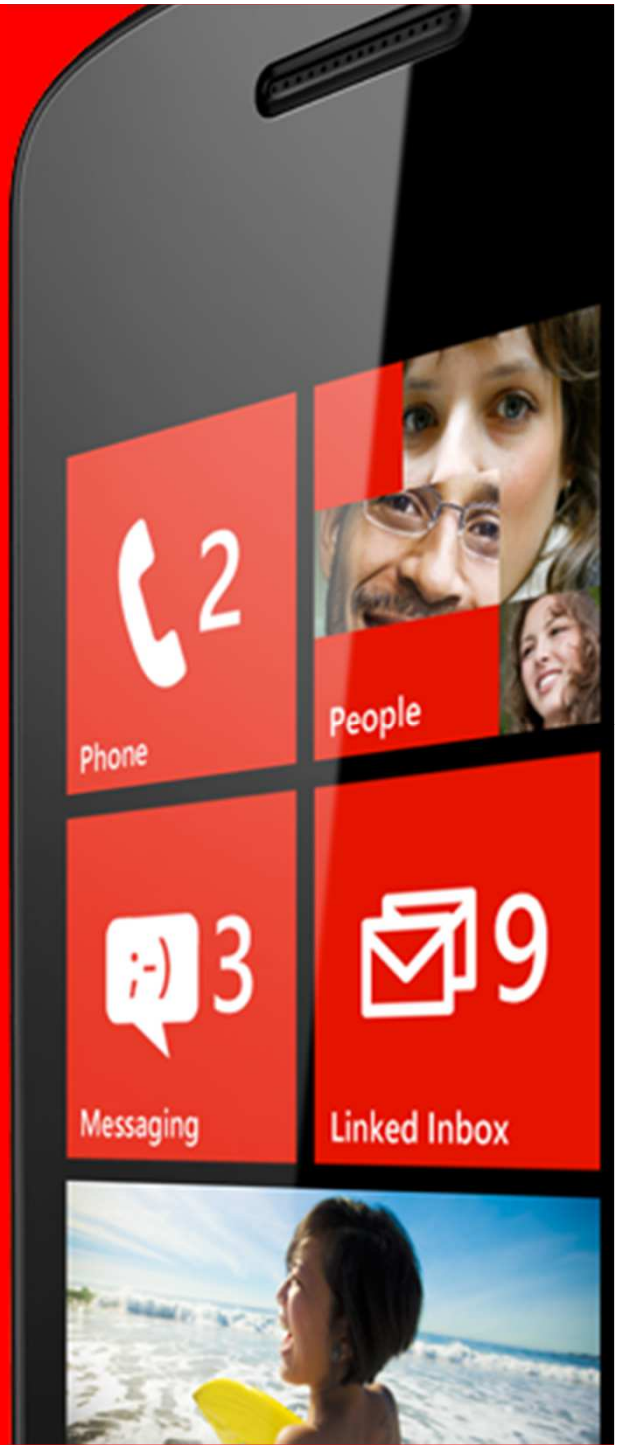
Windows® Phone

Launchers and Choosers

Session 9.3



Topics

- Launchers and choosers in context
- Tombstoning and Launchers and Choosers
- Using a Launcher
  - Starting an application and returning from it
- Using a Chooser
  - Starting an application and using the result that is returned

Launchers and Choosers

- Windows Phone provides a way for programs to interact with the phone itself:
  - Take photographs
  - Place phone calls
  - Interact with the address book
  - Select media
- Now we are going to find out how to do this

User Involvement

- Note that in all the applications the user is directly involved and has the final say on the action
- A program cannot just take a photograph, place a call or send and SMS
- The user must confirm these operations before they complete
- An application can only initiate the action

Launchers vs Choosers

- Applications call a **Launcher** makes use of a phone feature
  - Place a phone call, send an email or SMS
- A **Chooser** is used to select something
  - Take a picture and return it
  - Select an email address
- Both are used in the same way, but a Chooser will generate an event that delivers the result

Calling a Launcher or Chooser

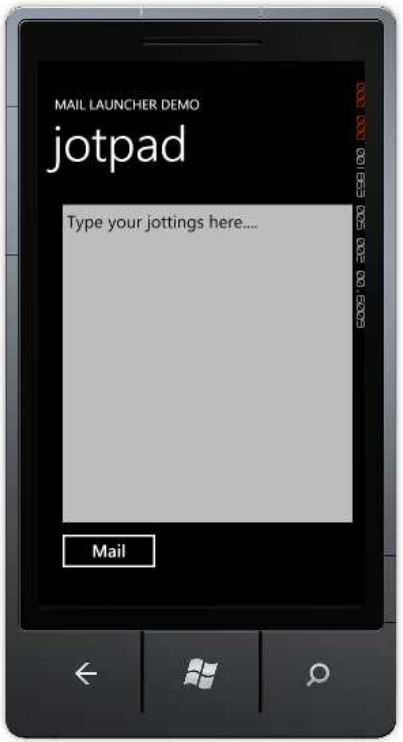
- When an application calls a Launcher or Chooser the new task gets control
- When the task is complete the application regains control
- If the user never returns from the Launcher/Chooser the application never gets control back
- This when the new task gets control an application may get tombstoned

Launcher Tasks

- Applications can create launchers to:
  - Open a web page
  - Search the Marketplace and find applications
  - Place a phone call
  - Send an email
  - Send an SMS message
  - Share a url
  - Share a status message
  - Start a search using Bing

Using a Launcher

- As an example, we could add an email feature to the JotPad application
- This would allow the user to send a jotting as an email
- When the Mail button is pressed the EmailComposeTask is started



The Mail button

```
private void mailButton_Click(object sender,
                              RoutedEventArgs e)
{
    sendMail("From JotPad", jotTextBox.Text);
}
```

- When the user clicks the mail button the event handler calls the **`sendMail`** method
- This is given the title and text of the email that is to be sent

The Mail button

```
private void sendMail(string subject, string body)
{
    EmailComposeTask email = new EmailComposeTask();

    email.Body = body;
    email.Subject = subject;
    email.Show();
}
```

- The **sendMail** method creates an **EmailComposeTask** instance and then calls **Show** on that instance
- When the email has been sent the jotPad program will resume

The Tasks namespace

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

- In order to use the Launcher and Chooser classes by name an application should add the above namespace
- Otherwise you will have to use the fully formed version of the class names

Demo

Demo 1: Email Jotpad

Choosers

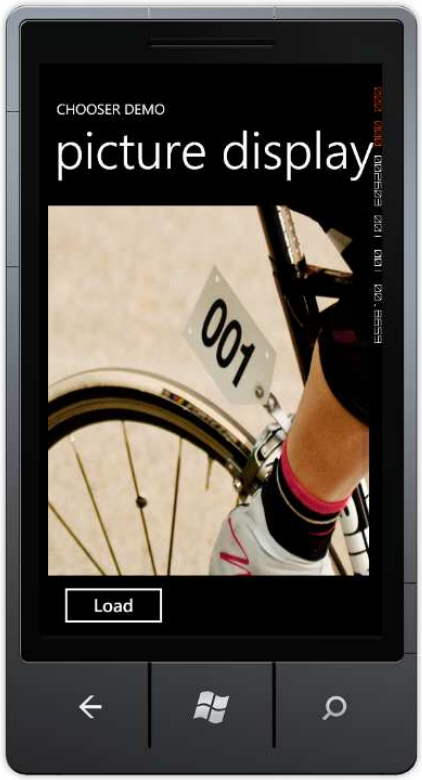
- Before an application calls a chooser it can bind to an event that the chooser task generates
- This is used to deliver a result object to the application when it regains control
- Choosers must be created in the constructor for a page and declared as members of the page class

Chooser Tasks

- Applications can create choosers to:
  - Get locations and routes from Bing Maps
  - Get addresses, phone numbers and email addresses
  - Select a picture from the media store
  - Capture a picture using the camera
  - Search the Marketplace and find applications
  - Invite players to a multi-game session

Picture display application

- The picture display application uses the PhotoChooserTask to allow the user to select a picture for display
- It then displays this on the phone screen



Creating the PhotoChooserTask

```
PhotoChooserTask photoChooser;

public MainPage()
{
  InitializeComponent();

  photoChooser = new PhotoChooserTask();

  photoChooser.Completed +=
   new EventHandler<PhotoResult>(photoChooser_Completed);
}
```

- The page constructor creates a **PhotoChooserTask** and binds a method to the Completed event

The Completed event handler

```
void photoChooser_Completed(object sender, PhotoResult e)
{
    if (e.TaskResult == TaskResult.OK)
    {
        selectedImage.Source =
            new BitmapImage(new Uri(e.OriginalFileName));
    }
}
```

- The event handler for the completed event creates a new bitmap image from the filename in the result and displays this

The TaskResult field

```
void photoChooser_Completed(object sender, PhotoResult e)
{
    if (e.TaskResult == TaskResult.OK)
    {
        selectedImage.Source =
            new BitmapImage(new Uri(e.OriginalFileName));
    }
}
```

- The **`TaskResult`** field in the result is set to **`TaskResult.OK`** if the user completed the choose action

The OriginalFileName field

```
void photoChooser_Completed(object sender, PhotoResult e)
{
    if (e.TaskResult == TaskResult.OK)
    {
        selectedImage.Source =
            new BitmapImage(new Uri(e.OriginalFileName));
    }
}
```

- The result also contains the filename of the photo that was selected
- We can use this to create a URI to the image

Create a new image

```
void photoChooser_Completed(object sender, PhotoResult e)
{
    if (e.TaskResult == TaskResult.OK)
    {
        selectedImage.Source =
            new BitmapImage(new Uri(e.OriginalFileName));
    }
}
```

- The program can create a new image from the URI and then set the source of the selected image to this

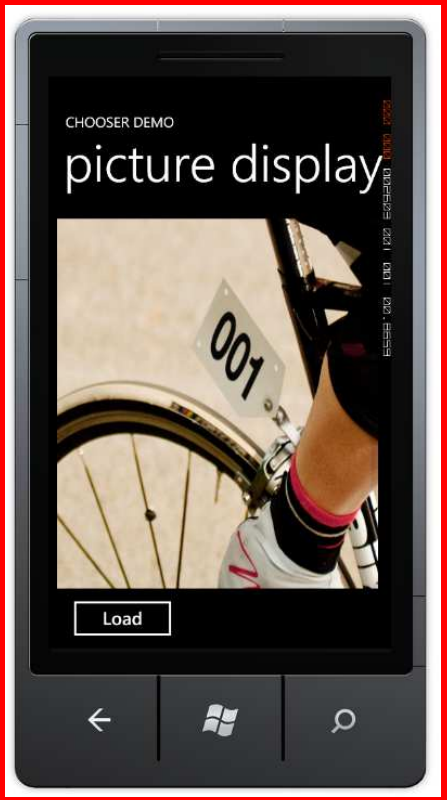
Load button event handler

```
private void loadButton_Click(object sender,
                              RoutedEventArgs e)
{
    photoChooser.Show();
}
```

- When the Load button is pressed the event handler just calls the Show method on the chooser that was created in the form constructor

Demo

Demo 2: Picture display



Summary

- Launchers and Choosers provide a way that applications can use phone features
- Launchers just start a phone feature running whereas a Chooser can return a result
- When a Launcher or Chooser is invoked the running application is tombstoned
- A Chooser will fire an event method in the application when/if it returns