


 Windows Phone

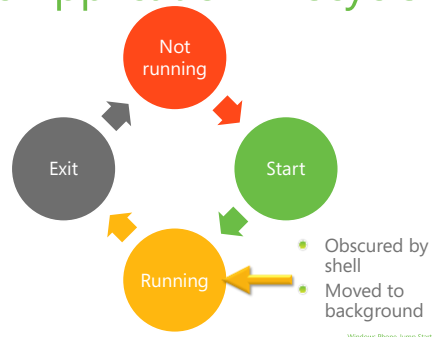
Application Lifecycle

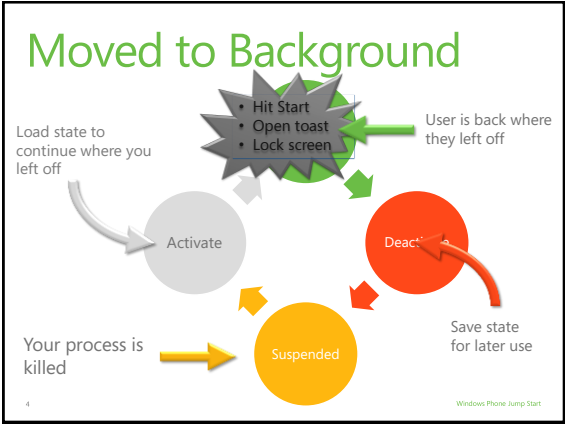
Saving and Restoring Application State

A World of Interruptions



Basic Application Lifecycle





WP7 App Management

- Windows Phone execution model is designed to provide end users with a fast, responsive experience
 - Only one application can run at a time
 - System terminates application when user navigates away
- App is **Tombstoned** when user navigates away
 - System saves state information then terminates your app
 - When user navigates back to the application, system restarts the app and passes state information back
- Developer must write code to respond to lifecycle events
 - Save and restore state
 - Maintain illusion that the application is running continuously

5

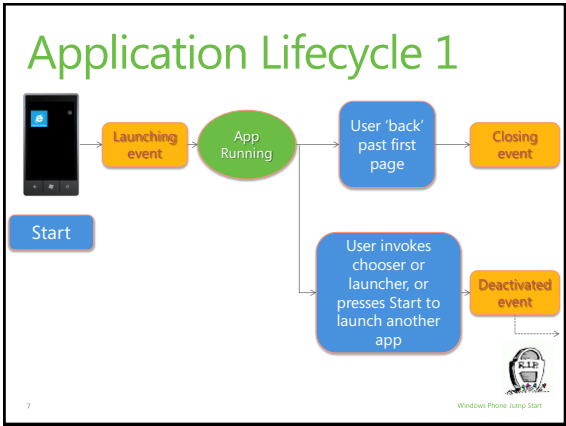
Windows Phone Jump Start

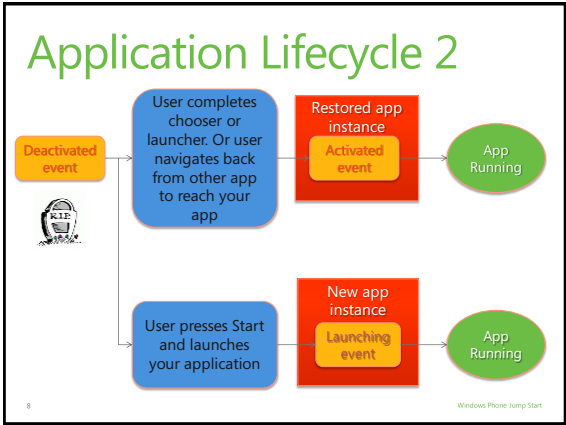
Lifecycle Events

- Application_Launching
 - When the application is launching (eg, from Start)
 - Not fired when the application is reactivated
- Application_Activated
 - When the application is activated (brought to foreground)
 - Not fired when the application is first launched
- Application_Deactivated
 - When the application is deactivated (sent to background)
 - Not fired when the application is closing
- Application_Closing
 - When the application is closing (eg, user hit Back)
 - Not fired when the application is deactivated

6

Windows Phone Jump Start





Launching Event Handler

```
1. // Code to execute when the application is launching (eg, from Start).
   // This code will not execute when the application is reactivated.
   // Application should load persisted data from isolated storage and display
   // the application's initial state
2. void Application_Launching(object sender, LaunchingEventArgs e)
3. {
4.     using (IsolatedStorageFile isf =
       IsolatedStorageFile.GetUserStoreForApplication())
5.     {
6.         if (isf.FileExists("MyPersistenceFile.dat"))
7.         {
8.             // Read persisted data from Storage
9.         }
10.        else
11.        {
12.            // No existing persisted data - create new data objects...
13.        }
14.    }
```

9

Windows Phone Jump Start

Closing Event Handler

```
1. // Code to execute when the application is closing (eg, user hit Back).
   // This code will not execute when the application is deactivated.
   // Application can save persisted data to isolated storage
2. void Application_Closing(object sender, ClosingEventArgs e)
3. {
4.     using (IsolatedStorageFile isf =
       IsolatedStorageFile.GetUserStoreForApplication())
5.     {
6.         //Create a new file and serialize data
7.         using (IsolatedStorageFileStream fs =
           isf.CreateFile("MyPersistenceFile.dat"))
8.         {
9.             XmlSerializer ser = new XmlSerializer(typeof(MyDataObject));
10.            ser.Serialize(fs, mydataobject);
11.        }
12.    }
13. }
```

10

Windows Phone Jump Start

Deactivated Event Handler

```
1. // Code to execute when the application is deactivated (sent to background).
   // This code will not execute when the application is closing.
   // Application should save transient application data to the
   // PhoneApplicationService.State dictionary so that if your application is
   // resumed, you can quickly restore state.
2. void Application_Deactivated(object sender, DeactivatedEventArgs e)
3. {
4.     // Add data that is "work in progress" (ie transient) to Application state
5.     PhoneApplicationService.Current.State.Add("MyUnsavedDataObject",
       myDataObject);
6.
7.     // Optionally, save any data that needs to be persisted - your app may
       // never be reactivated!
8.     ... persist some data ...
9. }
```

11

Windows Phone Jump Start

Activated Event Handler

```
1. // Code to execute when the application is activated (brought to foreground)
   // This code will not execute when the application is first launched.
   // Application should load app state from PhoneApplicationService.State.
   // Present the user with the experience they had before your application was
   // suspended.
2. void Application_Activated(object sender, ActivatedEventArgs e)
3. {
4.     MyDataObject mydataobject = null; //Create new data object variable
5.
6.     //Try to locate previous data in transient state of the application
7.     if (PhoneApplicationService.Current.State.ContainsKey("MyUnsavedDataObject"))
8.     {
9.         //If found, initialize data variable & remove from application's state
10.        mydataobject =
11.        PhoneApplicationService.Current.State["MyUnsavedDataObject"] as MyDataObject;
12.        PhoneApplicationService.Current.State.Remove("MyUnsavedDataObject");
13.    }
14. }
```

12

Windows Phone Jump Start

Application Lifecycle

Demo

13

Windows Phone Jump Start

Page State

- When app is reactivated, should restore the visual state of the page
 - User should see the page exactly as they left it
 - Include things such as scroll position of a **ScrollViewer** control and the contents of **TextBox** controls
- Use the [State](#) property of the [PhoneApplicationPage](#) class to store transient page state
 - Save state in OnNavigatedFrom event handler
 - Restore state in OnNavigatedTo event handler

14

Windows Phone Jump Start

Managing Page Transient State

Demo

15

Windows Phone Jump Start

Lab 1 – Complete After Session

Application Lifetime

- <http://channel9.msdn.com/learn/courses/WP7TrainingKit/WP7Silverlight/Applicati onLifetimeWP7Lab/>
- Complete the lab

Post your questions in the BTL Forum – if you have any questions or need support while doing your labs

Review

- System only runs one app at a time
- System terminates apps that go into the background
- Deactivated event fires when app is tombstoned
- Activated event fires when app is reactivated
- App that is tombstoned may never be reactivated
- Closed event fires when user backs out past first screen

☐ : True ☐ : False

17

Windows Phone Jump Start

Review

- System only runs one app at a time
- System terminates apps that go into the background
- Deactivated event fires when app is tombstoned
- Activated event fires when app is reactivated
- App that is tombstoned may never be reactivated
- Closed event fires when user backs out past first screen

☐ : True ☐ : False

18

Windows Phone Jump Start

Review

System only runs one app at a time

System terminates apps that go into the background

Deactivated event fires when app is tombstoned

Activated event fires when app is reactivated

App that is tombstoned may never be reactivated

Closed event fires when user backs out past first screen

: True : False

19Windows Phone Jump Start

Review

System only runs one app at a time

System terminates apps that go into the background

Deactivated event fires when app is tombstoned

Activated event fires when app is reactivated

App that is tombstoned may never be reactivated

Closed event fires when user backs out past first screen

: True : False

20Windows Phone Jump Start

Review

System only runs one app at a time

System terminates apps that go into the background

Deactivated event fires when app is tombstoned

Activated event fires when app is reactivated

App that is tombstoned may never be reactivated

Closed event fires when user backs out past first screen

: True : False

21Windows Phone Jump Start

Review

System only runs one app at a time

System terminates apps that go into the background

Deactivated event fires when app is tombstoned

Activated event fires when app is reactivated

App that is tombstoned may never be reactivated

Closed event fires when user backs out past first screen

: True : False

22

Windows Phone Jump Start

Review

System only runs one app at a time

System terminates apps that go into the background

Deactivated event fires when app is tombstoned

Activated event fires when app is reactivated

App that is tombstoned may never be reactivated

Closed event fires when user backs out past first screen

: True : False

23

Windows Phone Jump Start



10 Minute Break



Windows
Phone

© 2010 Microsoft Corporation. All rights reserved. Microsoft, Windows, Windows Vista and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries.

The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation.

MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.
