Launchers and Choosers for Windows Phone

This article demonstrates how to use *Launchers* and *Choosers* in Windows Phone

Introduction

Windows phone uses launchers and choosers to provide access to common device functionality and services. When a particular function is required, a launcher/chooser is called to start the built in app and perform the required task; on completion control returns to the calling app.

The main difference between launchers and choosers is that a chooser usually returns data back to the app (for example taking a photo) while a launcher does not (for example, sending an email).

Note: This architecture means that if you want to use a picture in your app you fire off a chooser to take the photo rather than embedding camera functionality in your own app. This is a bit frustrating if you want to write your own camera, but it does provide a consistent experience for users and make device memory management a lot more controllable.

The code example app described here uses a launcher to start the Media Player and play a video, and uses the Chooser to launch the Camera and take a photo. Each of the *Launchers* and *Choosers* have their own set of properties. After setting any of those properties, we need to call `Show()` method. For *Choosers* we need to implement the event handler for when the user has taken a picture, in order to get the image back to handle.

Launcher and Chooser

Media Player

Camera

Implementation

First create a project with **Windows Phone Application** Template. Once the project is created, add the reference **Microsoft.Phone.Tasks** (needed for both choosers and launchers).

```
using Microsoft.Phone.Tasks;
```

Launch Media Player

To launch the Media Player using the Launcher API, first create the instance of the MediaPlayerLauncher then set the media file to be launched and then call the `Show()` function to launch the media player.

```
MediaPlayerLauncher mediaPlayerLauncher = new MediaPlayerLauncher();
mediaPlayerLauncher.Media = new Uri(@"http://video-js.zencoder.com/oceans-clip.mp4", UriKind.Absolute);
mediaPlayerLauncher.Show();
```

This code launches the default media player of the device and start playing the file.

Launch Camera chooser

For Chooser API we launch the build-in Camera and take photo.

(C) Copyright Nokia 2013. All rights reserved.

```
CameraCaptureTask cameraCaptureTask;
cameraCaptureTask = new CameraCaptureTask();
cameraCaptureTask.Completed += new EventHandler<PhotoResult>(cameraCaptureTask_Completed);
cameraCaptureTask.Show();
```

When user completes the task, an event is raised and the event handler `cameraCaptureTask_Completed()` receives a photo in the result, which is then displayed in the screen.

```
void cameraCaptureTask_Completed(object sender, PhotoResult e)
        {
            if (e.TaskResult == TaskResult.OK)
            {
                BitmapImage bmp = new BitmapImage();
                bmp.SetSource(e.ChosenPhoto);
                image1.Source = bmp;
            }
        }
```

Note: I have tested the application in emulator where I see a moving block when I launch the camera.

Source Code

The full source code of the example is available here: File:LauncherAndChooser.zip

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.