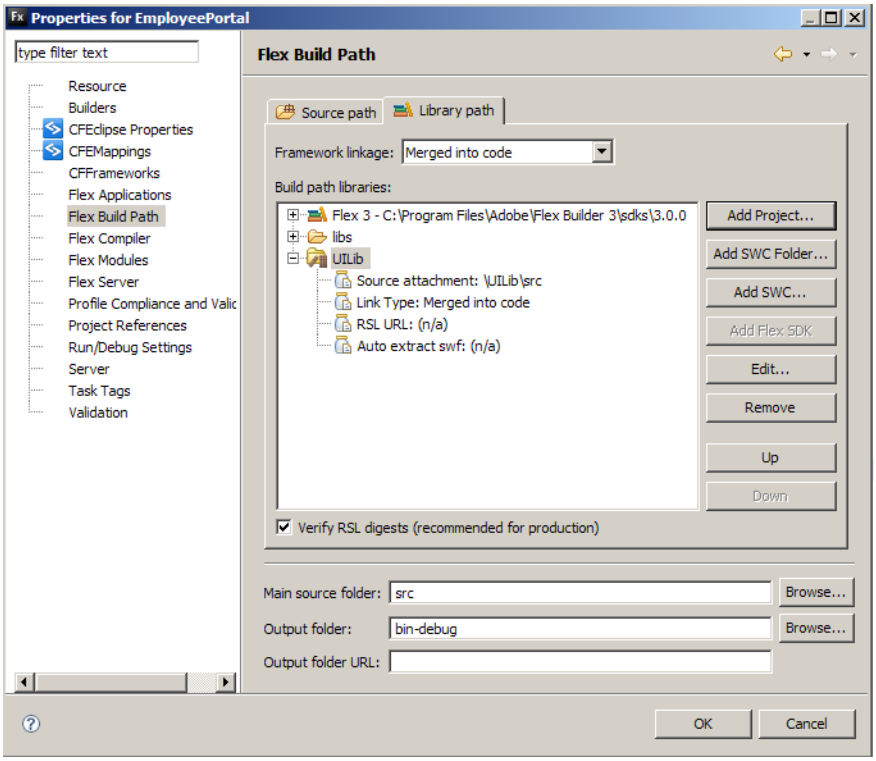
21. Click **OK**.
    You will notice the **UILib** project is not longer showing an error icon.

Linking the Flex library with the project

22. Right-click on the project name `EmployeePortal` and select **Properties**.
23. Select the **Flex Build Path** from the left hand menu.
24. Select the **Library path** tab.
25. Click the **Add Project** button
26. In the **Add Library Project** dialog select **UILib** as the **Flex Library Project**.



27. Click **OK** to add the library to the application. Notice the `UILib` now exists as one of the Build Path Libraries associated with the project.



28. Click **OK**.

You will notice the **EmployeePortal** is no longer showing an error icon.

29. Using the **Flex Navigator** panel, open the **EmployeePortal > src** folder.
30. Run the `EmployeePortal.mxml` file again.

    Notice there is no change in the application and its layout even though the resources for the project have been moved to a separate Flex library.

    Creating a Flex Library Project like explained above will allow applications to reuse code without bundling it with each project. You can create flex libraries for any code or assets that will be used in multiple applications. We will now use a skin for the Flex library just created in another application.

Apply the skin to the second project

31. In Flex Builder, select **File > Import > Flex Project**.
32. In the dialog window, select **Archive File** and browse to where `Ex21b_Starter.zip` is located in your local file system.
33. Uncheck **Use default location**.
34. Enter `C:\adobeFlexTraining\ReservationSystem1`.
35. Click **Finish**.
36. Using the **Flex Navigator** view, open the **ReservationSystem1 > src** folder.
37. Select and run the `ReservationSystem.mxml` file.

    Notice the application and its layout when it gets launched.
    This is a simple login application with a text field for Username and Password and a submit button. We are only interested in the look and feel of this application for our leaning purposes, and not its functionality.

38. Right-click on the project name **ReservationSystem1** and select **Properties.**
39. Select **Flex Build Path** from the left hand menu.
40. Select the **Library Path** tab.
41. Click the **Add Project** button.
42. In the **Add Library Project** dialog select **UILib** as the **Flex Library Project**.
43. Click **OK** to add the library to the application.
    Notice the **UILib** as one of the **Build Path Libraries** associated with the project.
44. Using the **Flex Navigator** view, open the `ReservationSystem.mxml` located in the **ReservationSystem1 > src** folder.

45. Using the **Outline** view locate the `<mx:Canvas>` tag.
46. Add the `borderSkin` property to the `<mx:Canvas>` tag and set it equal to `skins.PortalContainerSkin`.

    Your code should appear as follows.

```
<mx:Canvas width="864" height="604"
   horizontalCenter="0"
   verticalCenter="0"
   y="20"
   borderSkin="skins.PortalContainerSkin">
```

47. Save and run the `ReservationSystem.mxml` file.

    Notice the layout of the application changes after the skin is applied.