



TRUNG TÂM TIN HỌC

Trường Đại học Khoa Hoc Tự Nhiên - Đại học quốc gia TP.HCM

Công GMC.blogspot.com  
MediaFire Blogger

## LẬP TRÌNH WINDOWS PHONE

# Module 2

☞ *Click vào phụ lục để chuyển tới bài cần đọc*

### Phụ lục

|  |    |
|--|----|
| Bài 1. Giới thiệu Windows Phone.....                     | 2  |
| Bài 2 Tìm hiểu XAML và các control cơ bản trong WP ..... | 32 |
| Bài 3 Các control nâng cao trong Windows Phone.....      | 50 |
| Bài 4 Các loại ứng dụng và chuyển trang trong WP .....   | 70 |
| Bài 5 Data binding cơ bản.....                           | 86 |

Công GMC.blogspot.com  
 MediaFire Blogger



Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh  
TRUNG TÂM TIN HỌC

Go Screen Capture



## Lập trình Windows Phone

### Bài 1. Giới thiệu Windows Phone

*Ngành Mạng và Thiết bị di động*



2014



PTOS



## Nội dung



1. Lịch sử phát triển Windows Phone
2. Các tính năng mới trong Windows Phone 8
3. Giới thiệu Windows Phone 8 SDK
4. Bắt đầu với lập trình Windows Phone 8
5. Vòng đời ứng dụng Windows Phone (application life cycle)



## 1. Lịch sử phát triển Windows Phone



- Windows Phone 7.0 (10.2010)
  - ✓ Phiên bản đầu tiên của hệ điều hành Windows Phone



## 1. Lịch sử phát triển Windows Phone



- Windows phone 7.1 Mango (05/2011)
  - ✓ People Hub: Facebook, Hotmail, Gmail, LinkedIn và Twitter
  - ✓ Tính năng Groups: cho phép người dùng giao tiếp trên nhiều ứng dụng cùng lúc (SMS, facebook, Hotmail, Gmail, Twitter, LinkedIn).
  - ✓ Internet Explorer 9



## 1. Lịch sử phát triển Windows Phone



- Windows phone 7.5 Mango (05/2011)
  - ✓ Còn gọi là “No Do”
  - ✓ Hỗ trợ CDMA, Copy và Paste
  - ✓ Hỗ trợ khởi động nhanh ứng dụng



## 1. Lịch sử phát triển Windows Phone



- Windows phone 7.5 Refresh (03/2012)
  - ✓ Hỗ trợ mạng 4G (LTE)
  - ✓ Phát hành 2 phiên bản khác nhau cho Nokia Lumia 900 và HTC Titan II của AT&T
  - ✓ Sửa lỗi mất bàn phím.
  - ✓ Sửa lỗi truy cập định vị
  - ✓ ..và một số lỗi khác



## 1. Lịch sử phát triển Windows Phone



- Windows phone 7.5 Tango(06/2012)
  - ✓ Nâng cấp tin nhắn đa phương tiện
  - ✓ Có thể gửi nhạc chuông qua MMS
  - ✓ Có thể xuất và quản lý danh bạ từ SIM card
  - ✓ Sửa một số lỗi về thanh toán trên ứng dụng ở một số nước



## 1. Lịch sử phát triển Windows Phone



- Windows phone 7.8 (01/2013)
  - ✓ Phiên bản cuối cùng của Windows Phone 7.x.
  - ✓ Mang một số tính năng mới trên Windows Phone 8 như:
    - Thay đổi kích thước của tiles ở màn hình chính
    - Màn hình khởi động mới
    - Các biểu tượng mới cho các ứng dụng cơ bản như Games, Office, Windows Phone Store

Các thiết bị chạy Windows Phone 7.x sẽ không thể nâng cấp lên Windows Phone 8.



## 1. Lịch sử phát triển Windows Phone



- Windows phone 8 (29/10/2012)
  - ✓ Chuyển sang sử dụng các component từ Windows 8 (Windows Shared Core) như: kernel, file system, drivers, network, media and graphics
  - ✓ Hỗ trợ CPU đa nhân
  - ✓ Hỗ trợ nhiều độ phân giải (1280x720, 1280x768, 1080p)
  - ✓ Hỗ trợ thẻ nhớ
  - ✓ Internet Explorer 10
  - ✓ Hỗ trợ màn hình 6"



## 2. Các tính năng mới trong Windows Phone 8



- ❖ Điện thoại
- ❖ Nhạc + videos
- ❖ Danh bạ và mạng xã hội
- ❖ Office trên di động
- ❖ Email + Tin nhắn
- ❖ Networking
- ❖ Trình duyệt và bản đồ
- ❖ Phần mềm doanh nghiệp
- ❖ Hình ảnh + camera
- ❖ Windows Phone online

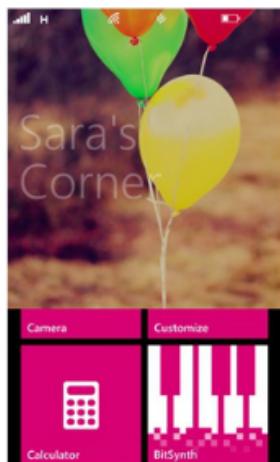
<http://www.windowsphone.com/en-us/how-to/wp8/start/whats-new-in-windows-phone>



## 2. Các tính năng mới trong Windows Phone 8



### ❖ Điện thoại



- ✓ Kid's Corner
- ✓ Data Sense
- ✓ Quad core support
- ✓ Eyes on the road
- ✓ Wallet
- ✓ Screen rotation lock
- ✓ Microsoft account
- ✓ Back up your stuff
- ✓ Start screen
- ✓ Tap + send



## 2. Các tính năng mới trong Windows Phone 8



### ❖ Danh bạ và mạng xã hội

- ✓ **Rooms:** nơi chia sẻ dữ liệu riêng tư của các thành viên trong nhóm như: lịch làm việc, hình ảnh, chat, ghi chú



- ✓ **Groups sync:** Đồng bộ Groups lên tài khoản Microsoft

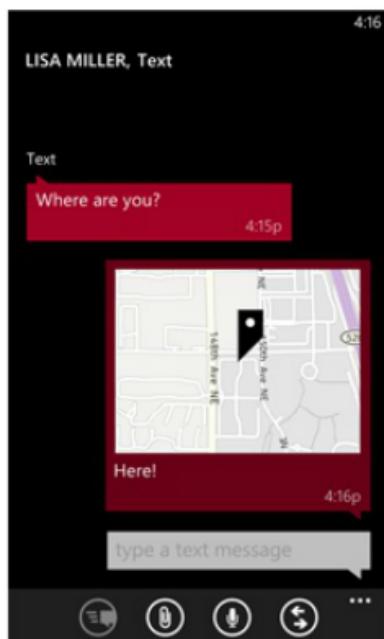
- ✓ **Contact sharing:** Có thể chia sẻ Danh bạ dễ dàng qua email, SMS, hay NFC



## 2. Các tính năng mới trong Windows Phone 8



### ❖ Email + Tin nhắn



- ✓ **Gởi mail bằng giọng nói**
- ✓ **Đặt âm báo cho từng người**
- ✓ **Hỗ trợ đính kèm vị trí hay danh bạ**
- ✓ **Cài tiến bàn phím.**
- ✓ **Cài tiến trong việc soạn thảo**
- ✓ **Skype app**
- ✓ **Xóa tin nhắn và chat cũ**

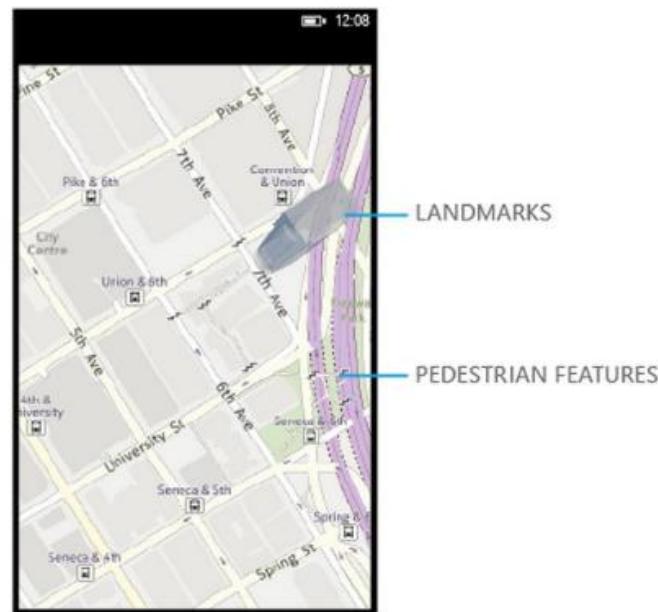


## 2. Các tính năng mới trong Windows Phone 8



### ❖ Trình duyệt và Bản đồ

- ✓ Cá nhân hóa việc tìm kiếm trên bản đồ
- ✓ Cài tiến trình duyệt
- ✓ Xóa một hoặc nhiều dòng dữ liệu trong Lịch sử trình duyệt



## 2. Các tính năng mới trong Windows Phone 8



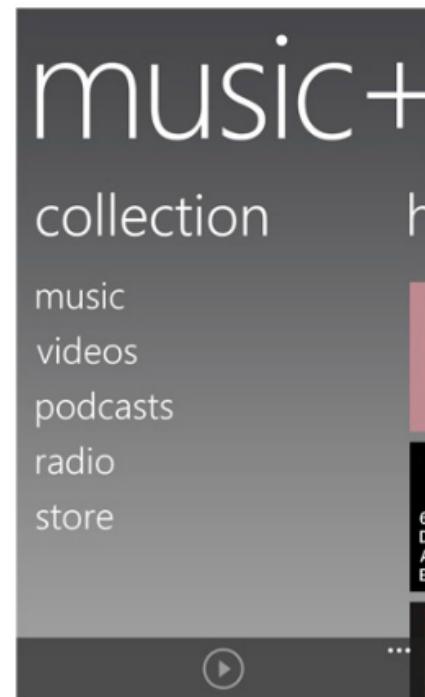
### ❖ Hình ảnh và camera

- ✓ Hiệu ứng
- ✓ Chỉnh sửa
- ✓ Tự động back up lên SkyDrive



## 2. Các tính năng mới trong Windows Phone 8

- ❖ Nhạc và videos
  - ✓ Xbox Music
  - ✓ FM radio
  - ✓ Tương tác nhiều hơn với playlist

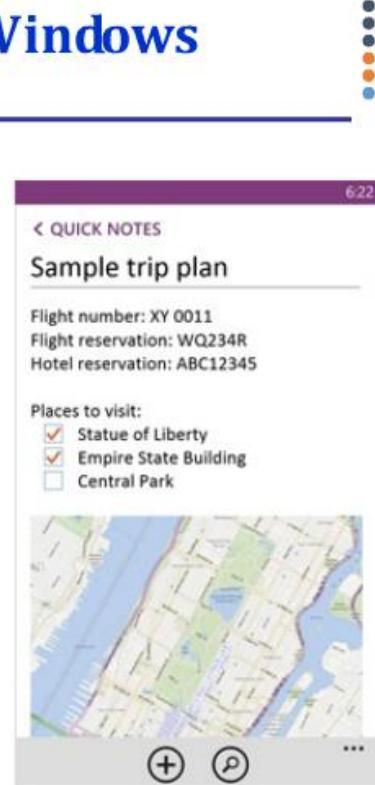


Bài 1: Giới thiệu Windows Phone

16

## 2. Các tính năng mới trong Windows Phone 8

- ❖ Office trên di động
  - ✓ Hỗ trợ OneNote
  - ✓ New Office Hub
  - ✓ Cài tiến các ứng dụng office



Bài 1: Giới thiệu Windows Phone

17

## 2. Các tính năng mới trong Windows Phone 8



### ❖ Ứng dụng và Games



- ✓ Windows Phone Store
- ✓ More Live Apps
- ✓ Even smarter apps

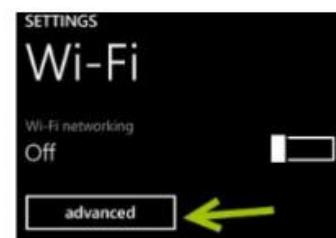


## 2. Các tính năng mới trong Windows Phone 8



### ❖ Networking

- ✓ Tùy chọn trong Wifi



- ✓ Chia sẻ qua Bluetooth



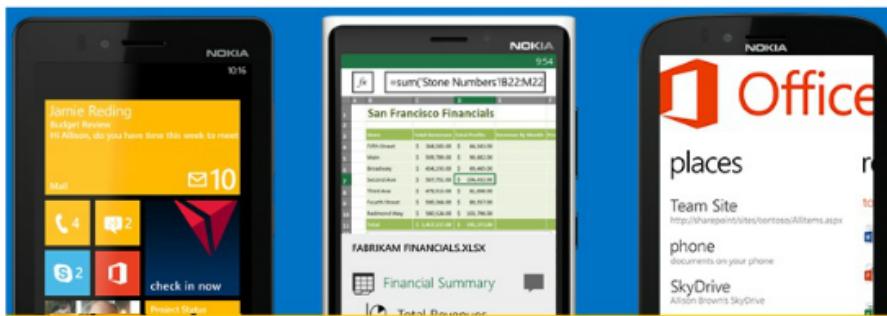
## 2. Các tính năng mới trong Windows Phone 8



### ❖ Phần mềm cho doanh nghiệp

- ✓ Các tính năng mới cho ứng dụng doanh nghiệp
- ✓ Đồng nhất môi trường làm việc
- ✓ Truy cập tài liệu mọi lúc, mọi nơi

Thu hẹp khoảng cách giữa nhu cầu của doanh nghiệp và nhu cầu của nhân viên về một ứng dụng trên di động



Bài 1: Giới thiệu Windows Phone

20

## 2. Các tính năng mới trong Windows Phone 8



### ❖ Windows Phone online

- ✓ Sử dụng My Family để quản lý các ứng dụng và game cho trẻ



Bài 1: Giới thiệu Windows Phone

21

### 3. Giới thiệu Windows Phone 8 SDK



- ❖ Yêu cầu phần cứng trên máy tính
- ❖ Yêu cầu phần cứng trên điện thoại
- ❖ Yêu cầu hệ điều hành
- ❖ Windows Phone 8 SDK



### 3. Giới thiệu Windows Phone 8 SDK



- ❖ Yêu cầu phần cứng trên máy tính
  - ✓ Hỗ trợ Hyper V: để chạy thiết bị giả lập
  - ✓ Slat (Second Level Address Translation): để chuyển đổi IP của máy ảo thành địa chỉ vật lý



### 3. Giới thiệu Windows Phone 8 SDK



- ❖ Yêu cầu phần cứng trên điện thoại
  - ✓ Thiết bị được cài sẵn Windows Phone 8
  - ✓ Không có yêu cầu bắt buộc về cấu hình



Bài 1: Giới thiệu Windows Phone

24

### 3. Giới thiệu Windows Phone 8 SDK



- ❖ Yêu cầu về hệ điều hành
  - ✓ Windows 8 Pro 64bit hoặc cao hơn



Bài 1: Giới thiệu Windows Phone

25

### 3. Giới thiệu Windows Phone 8 SDK



#### ❖ Windows Phone 8 SDK

Link download WP 8 SDK :

<http://www.microsoft.com/en-us/download/details.aspx?id=35471>

File download : Wpexpress\_full.exe

Thư mục cài đặt: \Program Files (x86)\Windows Phone Kits\8.0 directory.



### 4. Bắt đầu với lập trình Windows Phone 8

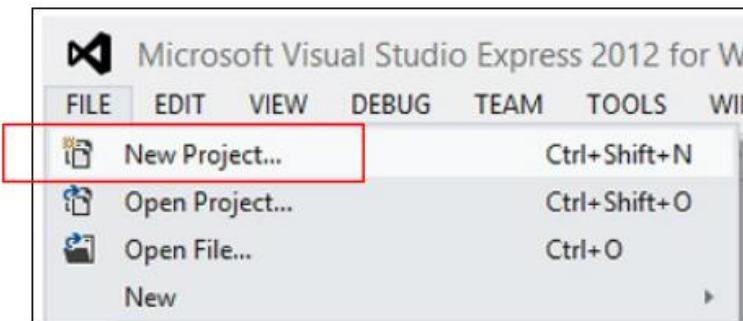


- **Tạo mới project Windows Phone 8**
- **Cấu trúc thư mục của project**
- **Khởi chạy ứng dụng**
- **Sử dụng emulator**
- **Đăng ký thiết bị thật để kiểm thử ứng dụng**



## 4. Bắt đầu với lập trình Windows Phone 8

### ➤ Tạo mới project Windows Phone 8

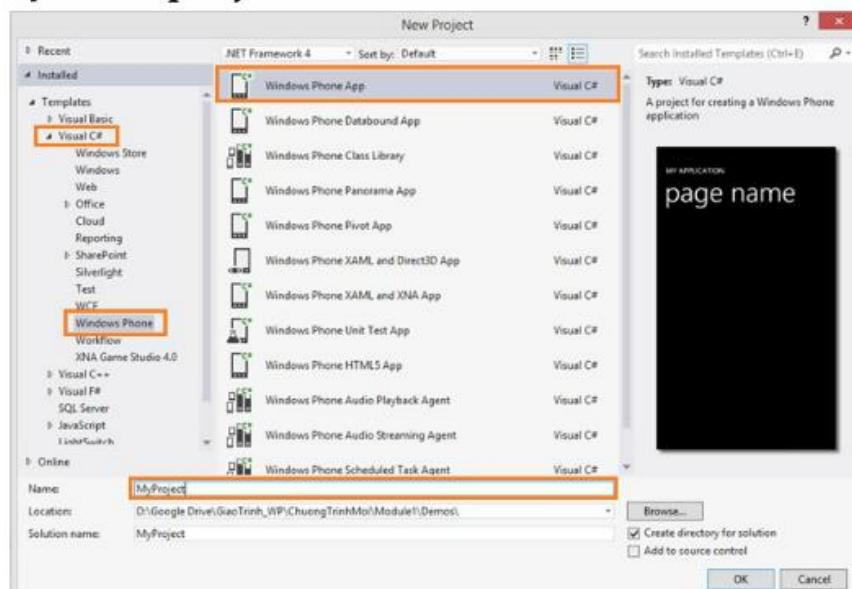


Mở Visual Studio 2012, chọn menu  
**File → New Project** để tạo project mới



## 4. Bắt đầu với lập trình Windows Phone 8

### ➤ Tạo mới project Windows Phone 8



Chọn kiểu project Windows Phone -> Windows Phone App





## 4. Bắt đầu với lập trình Windows Phone 8

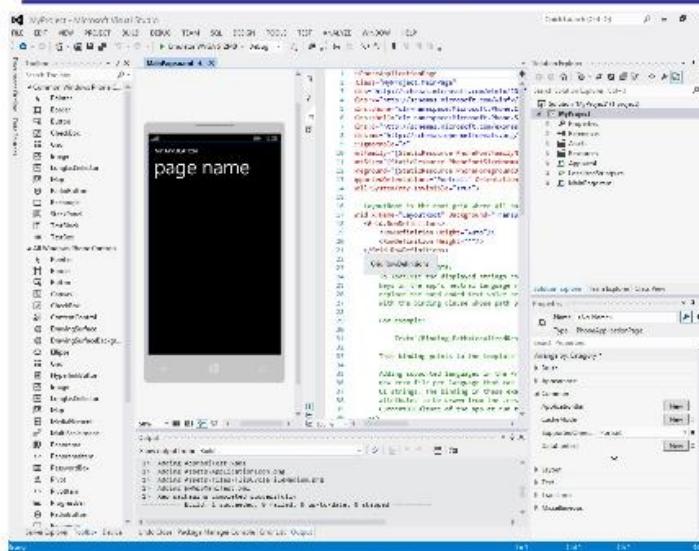
### ➤ Tạo mới project Windows Phone 8



Chọn kiểu loại ứng dụng Windows Phone 8.0



## 4. Bắt đầu với lập trình Windows Phone 8



Nhấn F5 để chạy thử ứng dụng



## 4. Bắt đầu với lập trình Windows Phone 8

### ❖ Các cửa sổ làm việc

The screenshot shows the Windows Phone development environment. On the left is the **Solution Explorer** pane, which lists the project 'MyProject' with files like Properties, References, Assets, Resources, App.xaml, LocalizedStrings.cs, and MainPage.xaml. In the center is the **Properties** pane, showing settings for a PhoneApplicationPage named 'No Name'. On the right is the **XAML designer**, displaying a preview of a page titled 'MY APPLICATION' with the text 'page name'. Below the preview is the XAML code for the page.

Bài 1: Giới thiệu Windows Phone 32

## 4. Bắt đầu với lập trình Windows Phone

### ❖ Cấu trúc thư mục đồ án

The screenshot shows the **Solution Explorer** pane of a Windows Phone project named 'MyProject'. It lists several files and folders: Properties (highlighted), AppManifest.xml (highlighted), AssemblyInfo.cs (highlighted), WMAppManifest.xml (highlighted), References, Assets, Resources, App.xaml, LocalizedStrings.cs, and MainPage.xaml. The **Properties** folder is expanded to show the manifest files.

**AppManifest.xml:** danh sách các resource được thêm vào khi cài đặt ứng dụng

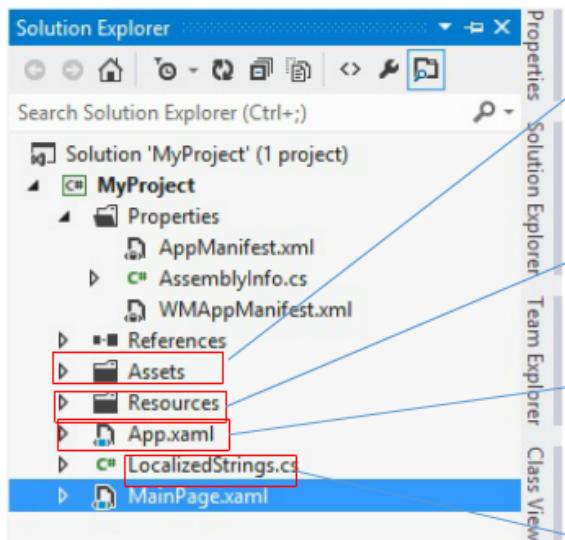
**WMAppManifest.xml:** mô tả chi tiết ứng dụng bao gồm tên ứng dụng, trang khởi tạo, biểu tượng, các tương thích phần cứng.



## 4. Bắt đầu với lập trình Windows Phone



### ❖ Cấu trúc thư mục đồ án



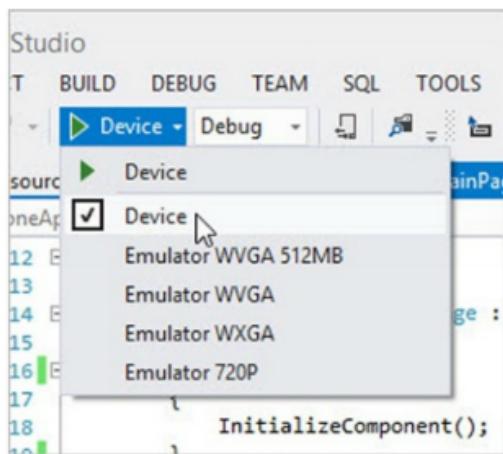
- ✓ The **Assets folder**: chứa hình ảnh của biểu tượng ứng dụng, các tile, các hình khởi động ứng dụng.
- ✓ The **Resources folder**: cho phép ứng dụng đa ngôn ngữ.
- ✓ The **App class** : Chứa các xử lý chung của ứng dụng.
- ✓ The **LocalizedStrings class**: lớp truy cập đến thư mục Resources.



## 4. Bắt đầu với lập trình Windows Phone



### ❖ Chạy ứng dụng từ Visual Studio



- ✓ Mặc định, ứng dụng sẽ chạy trên emulator.
- ✓ Bạn có thể chạy ứng dụng trên thiết bị thật thông qua kết nối cổng USB



## 4. Bắt đầu với lập trình Windows Phone



### ❖ Sử dụng emulator

Emulator Windows Phone 8 hỗ trợ 3 độ phân giải:

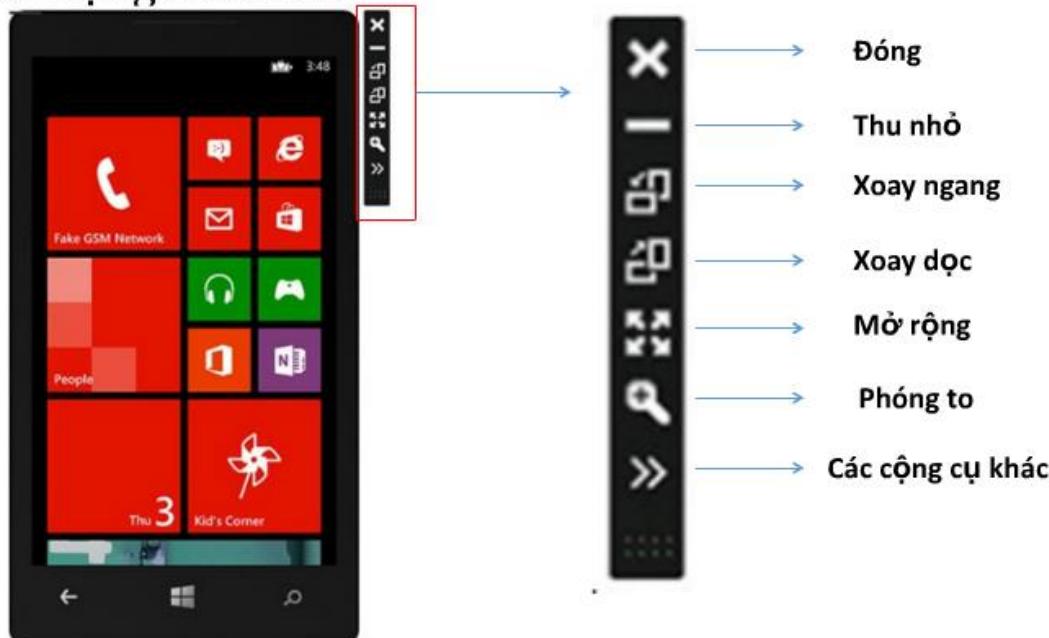
- ✓ WVGA- 800 × 480pixels. Có thêm tùy chọn RAM 512MB.
- ✓ WXGA- 1280 × 768pixels.
- ✓ 720p- 1280 × 720pixels.



## 4. Bắt đầu với lập trình Windows Phone



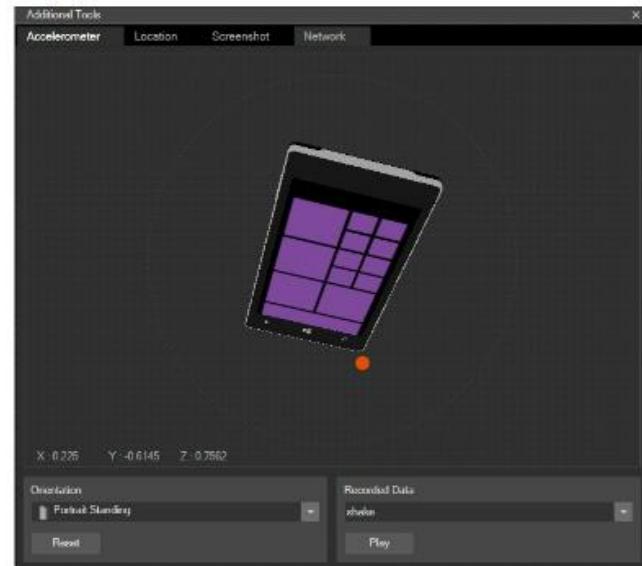
### ❖ Sử dụng emulator



## 4. Bắt đầu với lập trình Windows Phone



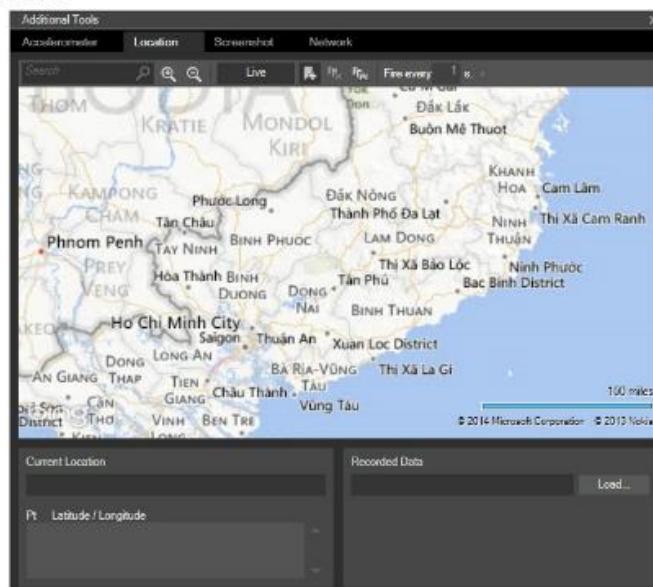
- ❖ Sử dụng emulator
- ✓ Các công cụ khác



## 4. Bắt đầu với lập trình Windows Phone



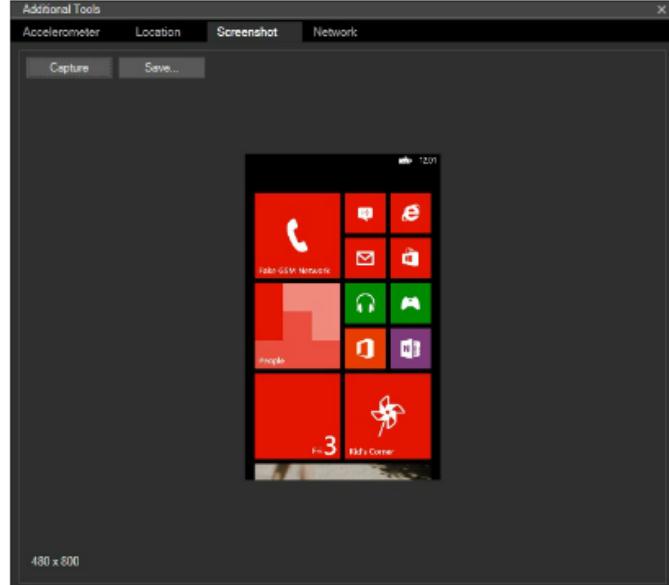
- ❖ Sử dụng emulator
- ✓ Các công cụ khác



## 4. Bắt đầu với lập trình Windows Phone



- ❖ Sử dụng emulator
- ✓ Các công cụ khác



**Screenshot tab**



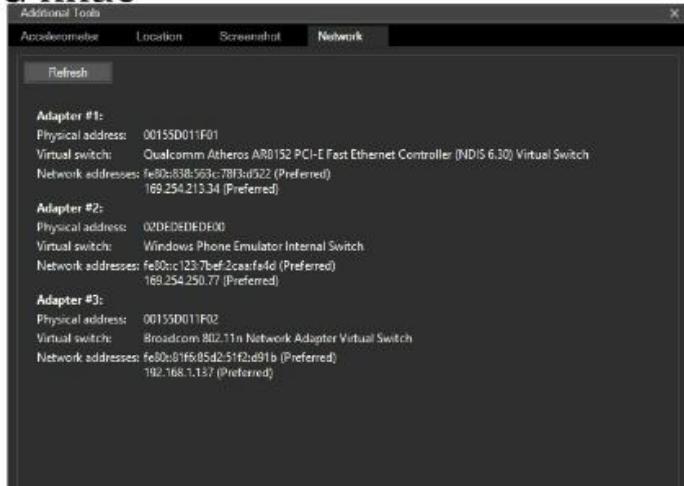
Bài 1: Giới thiệu Windows Phone

40

## 4. Bắt đầu với lập trình Windows Phone



- ❖ Sử dụng emulator
- ✓ Các công cụ khác



**Network tab**



Bài 1: Giới thiệu Windows Phone

41

## 4. Bắt đầu với lập trình Windows Phone



- ❖ Đăng ký thiết bị để cài đặt ứng dụng
  - ✓ Chuẩn bị
- Đăng ký 1 tài khoản lập trình viên tại website <http://dev.windowsphone.com>
- Kết nối điện thoại với Wifi hoặc 3G



## 4. Bắt đầu với lập trình Windows Phone



- ❖ Đăng ký thiết bị để cài đặt ứng dụng
  1. Mở điện thoại và mở khóa màn hình.
  2. Kiểm tra ngày giờ trên điện thoại.
  3. Kết nối điện thoại với máy tính thông qua USB.
  4. Trên máy tính, tìm kiếm “Windows Phone Developer Registration” và chạy ứng dụng tìm được



## 4. Bắt đầu với lập trình Windows Phone



- ❖ Đăng ký thiết bị để cài đặt ứng dụng

### 5. Click the Register button



## 4. Bắt đầu với lập trình Windows Phone



- ❖ Đăng ký thiết bị để cài đặt ứng dụng

### 6. Đăng nhập vào tài khoản Windows Live ID

### 7. Kiểm tra lại trạng thái để chắc chắn điện thoại đã đăng ký thành công

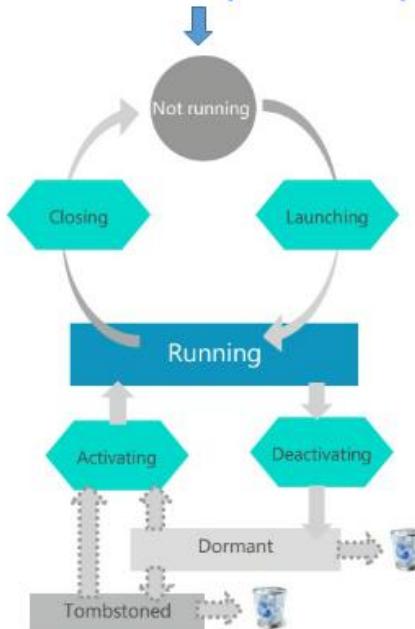
Status: Congratulations! You have successfully unlocked your Windows Phone.



## 5. Vòng đời ứng dụng



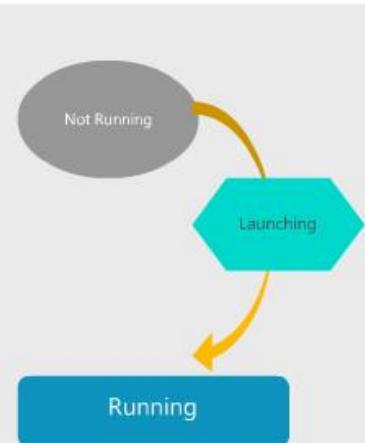
Ứng dụng Windows Phone chuyển đổi qua lại giữa các trạng thái



## 5. Vòng đời ứng dụng



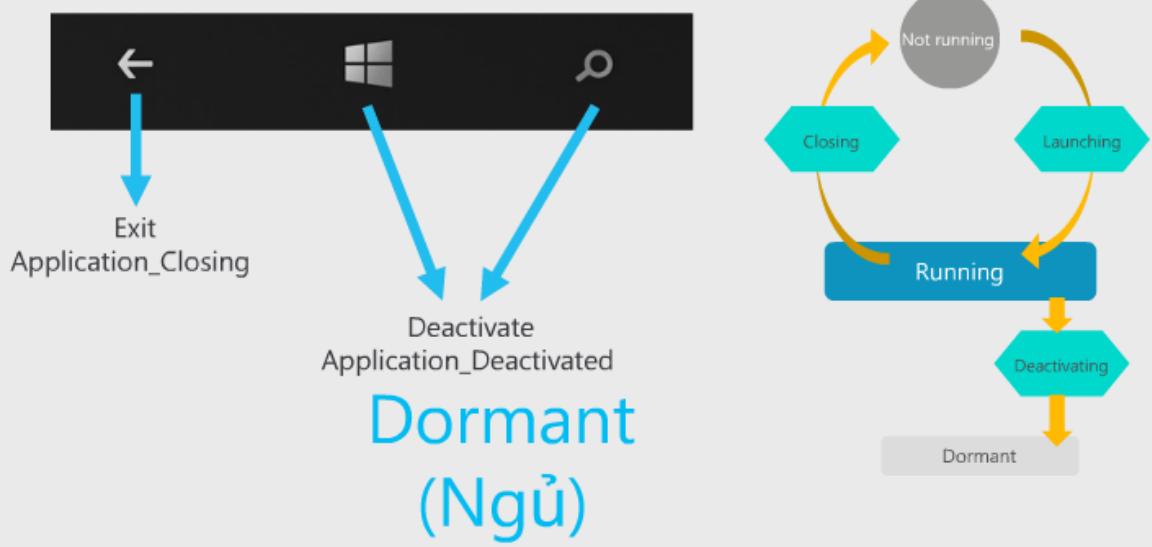
### Running (Đang chạy)



## 5. Vòng đời ứng dụng



### Closing vs Deactivating (Đóng – Tạm ngừng)

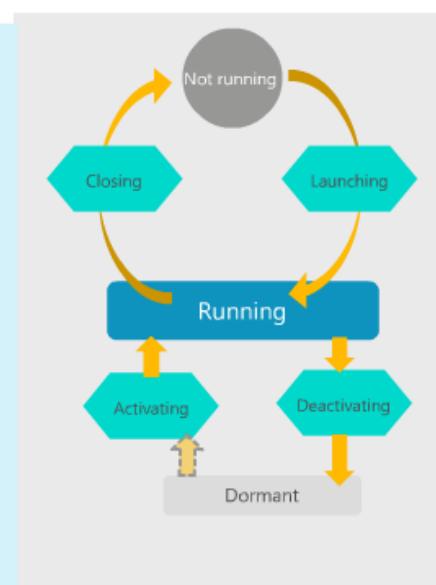


## 5. Vòng đời ứng dụng



### Dormant

- ❖ Trạng thái dormant có thể được gọi một cách chủ động khi
  - ✓ Người dùng mở ứng dụng khác
- ❖ Hoặc được gọi bị động khi
  - ✓ Có cuộc gọi đến
  - ✓ Khóa màn hình sau 1 khoảng thời gian không sử dụng
- ❖ Người dùng có thể resume ứng dụng (nhưng không đảm bảo)
- ❖ Ứng dụng sẽ resume lại trang mà trước khi bị deactivate (do OS quản lý)
- ❖ Các đối tượng vẫn được giữ nguyên khi resume

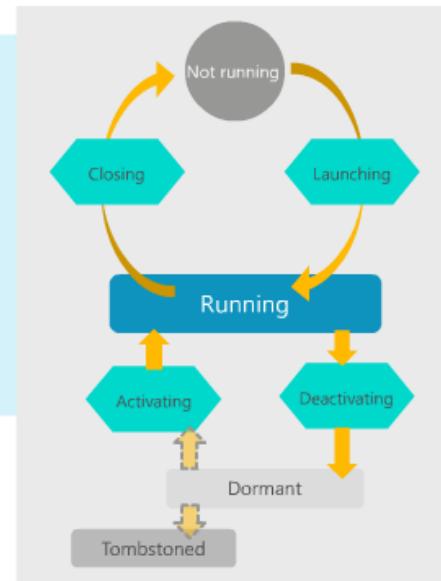


## 5. Vòng đời ứng dụng



### Tombstoned

- Khi bộ nhớ cạn, OS sẽ xóa cached của những ứng dụng đang ở trạng thái dormant lâu nhất.
- ⇒ Quá trình này gọi là Tombstoned
- Ứng dụng vẫn có thể resume nhưng các trạng thái sẽ bị mất – bạn phải load lại



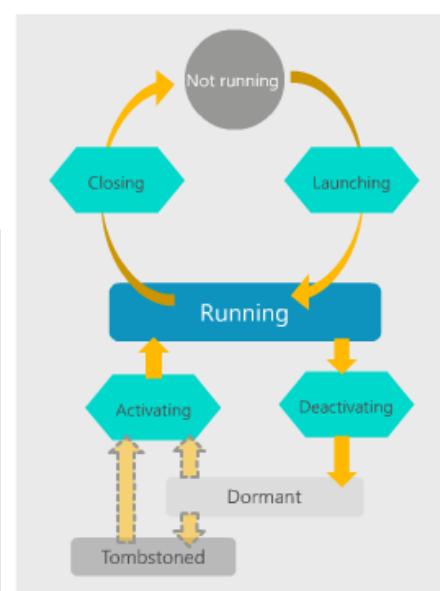
## 5. Vòng đời ứng dụng



### Resume từ Tombstoned hay Dormant?

```

private void Application_Activated(object sender,
ActivatedEventArgs e)
{
    if (e.IsApplicationInstancePreserved)
    {
        // Dormant - objects in memory intact
    }
    else
    {
        // Tombstoned - need to reload
    }
}
  
```

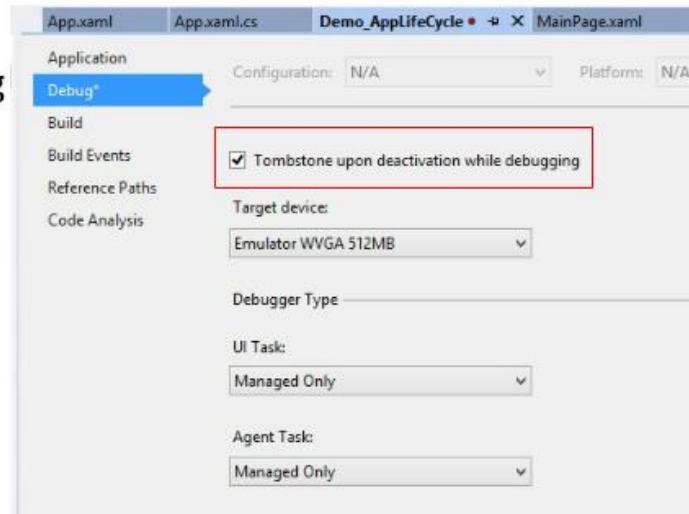


## 5. Vòng đời ứng dụng



### ❖ Debug Tomstone

- ✓ Bạn có thể buộc ứng dụng chuyển qua trạng thái Tomstone khi bị deactivate.



## 5. Vòng đời ứng dụng



### ❖ Sử dụng Application State dictionary

- ✓ Lưu trữ dữ liệu người dùng trong trường hợp ứng dụng bị tomstoned
- ✓ Dữ liệu được lưu dưới dạng dictionary (key-value)
- ✓ Sẽ bị mất nếu reboot thiết bị

```
PhoneApplicationService.Current.State["Url"] = "t3h.com.vn";
```



## 5. Vòng đời ứng dụng



- ❖ Sử dụng Application State dictionary
  - ✓ Lưu dữ liệu người dùng khi ứng dụng bị deactivate

```
protected override void OnNavigatedFrom(System.Windows.Navigation.NavigationEventArgs e)
{
    base.OnNavigatedFrom(e);
    if (e.NavigationMode != System.Windows.Navigation.NavigationMode.Back
        && e.NavigationMode != System.Windows.Navigation.NavigationMode.Forward)
    {
        // If we are exiting the page because we've navigated back or forward,
        // no need to save transient data, because this page is complete.
        // Otherwise, we're being deactivated, so save transient data
        // in case we get tombstoned
        this.State["incompleteEntry"] = this.logTextBox.Text;
    }
}
```



## 5. Vòng đời ứng dụng



- ❖ Sử dụng Application State dictionary
  - ✓ Load dữ liệu người dùng khi ứng dụng resume

```
protected override void OnNavigatedTo(System.Windows.Navigation.NavigationEventArgs e)
{
    base.OnNavigatedTo(e);

    // If the State dictionary contains our transient data,
    // we're being reactivated so restore the transient data
    if (this.State.ContainsKey("incompleteEntry"))
    {
        this.logTextBox.Text = (string)this.State["incompleteEntry"];
    }
}
```



## 5. Vòng đời ứng dụng



```

App.xaml.cs App.xaml < X
1 <Application
2     x:Class="LearnLifecycle.App"
3     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
4     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
5     xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
6     xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone">
7     <!--Application Resources-->
8     <Application.Resources>
9         <local:LocalizedStrings
10            xmlns:local="clr-namespace:LearnLifecycle"
11            x:Key="LocalizedStrings"/>
12     </Application.Resources>
13     <Application.ApplicationLifetimeObjects>
14         <!--Required object that handles lifetime events for the application-->
15         <shell:PhoneApplicationService
16             Launching="Application_Launching" Closing="Application_Closing"
17             Activated="Application_Activated" Deactivated="Application_Deactivated"/>
18     </Application.ApplicationLifetimeObjects>
19 </Application>

```



## 5. Vòng đời ứng dụng



```

// Code to execute when the application is launching ( eg, from Start)
// This code will not execute when the application is reactivated
private void Application_Launching(object sender, LaunchingEventArgs e)
{
}
// Code to execute when the application is activated (brought to foreground)
// This code will not execute when the application is first launched
private void Application_Activated(object sender, ActivatedEventArgs e)
{
}
// Code to execute when the application is deactivated (sent to background)
// This code will not execute when the application is closing
private void Application_Deactivated(object sender, DeactivatedEventArgs e)
{
}
// Code to execute when the application is closing ( eg, user hit Back)
// This code will not execute when the application is deactivated
private void Application_Closing(object sender, ClosingEventArgs e)
{
}

```



## Thảo luận





Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh  
TRUNG TÂM TIN HỌC

Go Screen Capture



## Lập trình Windows Phone

Bài 2. **Tìm hiểu ngôn ngữ XAML  
và các control cơ bản trong Windows Phone**

*Ngành Mạng và Thiết bị di động*



2014



PTOS



## Nội dung



1. Tìm hiểu XAML
2. Các control cơ bản trong Windows Phone



### 1. Tìm hiểu XAML



- ❖ XAML là gì?
  - ✓ XAML – Extensible Application Markup Language – phát âm zammel.
  - ✓ Là một ngôn ngữ đánh dấu dùng để thể hiện các đối tượng trong .NET.



## 1. Tìm hiểu XAML



- ❖ Vì sao sử dụng XAML?
  - ✓ Tách rời nội dung về đồ họa ra khỏi code.
  - ✓ Nâng cao khả năng tùy biến các control trong .NET.



## 1. Tìm hiểu XAML



- ❖ Các cú pháp cơ bản trong XAML
  - ✓ Khai báo 1 thành phần
    - <[Tên thành phần] [Tên thuộc tính]=[giá trị] ></ [Tên thành phần]>
    - Hoặc
    - <[Tên thành phần] [Tên thuộc tính]=[giá trị] />

VD:

```
<TextBlock Text="page name" FontSize ="15" />
```



## 1. Tìm hiểu XAML



- ❖ Các cú pháp cơ bản trong XAML
  - ✓ Khai báo namespace cho trang XAML

VD:

`xmlns:x=http://schemas.microsoft.com/winfx/2006/xaml`

- ✓ Thành phần thuộc tính (Property Element):  
dùng để khai báo các thuộc tính phức tạp

VD:

```
<Button Width="100" Height="50">
    <Button.Background>
        <SolidColorBrush Color="AliceBlue"/>
    </Button.Background>
</Button>
```



## 1. Tìm hiểu XAML



- ❖ Các cú pháp cơ bản trong XAML
  - ✓ Collection Syntax: thuộc tính của một element có thể là tập hợp của nhiều element

VD: Thuộc tính RowDefinitions của Grid

```
<Grid.RowDefinitions>
    <RowDefinition Height="Auto"/>
    <RowDefinition Height="*"/>
    <RowDefinition Height="*"/>
</Grid.RowDefinitions>
```



## 1. Tìm hiểu XAML



- ❖ Các cú pháp cơ bản trong XAML
  - ✓ Attached Property Syntax( thuộc tính đính kèm): một số thuộc tính ở element cha được đính kèm vào element con

VD:

```
<Grid>  
    <Button Grid.Row="1"/>  
</Grid>
```



## 2. Các control cơ bản



1. Layout controls
2. Text controls
3. Selection controls
4. Button
5. Image control



## 2.1 Layout controls



- ❖ Layout controls là những controls kế thừa từ Panel, dùng để chứa và sắp xếp các đối tượng con nằm trong nó. Có 3 loại Layout controls chính:
  - ✓ Grid: xếp các đối tượng con dựa theo dòng và cột
  - ✓ StackPanel: các đối tượng con được xếp liền kề nhau theo chiều dọc hay chiều ngang
  - ✓ Canvas: vị trí các đối tượng con được xác định bởi khoảng cách của nó đến lề trái và lề trên của thành phần chứa nó.



## 2.1 Layout controls



- ❖ Grid
  - ✓ Grid sắp xếp các thành phần con theo dòng và cột.
  - ✓ Dễ dàng mở rộng nếu cần.
  - ✓ Phải định nghĩa các dòng và cột trước khi dùng



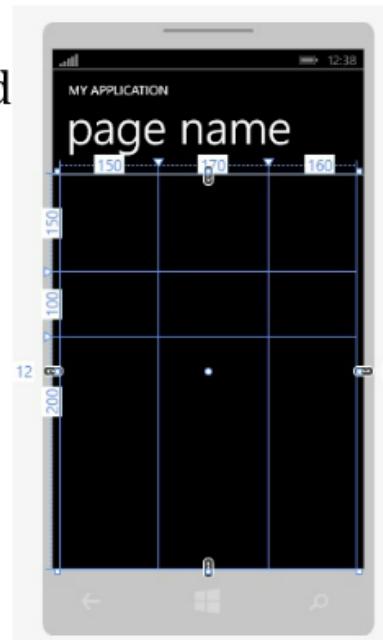
## 2.1 Layout controls



### ❖ Grid

- ✓ Khai báo dòng và cột cho Grid

```
<Grid x:Name="LayoutRoot" >
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="150"/>
    <ColumnDefinition Width="170"/>
    <ColumnDefinition Width="160"/>
  </Grid.ColumnDefinitions>
  <Grid.RowDefinitions>
    <RowDefinition Height="150"/>
    <RowDefinition Height="100"/>
    <RowDefinition Height="200"/>
  </Grid.RowDefinitions>
</Grid>
```



## 2.1 Layout controls



### ❖ Grid

- ✓ Đặt các control vào Grid

```
<Button Content="0,0" Grid.Row="0" Grid.Column="0"/>
<Button Content="0,1" Grid.Row="0" Grid.Column="1"/>
<Button Content="0,2" Grid.Row="0" Grid.Column="2"/>
<Button Content="1,0" Grid.Row="1" Grid.Column="0"/>
<Button Content="1,1" Grid.Row="1" Grid.Column="1"/>
<Button Content="1,2" Grid.Row="1" Grid.Column="2"/>
<Button Content="2,0" Grid.Row="2" Grid.Column="0"/>
<Button Content="2,1" Grid.Row="2" Grid.Column="1"/>
<Button Content="2,2" Grid.Row="2" Grid.Column="2"/>
```



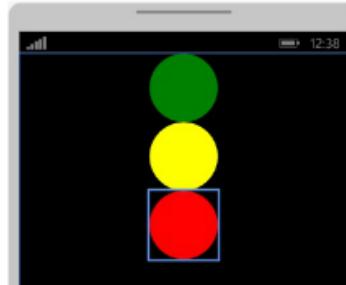


## 2.1 Layout controls

❖ StackPanel

- ✓ StackPanel sắp xếp các thành phần con nằm liên tiếp nhau theo chiều ngang hay dọc.

```
<StackPanel x:Name="LayoutRoot" >
    <Ellipse Width="100" Height="100" Fill="Green"/>
    <Ellipse Width="100" Height="100" Fill="Yellow"/>
    <Ellipse Width="100" Height="100" Fill="Red" />
</StackPanel>
```

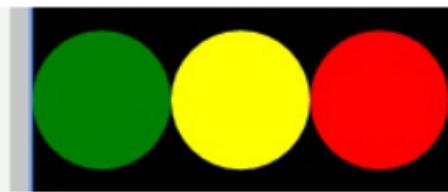


## 2.1 Layout controls

❖ StackPanel

- ✓ Mặc định nó sẽ xếp theo chiều dọc từ trên xuống dưới
- ✓ Dùng thuộc tính **Orientation** để sắp theo chiều ngang

**Orientation="Horizontal"**



## 2.1 Layout controls



### ❖ Canvas

- ✓ Canvas sắp xếp vị trí các thành phần con dựa trên khoảng cách tuyệt đối từ thành phần đó đến lề trên (Canvas.Top) và trái (Canvas.Left) của canvas

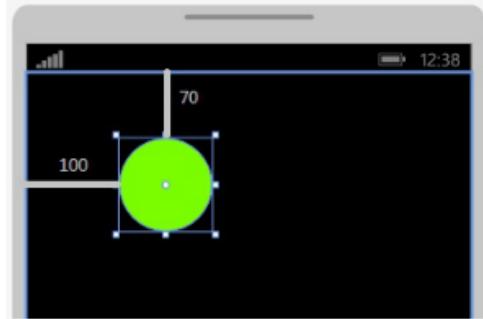
<Canvas>

```

<Ellipse Name="myEllipse"
    Canvas.Left="100"
    Canvas.Top="70"
    Width="100"
    Height="100"
    Fill="LawnGreen" />

```

</Canvas>



Bài 2: Tìm hiểu ngôn ngữ XAML và các control cơ bản

16

## 2.1 Layout controls



### ❖ Canvas

- ✓ Các thành phần con có thể nằm đè lên nhau
- ✓ Khi đó, sử dụng thuộc tính **ZIndex** để đặt thứ tự xếp trên dưới cho các thành phần con

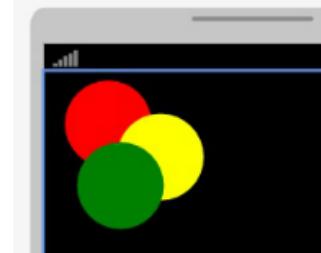
<Canvas>

```

<Ellipse Canvas.Left="38" Canvas.Top="81" Canvas.ZIndex="2"
    Width="100" Height="100" Fill="Green"/>
<Ellipse Canvas.Left="84" Canvas.Top="48"
    Width="100" Height="100" Canvas.ZIndex="1" Fill="Yellow"/>
<Ellipse Canvas.Left="24" Canvas.Top="10"
    Width="100" Height="100" Fill="Red" />

```

</Canvas>



Bài 2: Tìm hiểu ngôn ngữ XAML và các control cơ bản

17

## 2.1 Layout controls



### ❖ Border

- ✓ Border không phải là một layout controls thật sự.
- ✓ Dùng để tạo khung cho các thành phần
- ✓ Thường được dùng làm khung cho Grid và StackPanel

```
<Border BorderThickness="8,4,2,1" >
<Border.BorderBrush>
<LinearGradientBrush EndPoint="0.5,1"
    StartPoint="0.5,0">
    <GradientStop Color="????" Offset="0"/>
    <GradientStop Color="????" Offset="1"/>
</LinearGradientBrush>
</Border.BorderBrush>
<Grid x:Name="ContentPanel" >
</Grid>
</Border>
```



Bài 2: Tìm hiểu ngôn ngữ XAML và các control cơ bản

18

## 2.1 Layout controls



### ❖ ScrollViewer

- ✓ Dùng để chứa nội dung văn bản dài

```
<ScrollViewer Height="550">
<TextBlock>
[noi dung dai]
</TextBlock>
</ScrollViewer>
```



Bài 2: Tìm hiểu ngôn ngữ XAML và các control cơ bản

19

## 2.1 Layout controls



### ❖ ScrollViewer – Scrolling Programmatically

- ✓ Dùng 2 phương thức **ScrollToVerticalOffset()** và **ScrollToHorizontalOffset()** để scrollviewer có thể scroll từng đoạn

```
private void btnScrollTop_Click(object sender, RoutedEventArgs e)
{
    if (myScrollView.ScrollableHeight
        + myScrollView.VerticalOffset < myScrollView.ExtentHeight )
        myScrollView.ScrollToVerticalOffset(5 + myScrollView.Ve
rticalOffset);
}

private void btnScrollBottom_Click(object sender, RoutedEventArgs e)
{
    if (myScrollView.VerticalOffset >= 0)
        myScrollView.ScrollToVerticalOffset(myScrollView.Vertical
Offset - 5);
}
```

Mình dùng Nokia Camera Beta để chụp các ảnh trong bài viết này. Bạn có thể tải nó trên kho ứng dụng (link). Lưu ý phần mềm này khác với Nokia Camera có sẵn trong máy nên khi cài vào máy sẽ có 2 cái Nokia Camera khác nhau, một cái có chữ "Beta" còn cái mặc định thì không có. Phần mềm này cho phép chúng ta chụp ảnh tự động hoàn toàn (Auto) hoặc chụp Manual, tức là cho phép chỉnh tốc độ chụp, thời gian phơi sáng, thay đổi ISO, cự ly lấy nét và cân bằng trắng.  
Thao tác chụp Một điểm đáng tiếc ở Lumia 630 là máy không có phím chụp hình riêng

[Scroll To Top](#) [Scroll To Bottom](#)



Bài 2: Tìm hiểu ngôn ngữ XAML và các control cơ bản

20

## 2. Các control cơ bản



1. Layout controls
2. Text controls
3. Selection controls
4. Button
5. Image control



Bài 2: Tìm hiểu ngôn ngữ XAML và các control cơ bản

21

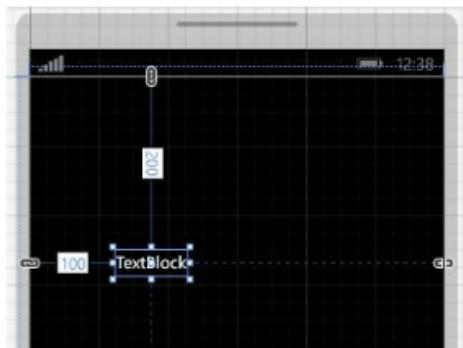
## 2.2 Text controls



### ❖ TextBlock

- ✓ Dùng để hiển thị chuỗi

```
<TextBlock Height="30" Text="TextBlock"
HorizontalAlignment="Left"
VerticalAlignment="Top"/>
```



Bài 2: Tìm hiểu ngôn ngữ XAML và các control cơ bản

22

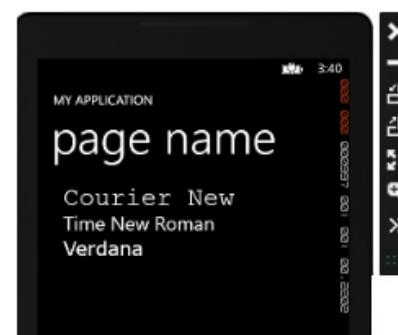
## 2.2 Text controls



### ❖ TextBlock

- ✓ Dùng thành phần <Run/> để thể hiện nhiều chuỗi với định dạng khác nhau

```
<TextBlock FontFamily="Arial">
<Run FontFamily="Courier New" >
    Courier New
</Run>
<LineBreak/>
<Run FontFamily="Time New Roman" >
    Time New Roman
</Run>
<LineBreak/>
<Run FontFamily="Verdana" >
    Verdana
</Run>
</TextBlock>
```



Bài 2: Tìm hiểu ngôn ngữ XAML và các control cơ bản

23

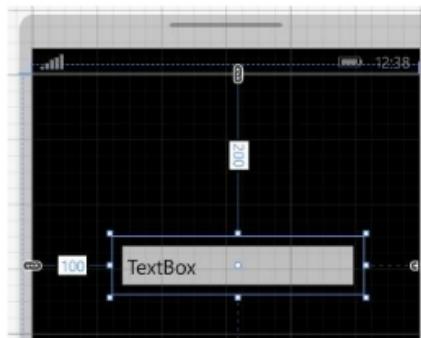
## 2.2 Text controls



### ❖ TextBox

- ✓ Dùng để nhập liệu

```
<TextBox Height="72" HorizontalAlignment="Left"
Margin="100,200,0,0" Text="TextBox" VerticalAlignment="Top" InputScope="Number"/>
```



Bài 2: Tìm hiểu ngôn ngữ XAML và các control cơ bản

24

## 2.2 Text controls



### ❖ TextBox

- ✓ Thuộc tính InputScope dùng để hỗ trợ người dùng nhập liệu

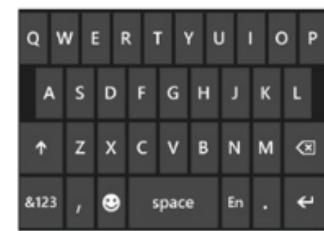
URL

Number

Text



|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |
| . | 0 | ⌫ |



Bài 2: Tìm hiểu ngôn ngữ XAML và các control cơ bản

25

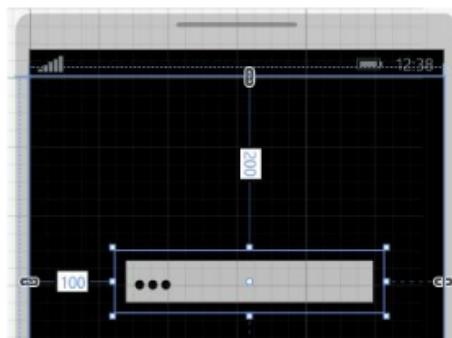
## 2.2 Text controls



### ❖ PasswordBox

- ✓ Dùng để nhập mật khẩu

```
<PasswordBox Height="72" HorizontalAlignment="Left"  
Name="passwordbox1" VerticalAlignment="Top" Passwo  
rd="abc" Width="310"/>
```



## 2. Các control cơ bản



1. Layout controls
2. Text controls
3. Selection controls
4. Button
5. Image control



## 2.3 Selection controls



### ❖ CheckBox

- ✓ Dùng để thể hiện cho những lựa chọn mang 2 giá trị đúng-sai

```
<CheckBox Height="72" IsChecked="true" Content="Bạn có thích Windows Phone"/>
```



## 2.3 Selection controls

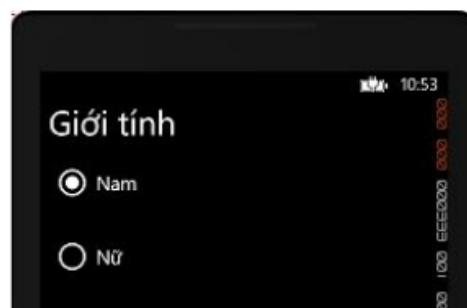


### ❖ RadioButton

- ✓ Dùng để chọn 1 trong nhiều lựa chọn

```
<RadioButton Name="radioButton1" Content="Nam" GroupName="GioiTinh"/>
```

```
<RadioButton Name="radioButton2" Content="Nữ" GroupName="GioiTinh"/>
```



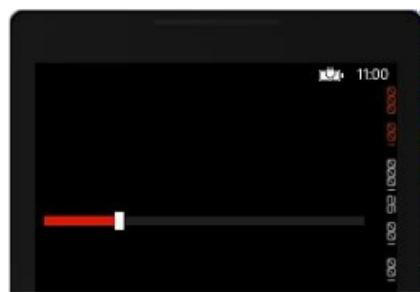
## 2.3 Selection controls



### ❖ Slider

✓ Dùng để thay đổi giá trị cần biến thiên

<Slider Name="slider1" VerticalAlignment = "Top" Width="450"/>



## 2. Các control cơ bản

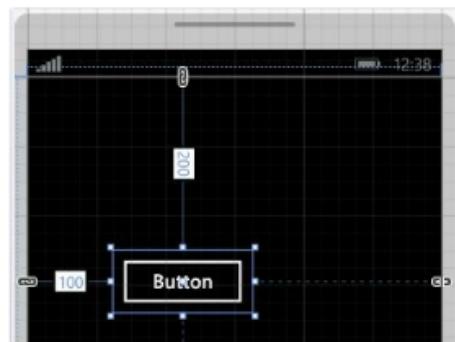
1. Layout controls
2. Text controls
3. Selection controls
4. Button
5. Image control



## 2.4 Button control



```
<Button Content="Button" Height="72"  
       HorizontalAlignment="Left"  
       Name="button1" VerticalAlignment="Top"  
       Width="160" />
```



## 2. Các control cơ bản

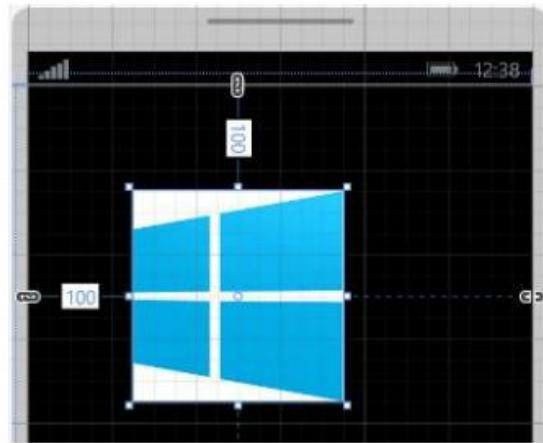
1. Layout controls
2. Text controls
3. Selection controls
4. Button
5. Image control





## 2.5 Image control

```
<Image Source="logoWindowsPhone.jpg" Width="200" Stretch="Fill" Height="200" />
```



Bài 2: Tìm hiểu ngôn ngữ XAML và các control cơ bản

34



## Thảo luận



Bài 2: Tìm hiểu ngôn ngữ XAML và các control cơ bản

35



Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh  
TRUNG TÂM TIN HỌC



Go Screen Capture

## Lập trình Windows Phone

### Bài 3. Các control nâng cao trong Windows Phone

*Ngành Mạng và Thiết bị di động*



2014



PTOS



## Nội dung



1. List controls
2. Menu controls
3. Media controls
4. MessageBox, Popups



### 1. List controls



1. ListBox control
2. LongListSelector control



## 1.1 ListBox control



- ❖ Dùng để hiển thị dữ liệu dạng danh sách
- ❖ Để sử dụng ListBox, ta có 2 cách
  - ✓ Khai báo các Item ngay trên XAML
  - ✓ Load các Item từ một nguồn dữ liệu thông qua data binding



## 1.1 ListBox control



VD: Khai báo các Item từ ngay trên XAML

```
<ListBox>
    <ListBoxItem>
        <TextBlock Text="Nguyễn Hữu Tài"/>
    </ListBoxItem>
    <ListBoxItem>
        <TextBlock Text="Đỗ Thị Thùy Nhung"/>
    </ListBoxItem>
    <ListBoxItem>
        <TextBlock Text="Dương Thùy Trang"/>
    </ListBoxItem>
</ListBox>
```



Designer Time and Run Time



## 1.1 ListBox control



VD: Load các Item từ nguồn dữ liệu

```
<ListBox Name="MyListBox">
    <ListBox.ItemTemplate>
        <DataTemplate>
            <TextBlock Text="{Binding}" />
        </DataTemplate>
    </ListBox.ItemTemplate>
</ListBox>
```

Code XAML

```
public List<string> Items =
    new List<string>
{
    "Nguyễn Hữu Tài",
    "Đỗ Thị Thùy Nhung",
    "Dương Thùy Trang",
};
public MainPage()
{
    InitializeComponent();
    MyListBox.ItemsSource = Items;
}
```

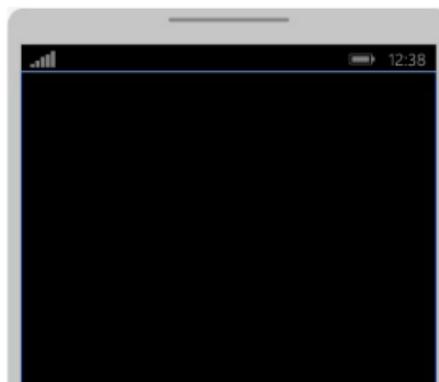
Code Behind



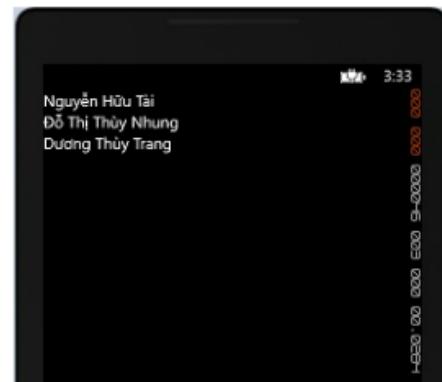
## 1.1 ListBox control



VD: Load các Item từ nguồn dữ liệu



Designer Time



Run Time



## 1.1 ListBox control



- ❖ Các Item của ListBox có thể được kết hợp từ nhiều thành phần

```
<ListBox Height="523">
    <ListBoxItem>
        <StackPanel Orientation="Horizontal">
            <Image Source="/Assets/4.png"/>
            <StackPanel Width="370">
                <TextBlock Text="Họ tên: Nguyễn Hữu Tài"/>
                <TextBlock Text="Tuổi: 30"/>
                <TextBlock Text="Công ty: FPT"/>
            </StackPanel>
        </StackPanel>
    </ListBoxItem>
</ListBox>
```



## 1. List controls



1. ListBox control
2. LongListSelector control



## 1.2 LongListSelector control



- ❖ Windows Phone cung cấp cho ta thêm control LongListSelector để hiển thị dữ liệu dạng danh sách
- ❖ Ngoài việc hiển thị dữ liệu dạng FlatList giống ListBox, LongListSelector còn hỗ trợ hiển thị dữ liệu dạng GroupList.
- ❖ Không hỗ trợ khai báo các Item trên XAML, chỉ có thể dùng data binding để load từ một nguồn dữ liệu



## 1.2 LongListSelector control



VD: Binding LongListSelector đơn giản

```
<phone:LongListSelector x:Name="longlistselector1">
    <phone:LongListSelector.ItemTemplate>
        <DataTemplate>
            <TextBlock Text="{Binding Data}" />
        </DataTemplate>
    </phone:LongListSelector.ItemTemplate>
</phone:LongListSelector>
```



Designer Time



## 1.2 LongListSelector control

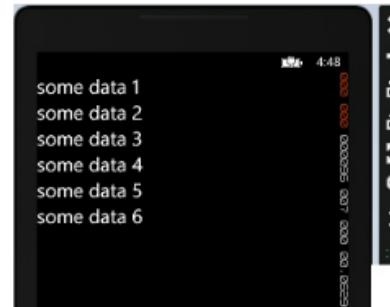


VD: Binding LongListSelector đơn giản

```
class MyObject
{
    public string Data { get; set; }
}

List<MyObject> flatList = new List<MyObject>()
{
    new MyObject() { Data = "some data 1" },
    new MyObject() { Data = "some data 2" },
    new MyObject() { Data = "some data 3" },
    new MyObject() { Data = "some data 4" },
    new MyObject() { Data = "some data 5" },
    new MyObject() { Data = "some data 6" };
}

public MainPage()
{
    InitializeComponent();
    longlistselector1.ItemsSource = flatList;
}
```



Run Time



## 1.2 LongListSelector control



VD: GroupList

- ❖ Khai báo lớp MyObject, đây là kiểu dữ liệu chính của chương trình

```
class MyObject
{
    Public string Category { get; set; }
    public string Data { get; set; }
}
```

- ❖ Khai báo lớp generic Groups để nhóm dữ liệu

```
class Group<TKey, T> : List<T>
{
    public Group(TKey key, IEnumerable<T> items) : base(items)
    {
        this.Key = key;
    }
    public TKey Key { get; set; }
}
```



## 1.2 LongListSelector control



- ❖ Khai báo các resource cho LongListSelector. Các resource được đặt trong tag

```
<phone:PhoneApplicationPage.Resources>
```

```
</phone:PhoneApplicationPage.Resources>
```

- ✓ Resource để binding dữ liệu

```
<DataTemplate x:Key="MyObjectItemTemplate">
```

```
    <TextBlock Text="{Binding Data}" Style={StaticResource PhoneTextLargeStyle}/>
```

```
</DataTemplate>
```

- ✓ Resource để định dạng cho Jumplist

```
<phone:JumpListItemBackgroundConverter x:Key="BackgroundConverter"/>
```

```
<phone:JumpListItemForegroundConverter x:Key="ForegroundConverter"/>
```



## 1.2 LongListSelector control



- ❖ Khai báo các resource cho LongListSelector.
- ✓ Resource định dạng cho JumpList

```
<Style x:Key="MyObjectJumpListStyle" TargetType="phone:LongListSelector">
    <Setter Property="GridCellSize" Value="120,120"/>
    <Setter Property="LayoutMode" Value="Grid" />
    <Setter Property="ItemTemplate"
        Value="{StaticResource GroupTile}" />
</Style>
```

- ✓ Resource định dạng cho GroupList



## 1.2 LongListSelector control



- ❖ Khai báo các resource cho LongListSelector.

- ✓ Resource định dạng cho GroupList

```
<DataTemplate x:Key="GroupTile">
    <Grid Width="100" Height="100"
        Background="{Binding Converter={StaticResource BackgroundConverter}}">
        <TextBlock Text="{Binding Title}"
            Style="{StaticResource PhoneTextExtraLargeStyle}"
            Foreground="{Binding Converter={StaticResource ForegroundConverter}}"/>
    </Grid>
</DataTemplate>
```



## 1.2 LongListSelector control



- ❖ Code behind

- ✓ Hàm khởi tạo nguồn dữ liệu

```
public void loadDataSample()
{
    var flatList = new List<MyObject>(){
        new MyObject() { Category = "A", Data = "some data 1" },
        new MyObject() { Category = "A", Data = "some data 2" },
        new MyObject() { Category = "B", Data = "some data 3" },
        new MyObject() { Category = "C", Data = "some data 4" },
        new MyObject() { Category = "C", Data = "some data 5" },
    };
    var groups = (from obj in flatList
        group obj by obj.Category into g
        orderby g.Key
        select new Group<string, MyObject>(g.Key, g));
    LongListSelector1.ItemsSource = groups.ToList();
}
```



## 1.2 LongListSelector control



- ❖ Code behind
- ✓ Gọi hàm khởi tạo trong phương thức khởi tạo của hàm MainPage

```
public MainPage()
{
    InitializeComponent();
    loadDataSample();
}
```



## 1.2 LongListSelector control



- ❖ Kết quả



## 2. Menu controls



### 1. ApplicationBar

- ❖ ApplicationBar được thiết kế để đặt một số chức năng của chương trình, ví dụ chức năng thêm, xóa, sửa dữ liệu



## 2. Menu controls



### 1. ApplicationBar

- ❖ ApplicationBar được thiết kế để đặt một số chức năng của chương trình, ví dụ chức năng thêm, xóa, sửa dữ liệu
- ❖ ApplicationBar thuộc namespace shell, được định nghĩa trong cặp thẻ

```
<phone:PhoneApplicationPage.ApplicationBar>  
</phone:PhoneApplicationPage.ApplicationBar>
```



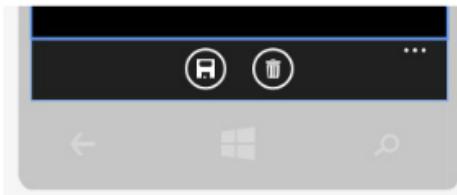
## 2. Menu controls



### 1. ApplicationBar

#### ❖ Định nghĩa ApplicationBar

```
<shell:ApplicationBar>
    <shell:ApplicationBarIconButton IconUri="/assets/save.png"
        Text="save" Click="AppBarIconButton_Click_1" />
    <shell:ApplicationBarIconButton
        IconUri="/assets/cancel.png" Text="cancel" />
</shell:ApplicationBar>
```



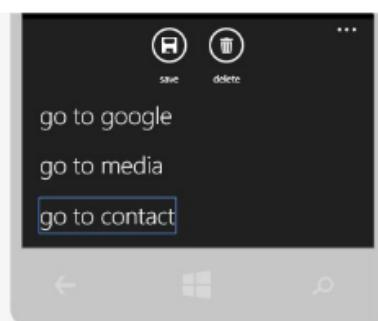
## 2. Menu controls



### 1. ApplicationBar

#### ❖ ApplicationBar còn hỗ trợ tạo menu

```
<shell:ApplicationBar>
    <shell:ApplicationBar.MenuItems>
        <shell:ApplicationBarMenuItem Text="Go to Google"/>
        <shell:ApplicationBarMenuItem Text="Go to Media" />
        <shell:ApplicationBarMenuItem Text="Go to Contact" />
    </shell:ApplicationBar.MenuItems>
</shell:ApplicationBar>
```



### 3. Media controls



1. Image
2. MediaElement

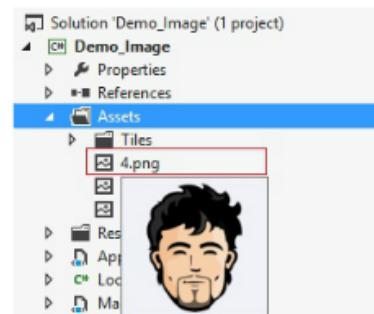
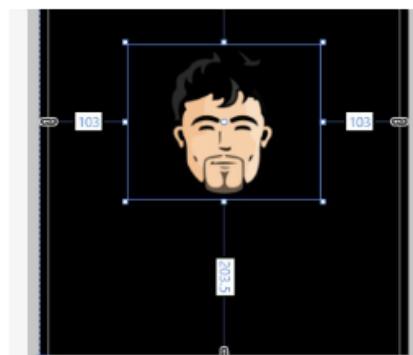


#### 3.1 Image



- ❖ Image control dùng để thể hiện hình ảnh trên ứng dụng
- ❖ Cách khai báo image control

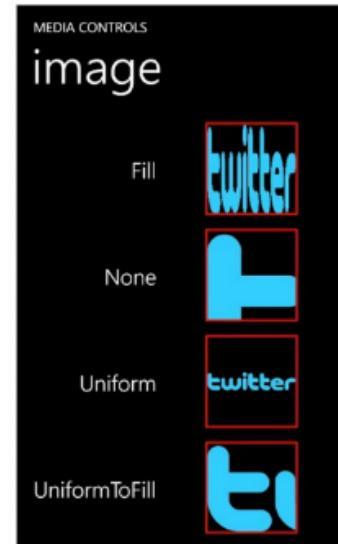
```
<Image Source= "/Assets/4.png"  
      Height="200" Width="250" />
```



## 3.1 Image



- ❖ Thuộc tính Stretch để định dạng cách hiển thị hình ảnh. Các giá trị của Stretch
  - ✓ **Fill** : hình sẽ được co giãn để đáp ứng kích thước đã định, không quan tâm đến tỉ lệ.
  - ✓ **None** : giữ nguyên kích thước cũng như tỉ lệ của hình
  - ✓ **The Uniform** : co giãn hình tối đa theo kích thước đã định nhưng vẫn đảm bảo được tỉ lệ và vẫn nằm lọt trong khung hình
  - ✓ **The UniformToFill** : co giãn hình tối đa theo kích thước đã định, vẫn đảm bảo được tỉ lệ nhưng không đảm bảo việc hình có bị mất không



## 3. Media controls



1. Image
2. MediaElement

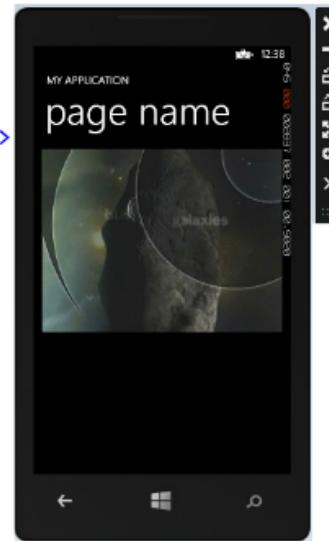


## 3.2 MediaElement



- ❖ Để chơi các file media như nhạc hay video, ta sử dụng MediaElement control.
- ❖ Cách khai báo MediaElement

```
<MediaElement Name="MediaWindow"
    Source="Assets/hm_helix_twist_320x240.wmv" />
```

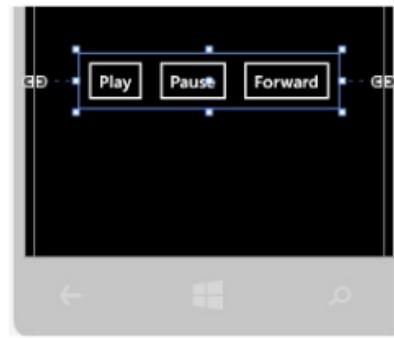


## 3.2 MediaElement



- ❖ Định nghĩa các nút để điều khiển

```
<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">
    <Button Name="btnPlay" Click="btnPlay_Click" Content="Play"/>
    <Button Name="btnPause" Click="btnPause_Click" Content="Pause"/>
    <Button Name="btnForward" Click="btnForward_Click" Content="Forward"/>
</StackPanel>
```



## 3.2 MediaElement



```

private void btnPlay_Click(object sender, RoutedEventArgs e)
{
    MediaWindow.Play();
}
private void btnPause_Click(object sender, RoutedEventArgs e)
{
    MediaWindow.Pause();
}
private void btnForward_Click(object sender, RoutedEventArgs e)
{
    TimeSpan current = MediaWindow.Position;
    MediaWindow.Position = current.Add(new TimeSpan(0, 0, 1));
}

```

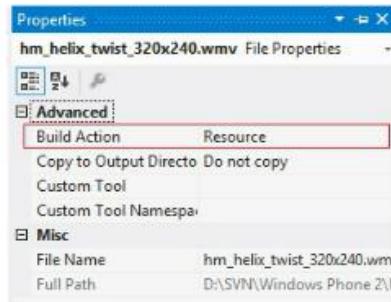


## 3.2 MediaElement



❖Ghi chú: đổi thuộc tính Build Action của file video thành Resource

- ✓ Click phải vào file video, chọn Properties
- ✓ Tìm đến thuộc tính Build Action, đổi giá trị thành Resource



## 4. Popup, MessageBox



1. MessageBox
2. Popup

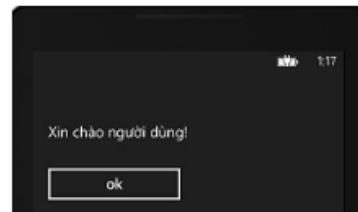


### 4.1 MessageBox



- ❖ MessageBox cho phép ứng dụng đưa ra những thông báo cho người dùng dưới dạng panel nằm đè lên màn hình hiện tại.
- ❖ Với MessageBox, ta chỉ có thể tương tác với người dùng bằng những câu thông báo và các nút nhấn đơn giản như Yes, No, Cancel
- ❖ MessageBox được gọi từ code behind

```
MessageBox.Show("Xin chào người dùng!");
```





## 4.1 MessageBox

- ❖ Nếu MessageBox có 2 nút nhấn OK- Cancel, ta dùng biến kiểu **MessageBoxResult** để nhận kết quả

**MessageBoxResult result =**

```
MessageBox.Show("Bạn có lưu dữ liệu không!",  

    "Câu hỏi", MessageBoxButton.OKCancel);  

if (result == MessageBoxResult.OK)  

{ }
```



## 4.2 Popup

- ❖ Tương tự MessageBox, nhưng popup cho phép ta thay đổi giao diện và nhận dữ liệu nhập từ người dùng.
- ❖ Ta có thể định dạng các popup trên code XAML và gọi popup từ code behind



## 4.2 Popup



```
<Popup x:Name="MyPopup" VerticalOffset="150" HorizontalOffset="120">
    <Popup.Child>
        <Border BorderBrush="Red" BorderThickness="3">
            <TextBlock Margin="5" Text="Text for my popup"
                Foreground="White"/>
        </Border>
    </Popup.Child>
</Popup>
```



## 4.2 Popup



- ❖ Để ẩn hay hiện popup, ta dùng thuộc tính **IsOpen**
  - ✓ **IsOpen="True"** : hiện popup
  - ✓ **IsOpen="False"** : ẩn popup



## Thảo luận





Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh  
TRUNG TÂM TIN HỌC

Go Screen Capture

Lập trình Windows Phone

Bài 4. Các loại ứng dụng  
và chuyển trang trong Windows Phone

*Ngành Mạng và Thiết bị di động*

2014

 Windows Phone



PTOS

Windows Phone

## Nội dung



1. Các loại ứng dụng
2. Chuyển trang (Navigation)



### 1. Các loại ứng dụng



1. Single Page
2. Pivot
3. Panorama



## 1.1 Single Page



- ❖ Đây là loại ứng dụng cơ bản nhất của Windows Phone
- ❖ Bao gồm những trang riêng lẻ
- ❖ Di chuyển giữa các trang bằng cách sử dụng NavigationService.



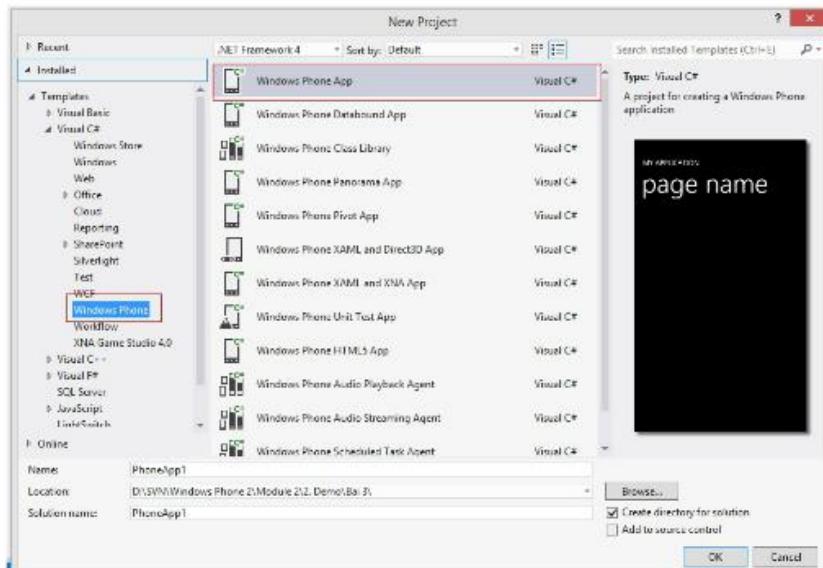
Bài 4: Các loại ứng dụng và chuyển trang trong Windows Phone

4

## 1.1 Single Page



- ❖ Để tạo project dạng Single Page, ta chọn template mặc định khi tạo mới project



Bài 4: Các loại ứng dụng và chuyển trang trong Windows Phone

5

## 1. Các loại ứng dụng



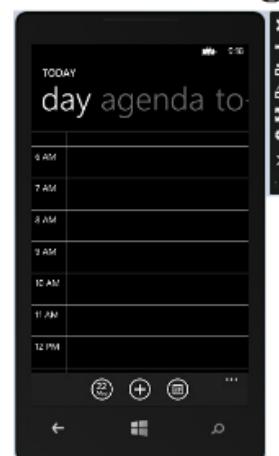
1. Single Page
2. Pivot
3. Panorama



### 1.2 Pivot



- ❖ Pivot control tạo ra ứng dụng kiểu tabbed
- ❖ Bản chất pivot control là 1 container chứa 1 hay nhiều PivotItem control.
- ❖ Ta có thể di chuyển giữa các view bằng cách
  - ✓ Panning
  - ✓ Flick
  - ✓ Tap on header



## 1.2 Pivot



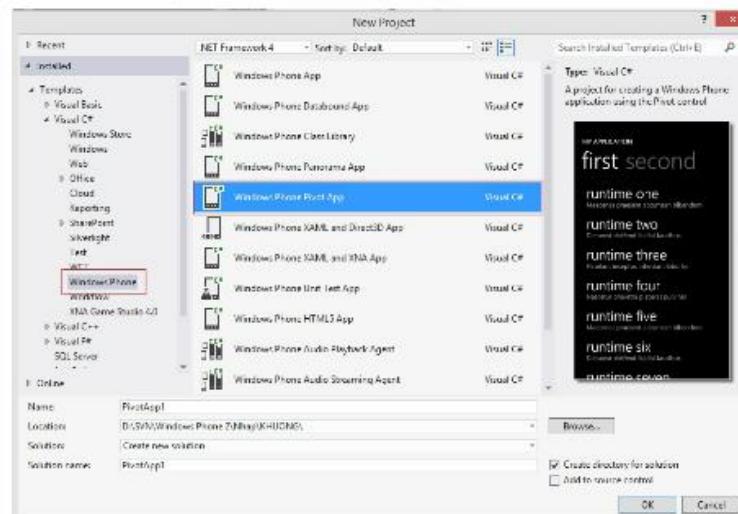
- ❖ Khi nào sử dụng Pivot
  - ✓ Lọc các bộ dữ liệu lớn
  - ✓ Xem nhiều bộ dữ liệu



## 1.2 Pivot



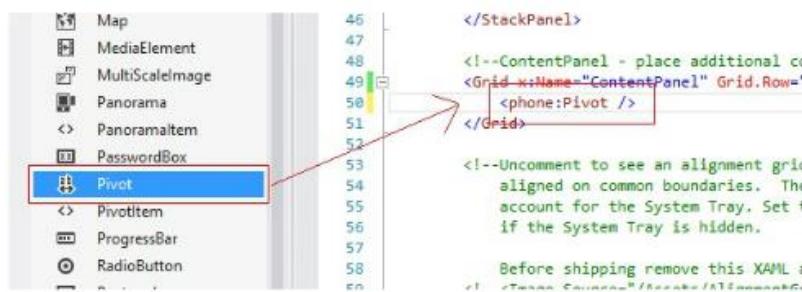
- ❖ Để tạo một Pivot application, ta có thể:
  - ✓ Chọn template “Windows Phone Pivot App” khi tạo project mới



## 1.2 Pivot



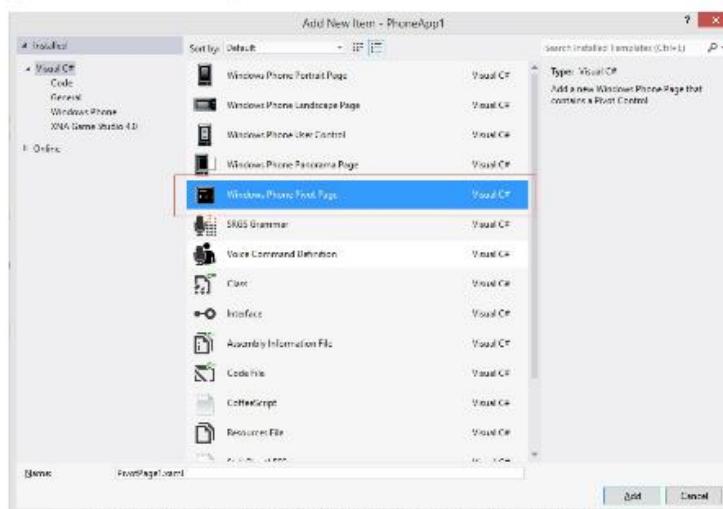
- ❖ Để tạo một Pivot application, ta có thể:
  - ✓ Chọn template mặc định và kéo thả control Pivot từ ToolBox



## 1.2 Pivot



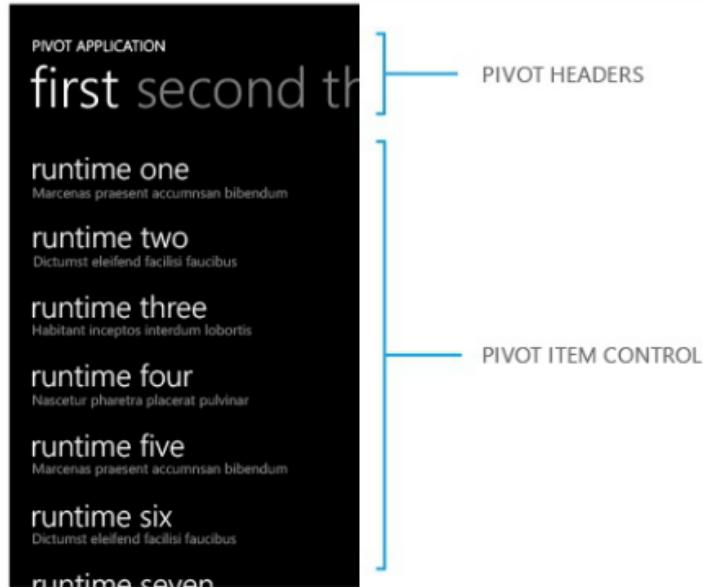
- ❖ Để tạo một Pivot application, ta có thể:
  - ✓ Thêm 1 page dạng “Windows Phone Pivot Page” vào project hiện tại



## 1.2 Pivot



### ❖ Cấu trúc 1 trang Pivot



Bài 4: Các loại ứng dụng và chuyển trang trong Windows Phone

12

## 1.2 Pivot



### ❖ Cách khai báo 1 trang Pivot

```
<phone:Pivot Title="MY APPLICATION">
    <!--Pivot item one-->
    <phone:PivotItem Header="first">
        </phone:PivotItem>
    <!--Pivot item two-->
    <phone:PivotItem Header="second">
        </phone:PivotItem>
</phone:Pivot>
```



Bài 4: Các loại ứng dụng và chuyển trang trong Windows Phone

13

## 1.2 Pivot



- ❖ Các lưu ý khi thiết kế trang Pivot
  - ✓ Rút ngắn số trang pivot (tối đa 4 trang)
  - ✓ Các trang pivot nên hiển thị cùng loại dữ liệu
  - ✓ Rút ngắn tiêu đề từ 1 đến 2 từ
  - ✓ Không thể thay đổi chiều cao của header
  - ✓ Không nên dùng các control như slider, Toogle Button, hay map trong trang pivot



## 1. Các loại ứng dụng



1. Single Page
2. Pivot
3. Panorama



## 1.3 Panorama



- ❖ Panorama control tạo ra một ứng dụng gồm nhiều trang trải dài theo chiều ngang



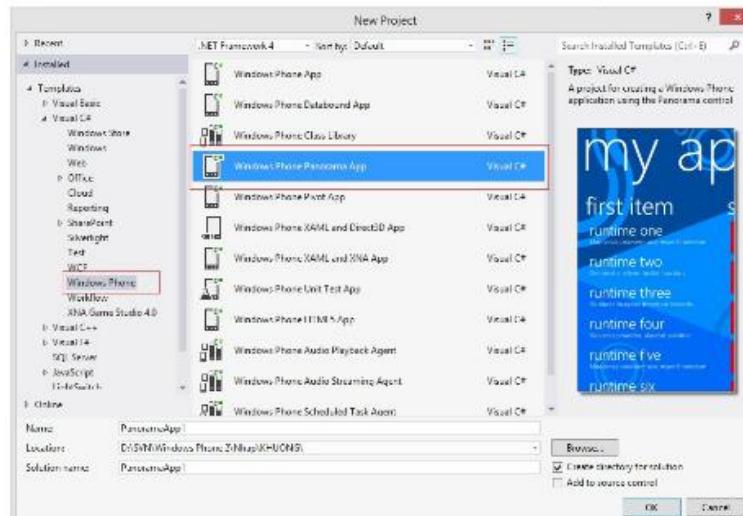
Bài 4: Các loại ứng dụng và chuyển trang trong Windows Phone

16

## 1.3 Panorama



- ❖ Để tạo một Panorama application ta có thể
  - ✓ Chọn template project dạng Windows Phone Panorama App khi tạo mới project



Bài 4: Các loại ứng dụng và chuyển trang trong Windows Phone

17

## 1.3 Panorama



- ✓ Kéo thả panorama control từ toolbox vào giao diện

```

<TextBlock Text="MY APPLICATION" Style="{StaticResource PhoneTextNormalStyle}"/>
<TextBlock Text="page name" Margin="9,-7,0,0" Style="{StaticResource PhoneTextSubtleStyle}"/>

```

```

<!--ContentPanel - place additional content here-->
<Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
    <phone:Panorama />
</Grid>

```

```



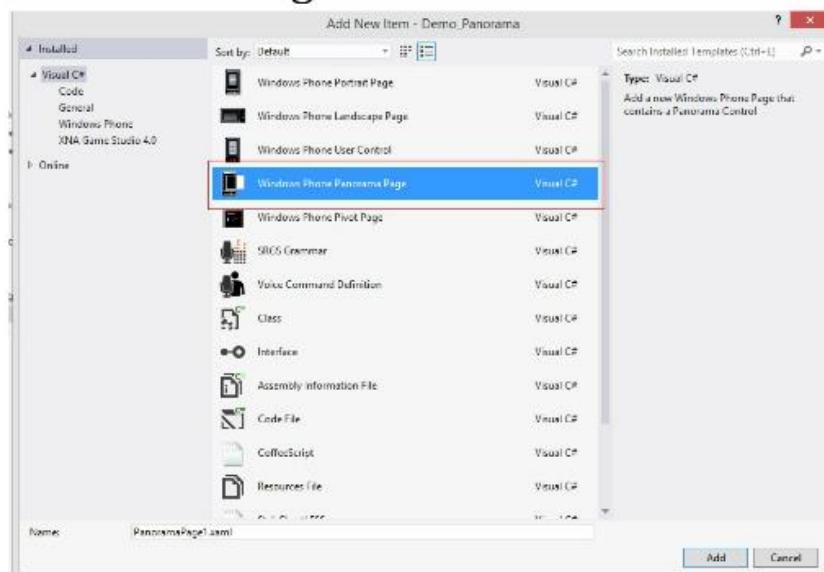
```



## 1.3 Panorama



- ✓ Thêm 1 trang dạng Windows Phone Panorama Page



## 1.3 Panorama



### ❖ Cấu trúc 1 panorama control



## 1.3 Panorama



### ❖ Cách khai báo panorama control

```
<phone:Panorama x:Name="MyPanorama">
    <phone:Panorama.Background>
        <SolidColorBrush Color="White"/>
    </phone:Panorama.Background>
    <!--Panorama item 1-->
    <phone:PanoramaItem>
    </phone:PanoramaItem>
    <!--Panorama item 2-->
    <phone:PanoramaItem>
    </phone:PanoramaItem>
</phone:Panorama>
```



## 2. Chuyển trang (Navigation)



1. Chuyển trang giữa các page
2. Truyền tham số giữa các trang
3. Làm việc với Navigation Backstack



### 2.1 Chuyển trang giữa các page



- ❖ Sử dụng phương thức Navigate() trong lớp  
NavigationService

```
NavigationService.Navigate(new Uri("/Pages/Page2.xaml",  
                               UriKind.Relative));
```



## 2.1 Chuyển trang giữa các page



- ❖ UriKind
  - ✓ Absolute : đường dẫn tuyệt đối
  - ✓ Relative : đường dẫn tương đối
  - ✓ RelativeOrAbsolute : chưa xác định



## 2.1 Chuyển trang giữa các page



- ❖ Các sự kiện quan trọng
  - ✓ OnNavigatedTo(): sự kiện navigate từ trang khác đến trang hiện tại
  - ✓ OnNavigatedFrom(): sự kiện navigate từ trang hiện tại đến trang khác
  - ✓ Loaded: sự kiện khi page đã load hoàn tất.



## 2.2 Truyền tham số giữa các trang



- ❖ Tương tự cách truyền tham số trên web

VD: Truyền giá trị ID từ page1 sang page2 khi chuyển trang từ page1 qua page2

Trên page 1:

```
NavigationService.Navigate(new Uri("/Page2.xaml?ID=2", UriKind.Relative));
```



## 2.2 Truyền giá trị giữa các trang



Trên page 2: nhận tham số

```
protected override void OnNavigatedTo(NavigationEventArgs e)
{
    base.OnNavigatedTo();
    if (NavigationContext.QueryString.ContainsKey("ID"))
    {
        int ketqua = int.Parse(NavigationContext
            .QueryString["ID"]);
    }
}
```

Lớp NavigationContext cung cấp thuộc tính kiểu collection tên là QueryString, chứa toàn bộ tham số được truyền đến dưới dạng.



## 2.2 Truyền giá trị giữa các trang



Hoặc có thể dùng cách sau

```
protected override void OnNavigatedTo(NavigationEventArgs e)
{
    base.OnNavigatedTo();
    string parameter = string.Empty;
    if (NavigationContext.QueryString.
        TryGetValue("parameter", out parameter))
    {
        //do something with the parameter
    }
}
```



## 2.2 Truyền giá trị giữa các trang



- ❖ Sử dụng ký tự ‘&’ để truyền nhiều tham số cùng lúc

```
NavigationService.Navigate(
    new Uri("/page1.xaml?ID=2&Ten=abv", UriKind.Relative));
```

- ❖ Sử dụng string format

```
string stringVariable = "This is a test string";
NavigationService.Navigate(new Uri(
    string.Format("/Page1.xaml?parameter={0}", stringVariable),
    UriKind.Relative));
```



## 2.3 Làm việc với Navigation BackStack



- ❖ BackStack chứa danh sách các trang mà người dùng đã di chuyển đến trong quá trình sử dụng app.
- ❖ Khi người dùng nhấn phím Back, trang cuối cùng được lưu trong BackStack sẽ được lấy ra.
- ❖ Để trở về trang trước đó, dùng phương thức GoBack() trong lớp NavigationService.
- ❖ Để xóa BackStack, sử dụng phương thức RemoveBackStackEntry().



## Thảo luận





Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh  
TRUNG TÂM TIN HỌC



## Lập trình Windows Phone

### Bài 5: Data binding cơ bản

Ngành Mạng và Thiết bị di động



2014



PTOS



## Nội dung



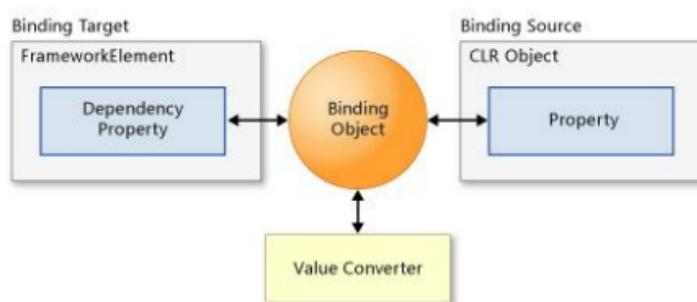
1. Data binding là gì
2. Các thành phần chính trong Data Binding
3. Cách tạo binding trong Windows Phone



### 1. Data binding là gì



- ❖ Data binding là quá trình thiết lập một kênh giao tiếp giữa 1 thành phần giao diện và nguồn dữ liệu cung cấp cho thành phần đó



## 2. Các thành phần trong Data Binding



- ❖ Binding target: đối tượng muốn binding
- ❖ Source: đối tượng cung cấp dữ liệu, nó được gán vào DataContext của thành phần UI cần binding.
- ❖ Path: tên thuộc tính của đối tượng dữ liệu đang làm source.



## 2. Các thành phần trong Data Binding



- ❖ Mode: chiều tương tác của dữ liệu với UI
  - ✓ OneTime: dữ liệu được load một lần duy nhất khi khởi tạo UI.
  - ✓ OneWay: mặc định, thành phần hiển thị được cập nhật từ đối tượng nguồn.
  - ✓ TwoWay: thành phần hiển thị được cập nhật từ đối tượng nguồn và ngược lại



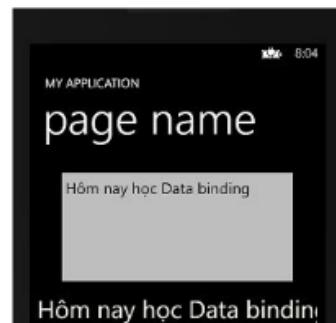
### 3. Cách tạo data binding trong Windows Phone



❖ Binding control to control

❖ Ví dụ:

```
<TextBox Height="200" Width="400" Text="Hôm nay học Data binding" x:Name="textBox1" />
<TextBlock x:Name="textBlock1" Text="{Binding ElementName=textBox1, Path=Text, Mode=OneTime}" />
```



Bài 5: Data binding cơ bản

6

### 3. Cách tạo data binding trong Windows Phone



❖ Binding control to object

```
<StackPanel x:Name="Author">      <TextBlock Text="Name"/>
    <TextBlock Text="{Binding Path=Name}"/>
    <TextBlock Text="Surname"/>
    <TextBlock Text="{Binding Path=Surname}"/>
</StackPanel>
```

Lưu ý: `<TextBlock Text="{Binding Path=Name}"/>`  
`<TextBlock Text="Name"/>`



Bài 5: Data binding cơ bản

7

### 3. Cách tạo data binding trong Windows Phone



```
public class Person
{
    public string Name {get;set;}
    public string SurName { get; set; }
}
```



### 3. Cách tạo data binding trong Windows Phone



```
public MainPage()
{
    InitializeComponent();
    Person person = new Person();
    person.Name = "An Khuong";
    person.SurName = "Nguyen";
    Author.DataContext = person;
}
```



### 3. Cách tạo data binding trong Windows Phone



#### ❖ Binding ListBox với List Object

```
<ListBox x:Name="lstNguoiDung" Height="523">
    <ListBox.ItemTemplate>
        <DataTemplate>
            <StackPanel>
                <Image Source="{Binding ImgUrl}" VerticalAlignment="Top"/>
                <StackPanel Width="370" Margin="10,0,0,0">
                    <TextBlock Text="{Binding Name}"/>
                    <TextBlock Text="{Binding Age}"/>
                    <TextBlock Text="{Binding Company}"/>
                </StackPanel>
            </StackPanel>
        </DataTemplate>
    </ListBox.ItemTemplate>
</ListBox>
```



Bài 5: Data binding cơ bản

10

### 3. Cách tạo data binding trong Windows Phone



```
public class Employee
{
    public string Name { get; set; }
    public int Age { get; set; }
    public string ImgUrl { get; set; }
    public string Company { get; set; }
}
```



Bài 5: Data binding cơ bản

11

### 3. Cách tạo data binding trong Windows Phone



```
List<Person> ListPerson = new List<Person>();  
public MainPage()  
{  
    InitializeComponent();  
    Listperson = Person.GetData();  
    PersonList.ItemsSource = ListPerson;  
}
```



### Thảo luận

