WP7 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 20). The Permissions that you want to get from the user goes into the “permission” string variable and the values are separated by commas (line 21).

The failedLogin method (lines 39-42) is where you should handle a login failure (probably caused by the user canceling).

The loginSucceeded method (lines 45-54) is the routine that is called when the login process succeeds. In the original demo the app is asking for permissions to basic user data and are binding this to a listbox control.

At the time of this writing there is a Facebook bug where the allow buttons are off screen (no matter the height of the browser window), so this sample shows a workaround in the code to fix this issue. This makes it so that the user doesn’t have to know that they need to scroll. You may need to tweak this value which is in the allowButtonPosition variable on line 24. If you have 3 or more permissions you are requesting you will probably need to tweak this setting. When the bug is finally fixed lines 57-75 will no longer be necessary.

## 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 “touch” display which instructs Facebook to log the user in and then to give the app back an access token when this all completes successfully. 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 this URI (a Facebook login page). At this point the user will interact with the browser control (you have no control at this point).

The application detects the various pages that the user hits via the browser controls’ Navigated event and handles things accordingly. When it sees the success page it looks for the Access Token and when it finds one it updates a few variables, sets up the fbApp variable (A FacebookApp instance from the SDK), and then calls the loginSucceeded method. If the user lands on the failure page then the app will call the loginFailed method. Additionally if the user gets to the success page but the app doesn’t get an Access Token then the loginFailed method is called.

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 it sets this IDictionary<string, object> object as the item source for the listbox using the UI Thread –aka Dispatcher).