SL4 Out Of Browser Sample Notes

In order to login to Facebook and use the Facebook graph data you need to acquire an Access Token. The sample app does this with minimal effort on your part. Customizing the first 2 lines of the MainPage class (lines 20 and 21) in the MainPage.xaml.cs is all that you need to do to see the application in action (you do need to create a Facebook application and have your Application ID handy). The Application ID goes into the apiKey variable (line 18). The Permissions that you want to get from the user goes into the “permission” string variable and the values are separated by commas (line 20).

This example also works in conjunction with a web site. There are a couple pages that have to be served up from a webserver. This gets around a few issues. These pages aren’t ASPX pages, but they have to be located on the same web server that the Out of Browser application originated from (or else they won’t work properly). You can customize the location of these two pages in lines 22 and 24.

The failedLogin method (lines 37-40) is where you should handle a login failure (probably caused by the user canceling).

The loginSucceeded method (lines 43-55) is the routine that is called when the login process succeeds. The original demo asks for permissions to basic user data and is then binding this data to a listbox control.

## Flow

What takes place with this app is that after the browser control has been created and is ready to go the app constructs a URI using a “popup” display. The URI also tells Facebook which permissions are needed and which pages to land on when the process is complete or has failed (user aborts). Finally the app points the browser control to the Facebook login page. At this point the user will interact with the browser control (you have no control at this point). Once the user completes or cancels the login process one of the 2 landing pages will be called and these pages will pass the QueryString (containing the access token) or a message telling the application that the user canceled login.

When the app gets a QueryString it looks for the Access Token and when there is one it updates a few variables, sets up the fbApp variable (A FacebookApp instance from the SDK), and then calls the loginSucceeded method. If it gets a failure message (or a “user canceled” message), it will call the loginFailed method. Additionally if it gets to the success page but doesn’t get an Access Token then it will call the loginFailed method.

For this sample when the loginSucceeded method is called the app makes a few things visible and hides the browser control. Finally it will call the FacebookApp’s ApiAsync method to get the user’s “me” data (and when this completes we set this JsonObject/IDictionary<string, object> object as the item source for the listbox using the UI Thread –aka Dispatcher).