Windows Phone 8 - Adding native UI elements

Overview

While writing hybrid application can be done by using solely web technologies, IBM MobileFirst Platform Foundation also allows to mix & match native code with web code as needed.

For example, use native UI controls, use native elements, provide an animated native introduction screen, etc. In order to do so, taking control of part of the application startup flow is needed. This tutorial assumes working knowledge of Windows Phone development (C#).

Send Action From JavaScript to Native

In MobileFirst applications, commands are sent with parameters from the web view (via JavaScript) to a native class (written in C#).
This feature can be used to trigger native code to be run in the background, to update the native UI, to use native-only features, etc.

Step 1
In JavaScript, the following API is used:
`WL.App.sendActionToNative("doSomething", {customData: 12345});`

`doSomething` is an arbitrary action name to be used in the native side (see the next slide), and the second parameter is a JSON object that contains any data.

Step 2
The native class to receive the action must implement the `WLActionReceiver` protocol:
`public class action : WLActionReceiver`

The `WLActionReceiver` protocol requires an `onActionReceived` method in which the action name can be checked for and perform any native code that the action needs:

```
public void onActionReceived(string action, JObject data)
 {
   if (action == "displayAddress")
   {
     Deployment.Current.Dispatcher.BeginInvoke(() =>
     {
       // perform required actions
     });
   }
 }
```

Step 3
For the action receiver to receive actions from the MobileFirst Web View, it must be registered.
The registration can be done during the startup flow of the application to catch any actions early enough:

```
InitializeComponent();
WL.createInstance(); //Create the instance of the ActionSender API'
s
myReceiver = new Action();
Loaded += PhoneAppPage_Loaded;
WL.getInstance().addActionReceiver(myReceiver);
```

You can later remove this ActionReceiver registration by using the following API:

```
WL.getInstance().removeActionReceiver(myReceiver);
```

Send Action From Native to JavaScript

In MobileFirst applications, commands can be sent with parameters from native C# code to web view JavaScript code.
This feature can be used to receive responses from a native method, notify the web view when background code finished running, have a native UI control the content of the web view, etc.

Step 1
In C#, the following API is used:

```
WL.getInstance().sendActionToJS(doSomething, data);
```

`doSomething` is an arbitrary action name to be used on the JavaScript side and the second parameter is a `JObject` that contains any data.

Step 2
A JavaScript function, which verifies the action name and implements any JavaScript code.

```
function actionReceiver(received)
  if (received.action == "doSomething" && received.data.someProperty == "12345")
{
    //perform required actions, e.g., update web user interface
  }
}
```

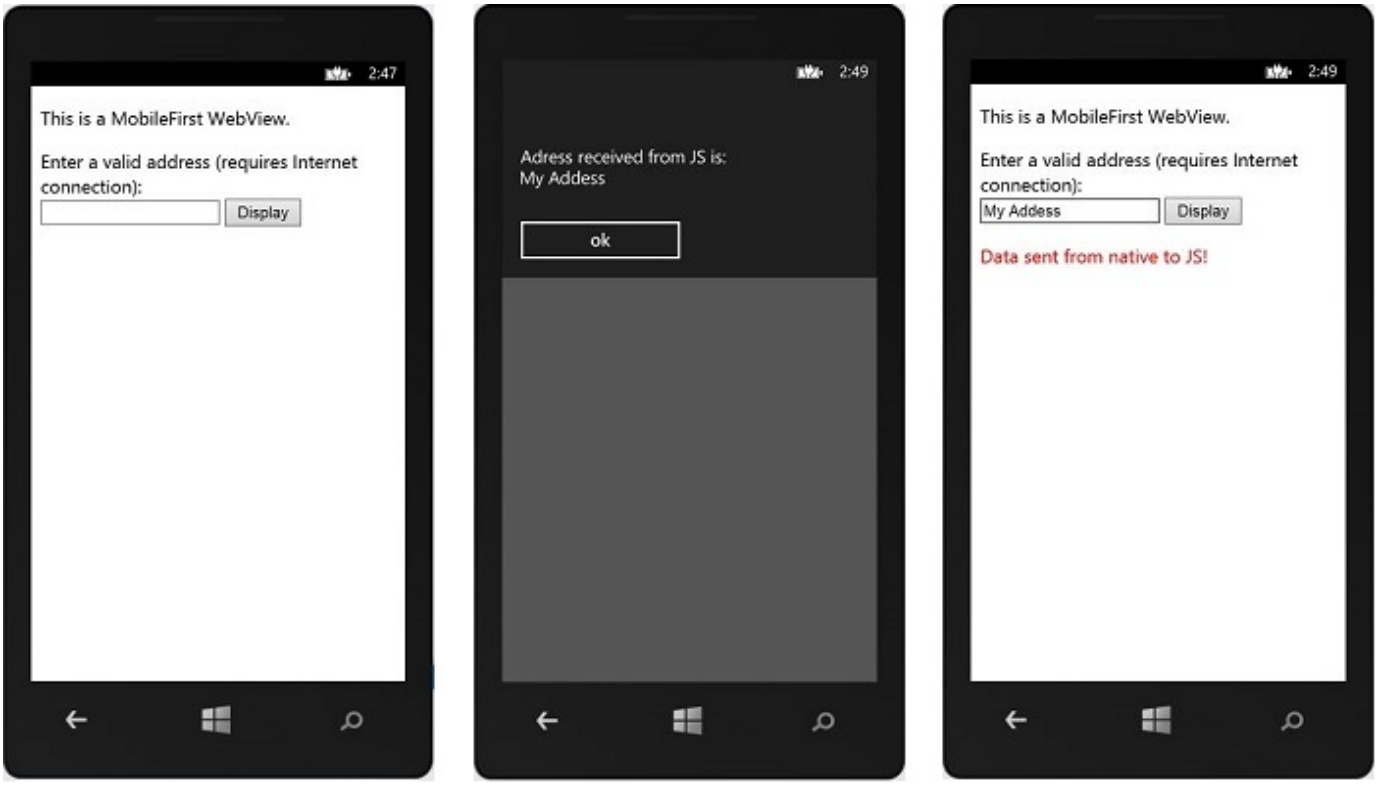
Step 3
For the action receiver to receive actions, it must first be registered. This should be done early enough in the JavaScript code so that the function will handle those actions as early as possible.

```
WL.App.addActionReceiver ("MyActionReceiverId", actionReceiver);
```

The first parameter is an arbitrary name. It can be used later to remove an action receiver.

```
WL.App.removeActionReceiver("MyActionReceiverId");
```

SendAction Sample



Download the NativeUIInHybrid project, which includes a hybrid application called SendAction.
In the first screen enter an address and press the "Display" button. This action will send this address to the native C# code and it will display a MessageBox that will display the address you entered. When pressing the OK button of the MessageBox the native SendAction method will be called and send a message back to the JS code that will be displayed using the actionReceiver method.

HTML
The HTML page shows the following elements:

- A simple input field to enter an address
- A button to trigger validation
- An empty line to show potential error messages (We use it to display the message we send from the native code to the JS code)

```
<p>This is a MobileFirst WebView.</p><br />
<p>Enter a valid address (requires Internet connection):<br/><br />
    <input type="text" name="address" id="address"/><br />
<input type="button" value="Display" id="displayBtn"/><br />
 </p><br />
 <p id="errorMsg" style="color:red;"></p>
```

JavaScript
When the button is clicked, the `sendActionToNative` method is called to send the address to the native code.

```
$('#displayBtn').on('click', function(){
 $('#errorMsg').empty();
 WL.App.sendActionToNative("displayAddress",
   { address: $('#address').val()}
 );
});
```

The code also registers an action receiver to display potential error messages from the native code.

```
WL.App.addActionReceiver ("MyActionReceiverId", function actionReceiver(received){
  if(received.action == 'displayError'){
   $('#errorMsg').html(received.data.errorReason);
  }
});
```

Shared Session

When you use both JavaScript and native code in the same application, you might need to make HTTP requests to MobileFirst Server (connect, procedure invocation, etc.)

HTTP requests are explained in other tutorials (both for hybrid and native applications).
IBM MobileFirst Platform Foundation 6.2 and later keeps your session (cookies and HTTP headers) automatically synchronized between the JavaScript client and the native client.

Sample application

Click to download (http://public.dhe.ibm.com/software/products/en/MobileFirstPlatform/docs/v630/NativeUIInHybridProject.zip) the Studio project.