



A lap around Windows Phone codename "Mango"

Louis-Philippe Pinsonneault
MVP Device Application
Development

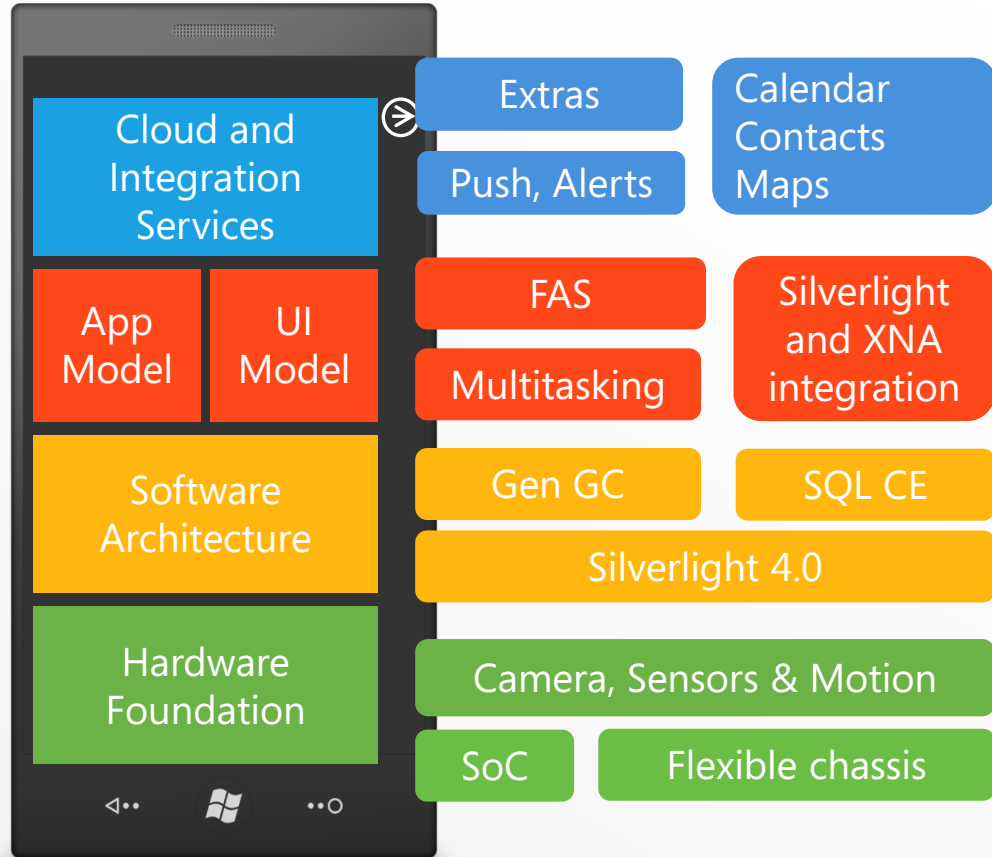


Louis-Philippe Pinsonneault

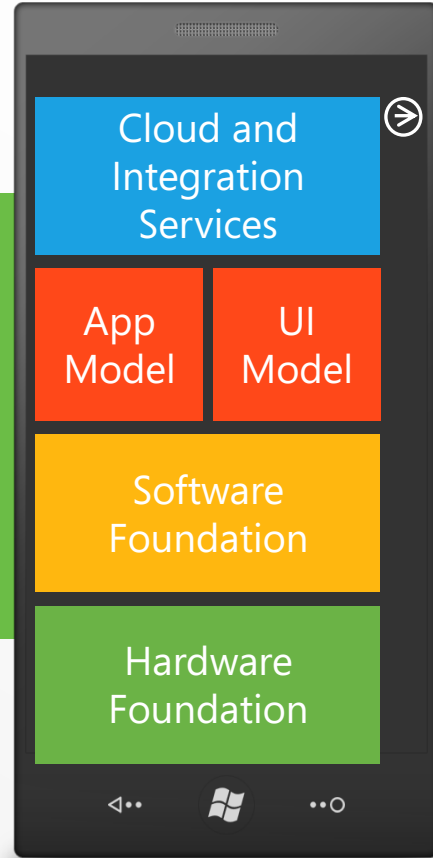


- Développeur .NET et formateur
- +11 ans d'expérience en .NET
- MVP Device Application Development
- Spécialités
 - Silverlight
 - Windows Phone 7
 - WPF
 - ASP.NET
- Courriel: lppinson@runatserver.com
- Blog: <http://blogs.runatserver.com/lppinson>
- Twitter: [@lpinsonneault](https://twitter.com/lpinsonneault)

Agenda: Windows Phone codename "Mango"



Hardware Foundation



Hardware Foundation Updates

800

480

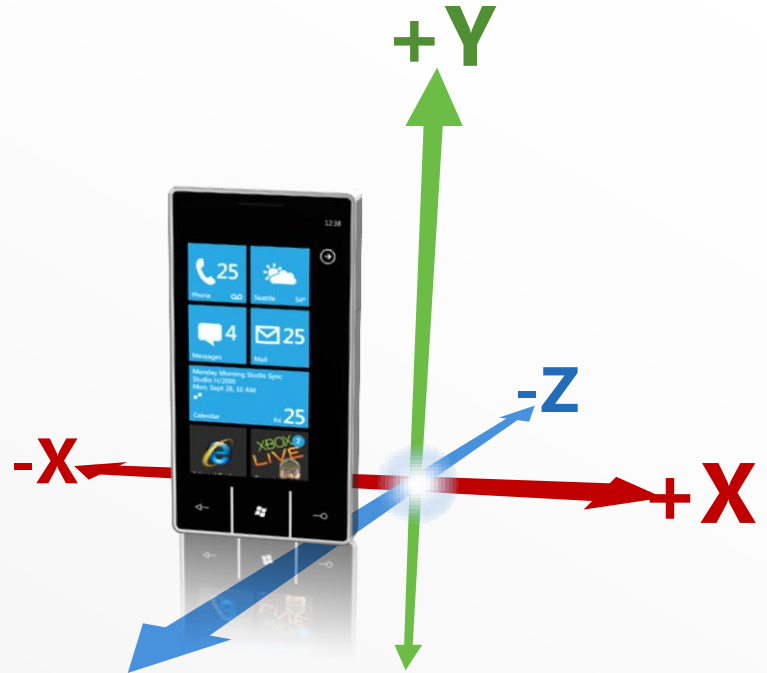
- Capacitive touch**
4 or more contact points
- Sensors** **Motion Sensor**
A-GPS, Accelerometer, **Compass**, Light, Proximity, **Gyro**
- Camera**
5 mega pixels or more
- Multimedia**
Common detailed specs, Codec acceleration
- Memory**
256MB RAM or more, 8GB Flash or more
- GPU**
DirectX 9 acceleration
- CPU**
Qualcomm MSM8x55 800Mhz or higher **MSM7x30**

Hardware buttons | Back, Start, Search

**Improved
capability
detection APIs**

Accelerometer

- Measures resultant acceleration (force) on device
- Pros:
 - Available on all devices
- Cons:
 - Difficult to tell apart small orientation changes from small device motions



Accelerometer demo

Camera

- Access to live camera stream
 - PhotoCamera
 - Silverlight 4 Webcam
- Display in your app
 - Video Brush



When to use each approach

PhotoCamera

- Take High Quality Photos
- Handle Hardware Button
- Handle Flash mode and Focus
- Access Samples (Pull Model)

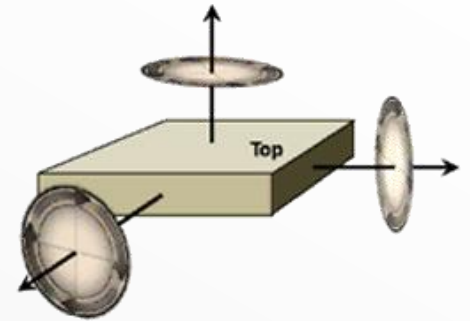
Webcam

- Record Video
- Record Audio
- Share code with desktop
- Access Samples (Push Model)

Camera demo

Gyroscope

- Measures rotational velocity on 3 axis
 - Optional on Mango phones
 - Not present in pre-Mango WP7 phones



Gyroscope API

```
private void start_button_click(object sender, RoutedEventArgs e)
{
    if (Gyroscope.IsSupported)
    {
        gyroSensor = new Gyroscope();
        gyroSensor.TimeBetweenUpdates = TimeSpan.FromMilliseconds(Double.Parse(textTimeInput.Text,
                                                                    System.Globalization.NumberStyles.Number));

        gyroSensor.CurrentValueChanged += new EventHandler<SensorReadingEventArgs<GyroscopeReading>>(gyroSensor_CurrentValueChanged);
        lastUpdate = DateTimeOffset.Now;
        gyroSensor.Start();
        textTimeInput.Text = gyroSensor.TimeBetweenUpdates.TotalMilliseconds.ToString();
    }
}

void gyroSensor_CurrentValueChanged(object sender, SensorReadingEventArgs<GyroscopeReading> e)
{
    TimeSpan t = (lastUpdate - e.SensorReading.Timestamp);
    lastUpdate = e.SensorReading.Timestamp;
    currentPosition += e.SensorReading.RotationRate * (float)t.TotalSeconds;

    textRawData.Text = String.Format("X: {0:N3}, Y: {1:N3}, Z: {2:N3}",
                                     e.SensorReading.RotationRate.X,
                                     e.SensorReading.RotationRate.Y,
                                     e.SensorReading.RotationRate.Z);
    textIntegration.Text = String.Format("X: {0:N3}, Y: {1:N3}, Z: {2:N3}",
                                         currentPosition.X,
                                         currentPosition.Y,
                                         currentPosition.Z);
}
```

Compass (aka Magnetometer)

- Gives 3D heading of Earth's magnetic and Geographic North
 - Subject to external electromagnetic influences
 - Requires user calibration over time
 - Great inaccuracies in orientation, up to 20 degrees
 - Significant lag
- Availability:
 - Optional on "Mango" phones
 - Included in some pre-Mango WP7 phones

Compass API

```
protected override void OnNavigatedTo(NavigationEventArgs e){
    if (Compass.IsSupported) {
        compass = new Compass();
        compass.CurrentValueChanged += compass_CurrentValueChanged;
        compass.Start()
    }
}
private void compass_CurrentValueChanged(object sender,
                                         SensorReadingEventArgs<CompassReading> e) {
    Deployment.Current.Dispatcher.BeginInvoke(() =>
    {
        CompassRotation.Angle = -e.SensorReading.TrueHeading;
        Heading.Text = e.SensorReading.TrueHeading.ToString("0°");
    });
}
```

Compass demo

Motion Sensor

- Virtual sensor, combines gyro + compass + accelerometer
- Motion Sensor vs. gyro or compass or accelerometer
 - More accurate
 - Faster response times
 - Comparatively low drift
 - Can disambiguate motion types
- Has fall-back if gyro is not available

Always prefer Motion Sensor when available

Motion API

```
if (Motion.IsSupported) {  
    _sensor = new Motion();  
    _sensor.CurrentValueChanged += new  
        EventHandler<SensorReadingEventArgs<MotionReading>>  
(sensor_CurrentValueChanged);  
    _sensor.Start();  
}
```

```
void _sensor_CurrentValueChanged(object sender, SensorReadingEventArgs<MotionReading> e)  
{  
    Simple3DVector rawAcceleration = new Simple3DVector(  
        e.SensorReading.Gravity.Acceleration.X,  
        e.SensorReading.Gravity.Acceleration.Y,  
        e.SensorReading.Gravity.Acceleration.Z); ...  
}
```

Motion Sensor Adapts to Devices

Accelerometer	Compass	Gyro	Motion
Yes	Yes	Yes	Full
Yes	Yes	No	Degraded
Yes	No	Yes	Unsupported
Yes	No	No	Unsupported

- Degraded modes have lower quality approximations
- When `Motion.IsSupported` is false, apps should use accelerometer or other input and control mechanisms

Sensor Calibration

- Calibration Event is fired when calibration is needed
 - Both Compass and Motion sensors need user calibration
- Apps should handle it
 - Provide UI asking user to move device through a full range of orientations
 - Not handling will cause inaccurate readings
 - We are considering providing copy & paste solution

Mango Device

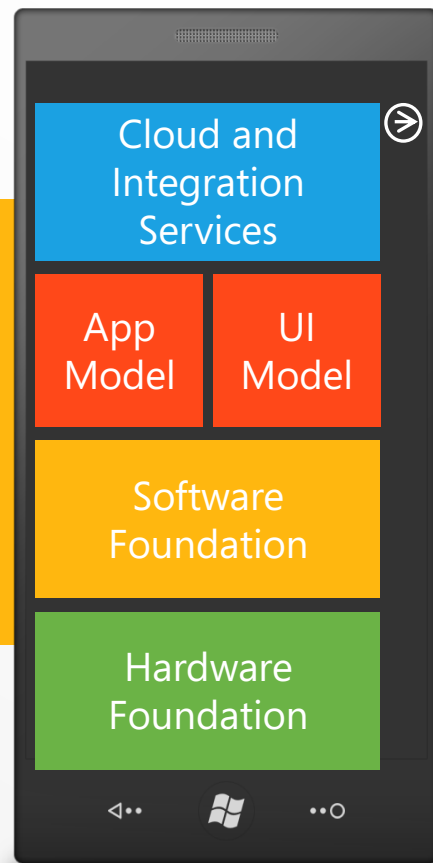
- Samsung Focus S
- Samsung Focus Flash
- Samsung Omnia W
- HTC Titan
- HTC Radar
- Nokia 800
- Nokia 710



Mango Device

- These device are not yet available in Canada
- Some should be available 1st quarter 2012

Software Foundation



Run-time improvements

Silverlight 4

- Implicit styles
- RichTextBox
- ViewBox
- More touch events (tap, double tap)

Features

- Sockets
- Clipboard
- IME
- WebBrowser (IE9)
- VideoBrush

Performance

- Generational GC
- Input thread
- Working set
- Profiler

Networking

- Sockets
 - TCP
 - UDP unicast, Multicast (on Wi-Fi)
- Connection Manager Control
 - Overrides and sets preferences (e.g. Wi-Fi or cellular only)
- HTTP
 - Full header access
 - WebClient returns in originating thread

Sockets demo

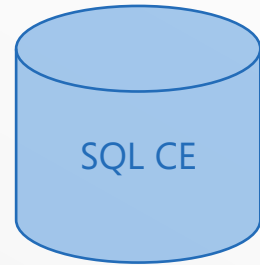
Silverlight and XNA Shared Graphics

- XNA inside Silverlight App
- Integration at Page Level
 - XNA takes over rendering
- Integration at Element level
 - Silverlight elements in XNA pipeline via UIElementRenderer
- Shared input

Silverlight + XNA demo

Local database

- SQL Compact Edition
 - Use object model for CRUD
 - LINQ to SQL to query, filter, sort
- Application level access
 - Sandboxed from other apps
 - Uses IsolatedStorage
 - Access for background agents
- DatabaseSchemaUpdater APIs for upgrades



Database APIs: DataContext and attributes

```
// Define the data context.
public partial class WineDataContext : DataContext
{
    public Table<Wine> Wines;
    public Table<Vineyard> Vineyards;
    public WineDataContext(string connection) : base(connection) { }
}

// Define the tables in the database
[Table]
public class Wine
{
    [Column(IsPrimaryKey=true)]
    public string WineID { get; set; }
    [Column]
    public string Name { get; set; }
    .....
}

// Create the database form data context, using a connection string
DataContext db = new WineDataContext("isostore:/wineDB.sdf");
if (!db.DatabaseExists()) db.CreateDatabase();
```

Queries: Examples

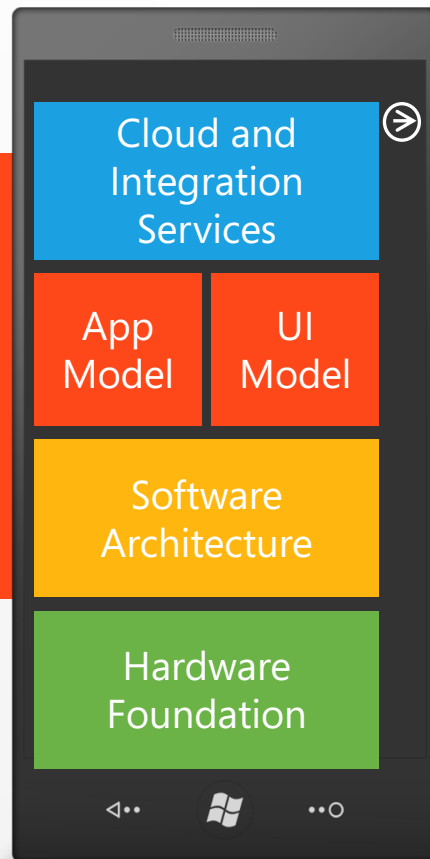
```
// Find all wines currently at home, ordered by date acquired
var q = from w in db.Wines
        where w.Varietal.Name == "Shiraz" && w.IsAtHome == true
        orderby w.DateAcquired
        select w;
```

```
Wine newWine = new Wine
{
    WineID = "1768", Name = "Windows Phone Syrah",
    Description = "Bold and spicy"
};
```

```
db.Wines.InsertOnSubmit(newWine);
db.SubmitChanges();
```

Local database demo

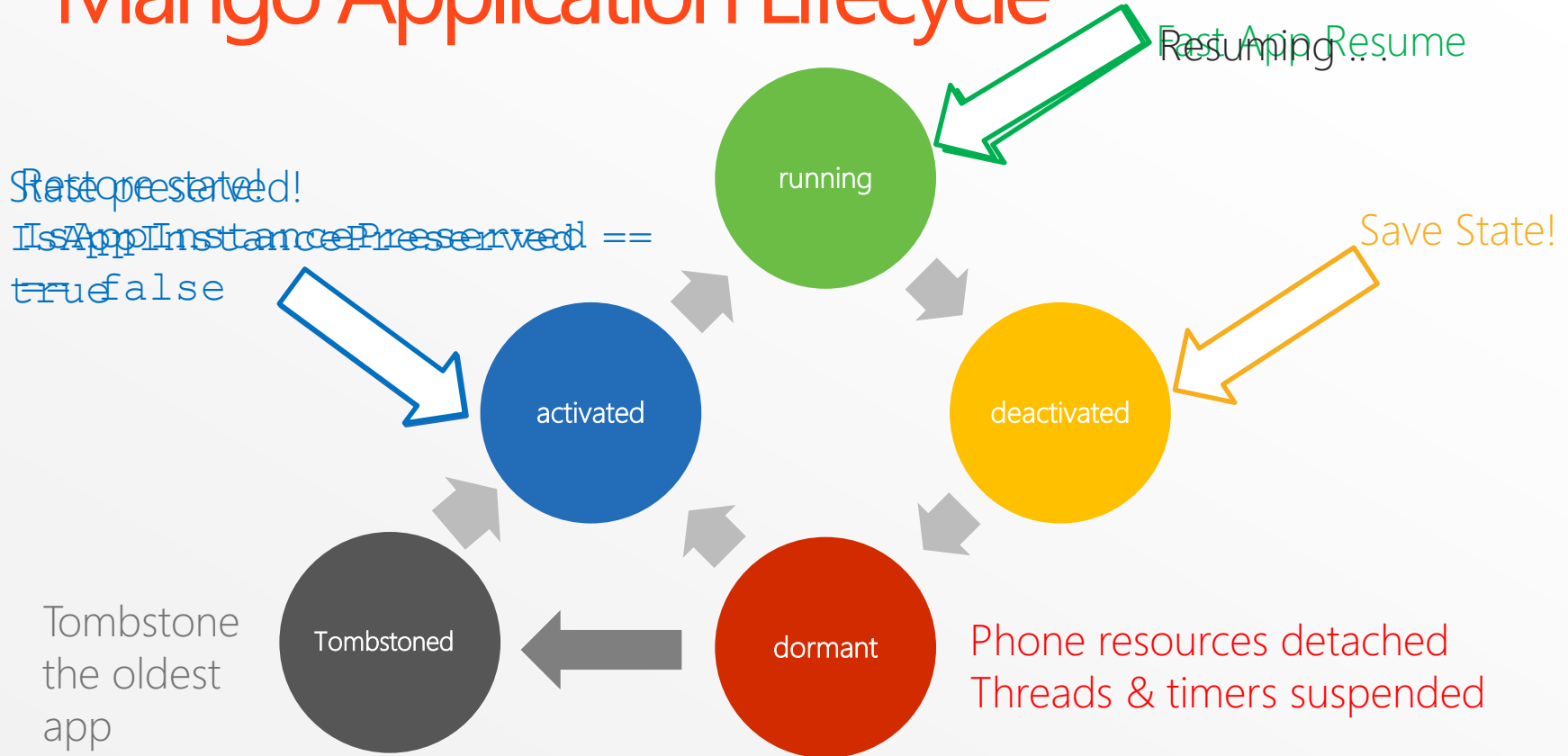
Application Model



Fast Application Resume

- Immediate Resume of recently used applications
 - Apps stay in memory after deactivation
- New “task switcher”
 - Long-press back button
- While dormant
 - Apps are not getting CPU cycles
 - Resources are detached
- You must recompile and resubmit targeting Mango

Mango Application Lifecycle



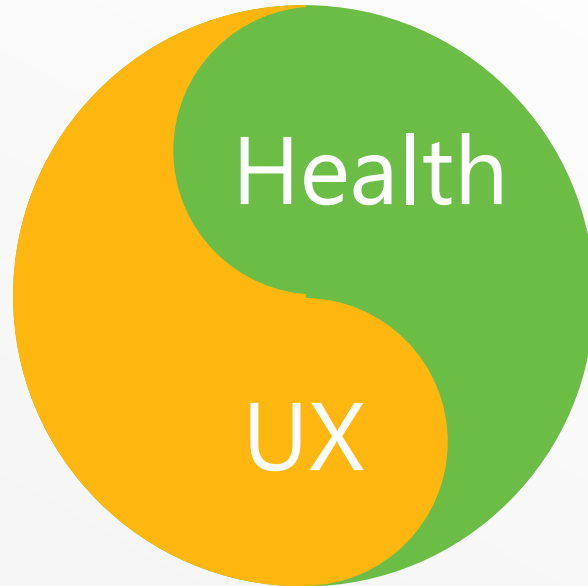
Fast App Resume demo

Multi-tasking design principles

Delightful and
Responsive UX

Never Regret
App Install

Integrated Feel



Battery
Friendly

Network
Conscience

Hardened
Services

Multi-tasking Options

- Background Transfer Service
- Background Audio
- Background Agents
 - Periodic
 - On Idle
- Alarms and Reminders

Background Audio

- Playback
 - App provides URL or stream to Zune
 - Audio continues to play even if app is closed
 - App is notified of file or buffer near completion
- Phone Integration
 - Music & Video Hub
 - Universal Volume Control (UVC), launch app, controls, contextual info
 - Contextual launch – Start menu, UVC, Music & Video Hub
- App Integration
 - App can retrieve playback status, progress, & metadata
 - Playback notification registration

Background Audio App Types

- URL PlayList
 - Provide URL to play
 - Pause, resume, stop, skip-forward, skip-backward
 - Stream Source
 - Provide audio buffers
 - Custom decryption, decompression
- Requires app to run some code in background

Background Agents

- Agents
 - Periodic
 - Resource-Intensive
 - An app may have up to one of each
- Initialized in foreground, run in background
 - Persisted across reboots
- User control through CPL
 - System maximum of 18 periodic agent
- Agent runs for up to 14 days (can be renewed)

Generic Agent Types

Periodic Agents

- Occurrence
 - Every 30 min
- Duration
 - ~25 seconds
- Constraints
 - ≤ 6 MB Memory
 - $\leq 10\%$ CPU

Resource- Intensive Agents

- Occurrence
 - External power, non-cell network
- Duration
 - 10 minutes
- Constraints
 - ≤ 6 MB Memory

Background Agent Functionality

Allowed

- Tiles
- Toast
- Location
- Network
- R/W ISO store
- Sockets
- Most framework APIs

Restricted

- Display UI
- XNA libraries
- Microphone and Camera
- Sensors
- Play audio
(may only use background audio APIs)

Agent
demo

Notifications

- Time-based, on-phone notifications
- Supports Alerts & Reminders
- Persist across reboots
- Adheres to user settings
- Consistent with phone UX

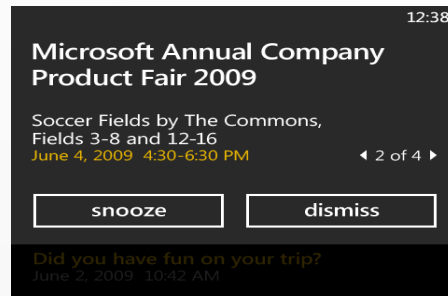
Alarms vs Reminders?

Alarms



- Modal
- Snooze and Dismiss
- Sound customization
- No app invocation
- No stacking

Reminders



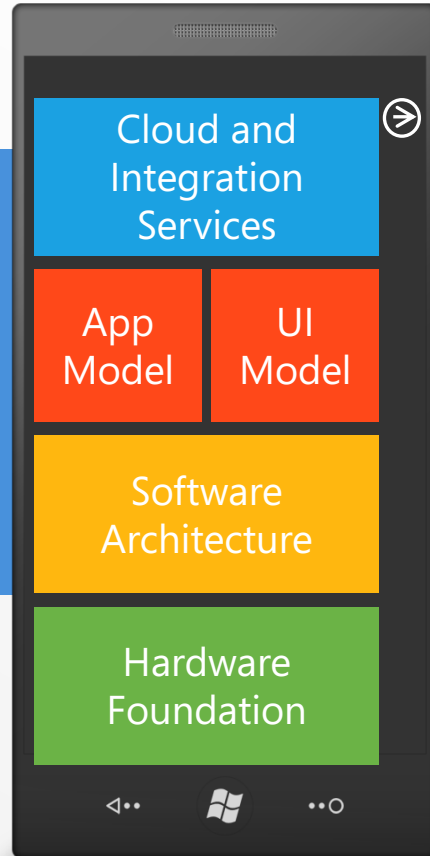
- Rich information
- Integrates with other reminders
- Snooze and Dismiss
- Launch app
- Follows the phones global settings

Alarms & reminders demo

Background Transfer Service

- Start transfer in foreground, complete in background, even if app is closed
- Queue persists across reboots
 - Queue size limit = 5
 - Queue APIs (Add, Remove, Query status)
- Single service for many apps, FIFO
- Download ~20 MB (> over Wi-Fi)
- Upload Size ~4 MB (limit to come)
- Transfers to Isolated Storage

Integration Services



Live Tile improvements

- Local Tile APIs
 - Full control of ALL properties
- Multiple tiles per app
 - Create, Update/Delete/Query
 - Launches direct to Uri



Application Tile

Launches main app experience

Secondary Tile

Launches world news page

Secondary Tile

Launches local news page

Live Tiles – Local Tile API Continued...

- Back of tile updates
 - Full control of all properties when your app is in the foreground or background
 - Content, Title, Background



- Flips from front to back at random interval
- Smart logic to make flips asynchronous

Live tiles demo

Push Notifications (Core) Enhancements

Reliability

- New TDET mechanism for broader network compatibility
- Lowered polling interval for non-persistent connection friendly networks

Efficiency

- TLS resume for sessions within 8 hours
- Hints for improved radio dormancy
- Concurrent tile downloads for less radio uptime

Performance

- Faster state machine for faster client service
- Smarter queue logic for less redundancy

30 Endpoints Limit !!!

Push Notifications – New Features!

- MultiTile/Back of Tile Support
 - Can update all tiles belonging to your application
 - No API Change! – BindToShellTile now binds you to all tiles
 - Send Tile ID to service and use new attribute to direct update
 - 3 new elements for back properties:
BackBackgroundImage, BackContent, BackTitle
- Deep Toast
 - Take users directly to an application experience
 - Uses standard SL navigation (OnNavigatedTo)
 - No API change! – BindToShellToast still all you need.
 - New element to send query parameters with a toast: Param

Extras

- Integration point between Bing Search and 3rd party apps
- User launches 3rd party from Bing Search – search parameter is passed to the app
- Four item types:
 - Movies
 - Places
 - Events
 - Products

New Choosers and Launchers

- SaveRingtoneTask
- AddressChooseTask
- BingMapsTask
- BingMapsDirectionsTask
- GameInviteTask
- Updates:
 - EmailAddressChooserTask
 - PhoneNumberChooserTask

Contacts

- Read-only querying of contacts
- Third party social data cannot be shared
- Requires
ID_CAP_CONTACTS

Calendar

- Read-only querying of calendar appointments
- Returns a snapshot (not live data)
 - You must refresh manually

- Requires ID_CAP_APPOINTMENTS

Appointments API

```
Appointments appointments = new Appointments();
appointments.SearchCompleted += new
EventHandler<AppointmentsSearchEventArgs>((sender, e) =>
{
    ... = e.Results;
});
```

```
// get next appointment (up to 1 week away)
appointments.SearchAsync(DateTime.Now,
                        DateTime.Now + TimeSpan.FromDays(7),
                        1, null);
```

Start date and time

end date and time

state

Maximum items to return

What are Search Extras?

- Added functionality 3rd party apps provide for Bing items
- Four item types:
 - Movies
 - Places
 - Events
 - Products

Three Easy Steps to Implement Search Extras

1. Update your app's Manifest

- Use the Extensions element
- One child Extension element for each category your app supports
 - Your app will appear in those items!
 - This is a great way to drive downloads if your app isn't yet installed

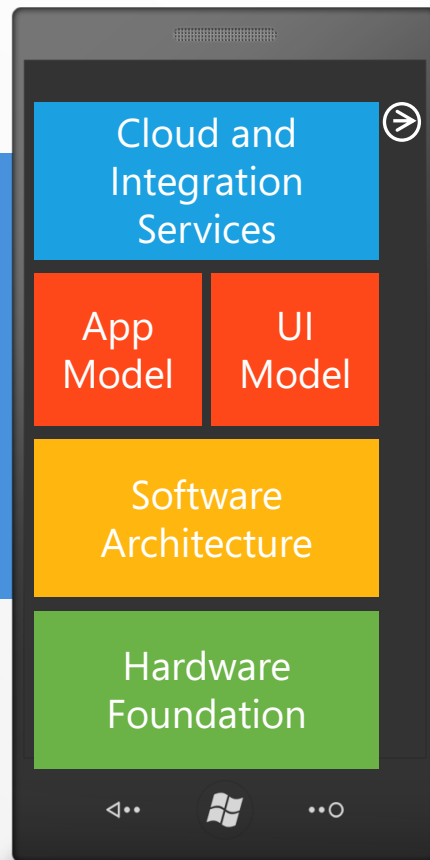
2. Add an Extras.XML file to your XAP

- Specify captions for each Bing category

3. Accept Context to automatically open the item in your app

- Create a SearchExtras page that accepts parameters.
- Search for the item using the parameters passed to your app.

Marketplace Test Kit



Testing for Marketplace?

- Marketplace Test Kit
- Help reproduce test case used by the certification team
 - Application Details
 - Automated Tests
 - Monitored Tests
 - Manual Tests

Application Details

Marketplace Test Kit

Application Details

Automated Tests

Monitored Tests


Manual Tests

Please provide the additional resources to run the tests provided by the MarketPlace Preparation Testkit. These test cases help prepare your application for the Marketplace submission process, but passing them does not guarantee that the application will be accepted in the Marketplace.

Application package:


D:\Workspace\Runatserver\Windows Phone\TechDays2010\TechDaysPhoneApp\Bin\Release\TechDaysPhoneApp.xap

Large application tile:




Browse...

Small application tile:




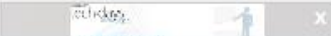

Browse...

Marketplace tile:



Browse...

Application screenshots:



Automated Tests

Marketplace Test Kit

Application Details

Automated Tests

Monitored Tests

Manual Tests

Click the Run Tests button below to run the automated test cases.

Run Tests

Passed: 4 Failed: 0

Result	Test Name	Test Description	Result Details
✓ Passed	XAP Package Requirements	Validation of XAP file size and content files	
✓ Passed	Capability Validation	Validation of application capabilities	[INFORMATION] : Capabilities used by application : ID_CAP_NETWORKING ID_CAP_LOCATION ID_CAP_IDENTITY_DEVICE ID_CAP_IDENTITY_USER ID_CAP_WEBBROWSERCOMPONENT
✓ Passed	Iconography	Validation of Application Icons	
✓ Passed	Screenshots	Validation of Screenshots	

Monitored Tests

The screenshot shows the 'Marketplace Test Kit' application window. On the left is a navigation pane with four items: 'Application Details', 'Automated Tests', 'Monitored Tests' (which is selected), and 'Manual Tests'. The main content area has a title bar 'Marketplace Test Kit' with standard window controls. Below the title bar, there are two buttons: 'Start Application' (highlighted with a blue dashed border) and 'Close Application'. The main content area contains the following text:

Monitored test cases: Use to analyze application performance and alignment with pre-certification requirements during usage.
Please follow these steps:

- Start application by clicking 'Start Application'.
- Navigate forward and back in the application and exercise context menus and dialog boxes.
- Exercise the application functionality. If the application requires network access, make sure to test in both networked and non-networked conditions.
- Use the back button to close the application.

Below the text is a table with a yellow header row that reads 'Test cases have not been run yet.' The table has four columns: 'Result', 'Test Name', 'Test Description', and 'Result Details'. There are four rows of test cases, all with a 'Pending' status and an information icon.

Result	Test Name	Test Description	Result Details
Pending	Launch time	Validation of application launch time.	
Pending	Peak memory consumption	Validation of application peak memory consumption	
Pending	Application closure	Validation of all exceptions being handled and application not closing unexpectedly.	
Pending	Use of Back Button	Validation of proper behavior when pressing the Back button.	

Manual Tests

Marketplace Test Kit

Application Details




Automated Tests

Monitored Tests

Manual Tests

Below are the list of manual testcases. Follow the instructions below to execute the testcases before submitting the app to marketplace.

Passed : 0 Failed : 0 Pending : 50

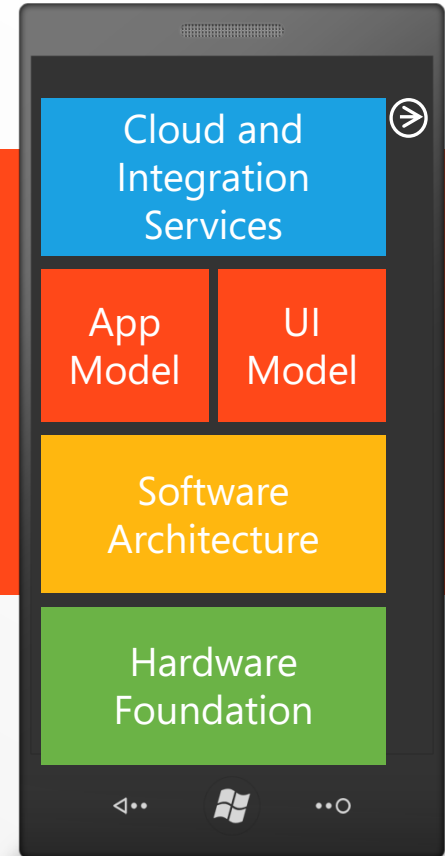
Result	Test Name	Test Description
 Pending	Applicable Application Tile Images	<ul style="list-style-type: none">View the Application list.Verify that the small mobile app tile image is representative of the application.From the Application list, tap and hold the small mobile app tile of your application and select 'pin to start'.Verify that the large mobile tile image on the Start screen is representative of the application. More info...
 Pending	Multiple Devices Support	<ul style="list-style-type: none">Install your application on two or more Windows Phone devices.Verify that the application can install and uninstall without error.After testing the above, ensure your application is installed, and launch it.Comprehensively test application functionality and features to verify that there are no device-specific issues.Verify that the application does not cause the device to stop responding or crash. More info...
 Pending	Application Closure	<ul style="list-style-type: none">Launch your application.Navigate throughout the application, and then close the application.

Call to Action

- Download the tools at <http://create.msdn.com>
- 7.5 allows you to build deeply integrated phone experiences – build them into your apps!
- Multitasking opens up completely new experiences
- Integration points are a key way for your app to shine

Ressources

<http://bit.ly/bloglpp>





© 2011 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.