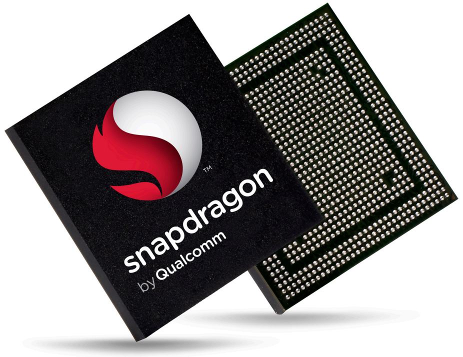
Как работать с внешними устройствами и оборудованием смартфона на Windows Phone

Michael
Samarin
@MichaelSamarin

Today's Topics

Camera and Image Processing



SensorCore SDK

Bluetooth RFComm

Windows Phone 8.x APIs

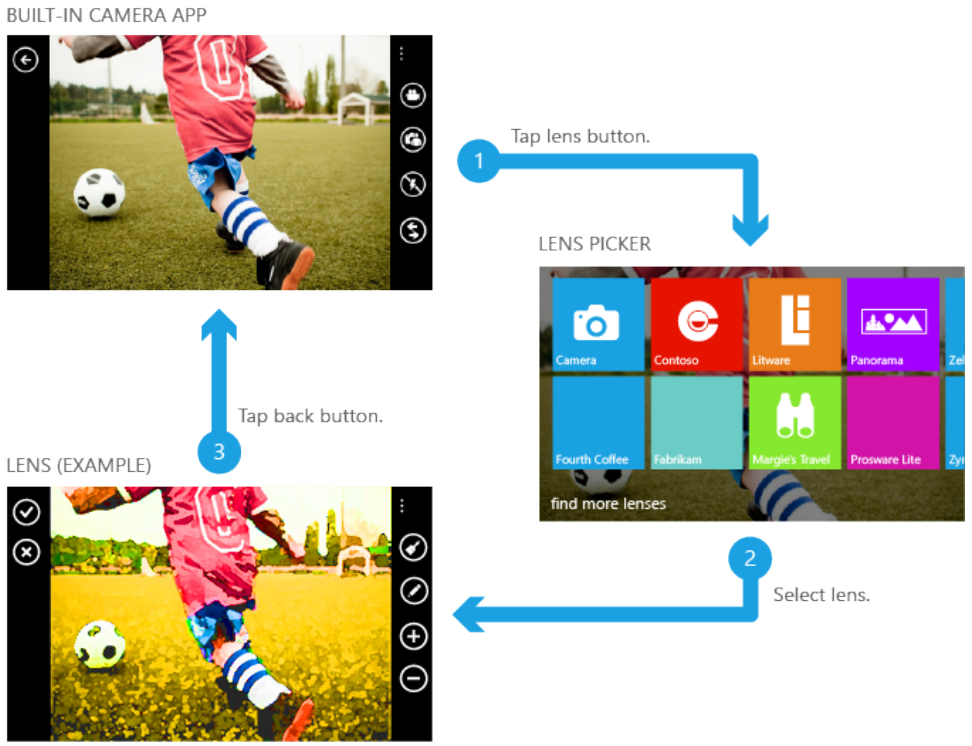
Windows Phone Silverlight 8.0
Windows Phone Silverlight 8.1
Windows Phone 8.1 (Universal)

futurice

Camera

And Image Processing

Lenses:


BUILT-IN CAMERA APP
Tap lens button.
1
LENS PICKER
Camera
Contoso
Litware
Panorama
Fourth Coffee
Fabrikam
Margie's Travel
Proseware Lite
find more lenses
2
Select lens.
LENS (EXAMPLE)
3
Tap back button.

Windows Phone 8.x Silverlight Camera Classes

- PhotoCaptureDevice
- AudioVideoCaptureDevice

Building Custom Camera Apps

- Making custom viewfinder
- Controlling Camera Parameters
- Accessing Hardware Shutter Button
- Accessing Live Preview Buffer

futurice LEAN SERVICE CREATION

System.Windows.Media.VideoBrush

- setSource(<camera>)

Microsoft.Devices.CameraButtons

- ShutterKeyHalfPressed
- ShutterKeyPressed
- ShutterKeyReleased

futurice LEAN SERVICE CREATION

Windows.Phone.Media.Capture.AudioVideoCaptureDevice

Properties

- AvailableSensorLocations
- CaptureResolution
- FocusRegion
- PreviewResolution
- SensorLocation
- SensorRotationInDegrees

Events

- PreviewFrameAvailable
- VendorSpecificDataAvailable

Methods

- Close
- CreateCaptureSequence
- FocusAsync
- GetAvailableCaptureResolutions
- GetAvailablePreviewResolutions
- GetPreviewBufferArgb
- GetPreviewBufferY
- GetPreviewBufferYCbCr
- GetProperty
- GetSupportedPropertyRange
- GetSupportedPropertyValues
- IsFocusRegionSupported
- IsFocusSupported
- OpenAsync
- PrepareCaptureSequenceAsync
- ResetFocusAsync
- SetCaptureResolutionAsync
- SetPreviewResolutionAsync
- SetProperty

futurice LEAN SERVICE CREATION

AudioVideoCaptureDevice

Access to live preview buffer

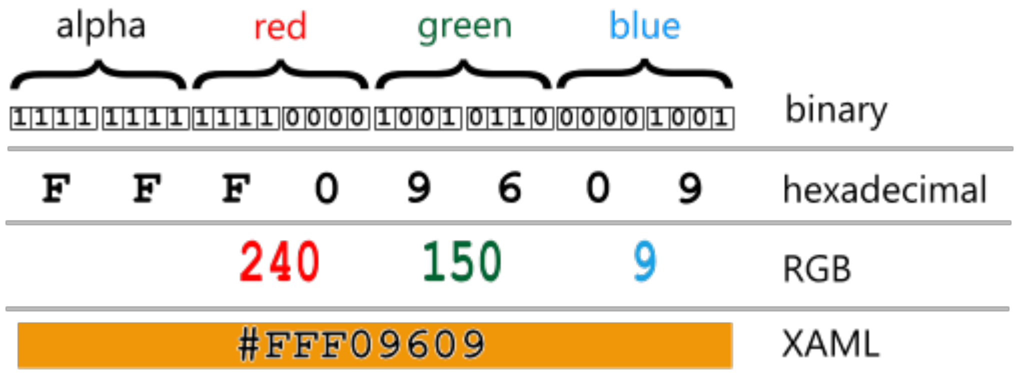
Properties

- YCbCrPixelLayout

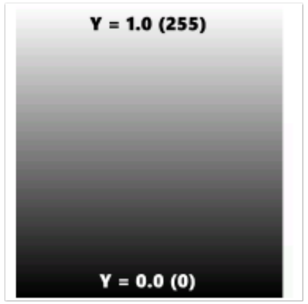
Methods

- GetPreviewBufferArgb
- GetPreviewBufferY
- GetPreviewBufferYCbCr

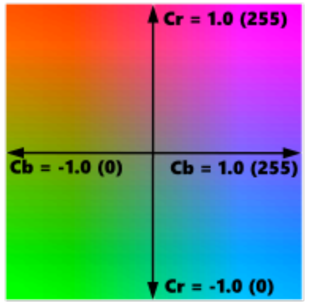
ARGB


alpha
red
green
blue
1111 1111 1111 0000 1001 0110 0000 1001
binary
F F F 0 9 6 0 9
hexadecimal
240 150 9
RGB
#FFF09609
XAML

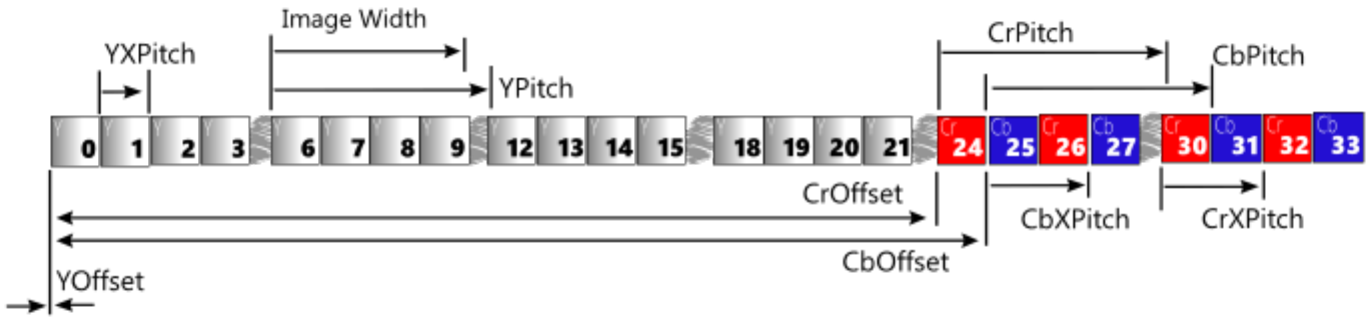
YCbCr



Luminance



Chrominance (Y=0.5)



Hands-On

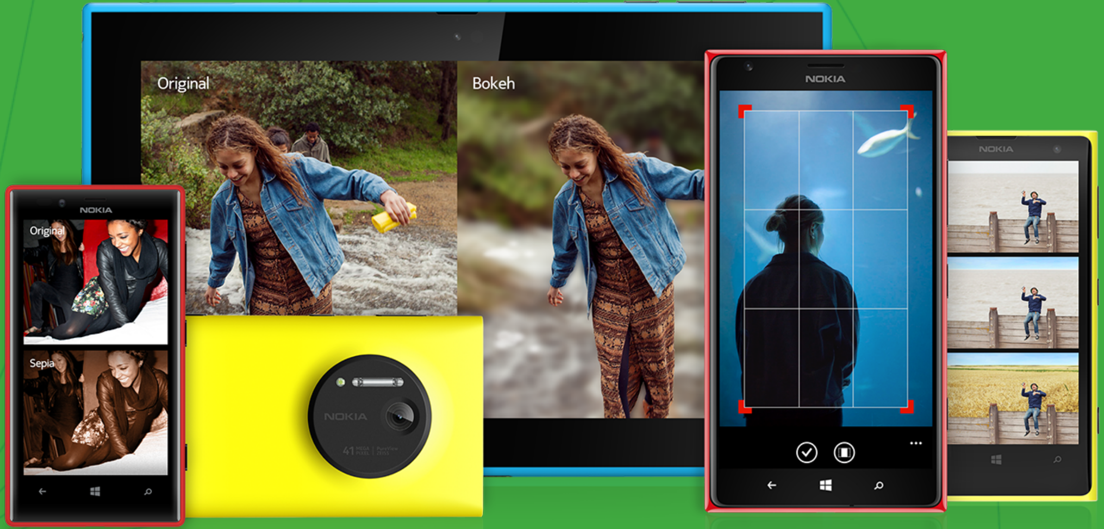
Camera APIs

futurice LEAN SERVICE CREATION

futurice

Imaging SDK

And Image Processing



What is Imaging SDK

- Native library, available for 3rd party developers and supports all Windows Phone 8 / 8.1, Windows 8.1/RT devices
- Includes more than 50 ready to use image processing filters and effects, with various adjustment parameters
- Supports creation of the totally custom filters and effects
- Accessible from C#, VB and C++ Projects
- Library doesn't require special knowledge of image processing algorithms or techniques

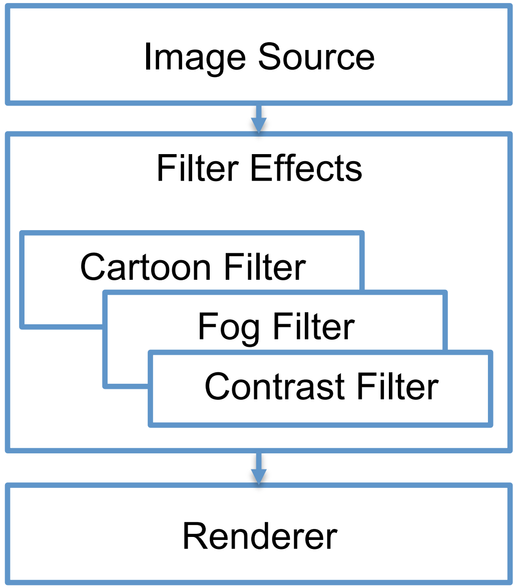
futurice LEAN SERVICE CREATION

What is Imaging SDK

- Filters and effects can be used sequentially, making possible virtually unlimited amount of combinations for imaging special effects
- Parameters of the filters can be changed without rebuilding rendering pipeline
- Directly supports various source types: bitmaps, streams, files and camera viewfinder
- Partial JPEG decoding - using RAJPEG technology, access image data without decoding a whole JPEG image for a fast previews, application of effects, rotation, and cropping of high resolution images.

General Architecture Overview

- Library contains three architectural building blocks:
  - Image sources (such as bitmaps, streams, files)
  - Effects (such as 50+ various filters, including custom)
  - Renderers (outputs bitmaps or files)
- Combining these building blocks, developer creates rendering pipeline
- Once pipeline is created, it is possible to change filter parameters, or their sequence.



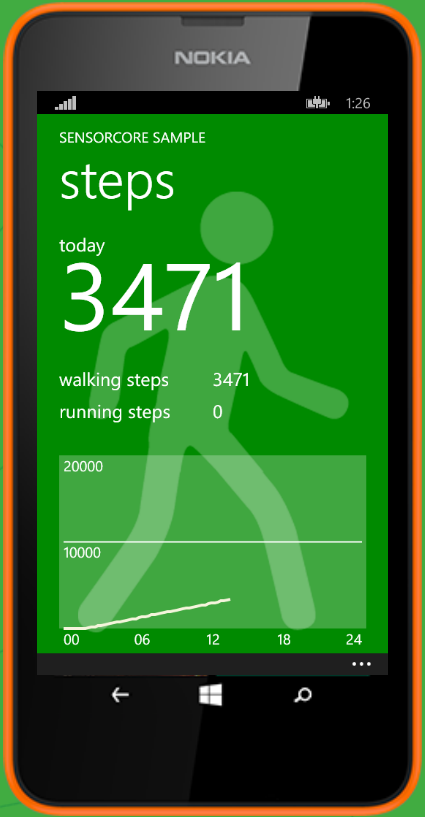
Hands-On

Camera APIs and Imaging SDK

futurice

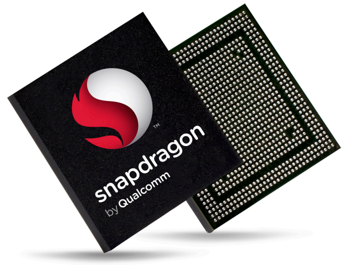
SensorCore SDK

Accessing Snapdragon’s Sensors



New Lumia Devices with Sensor Core

Lumia 630 / 635



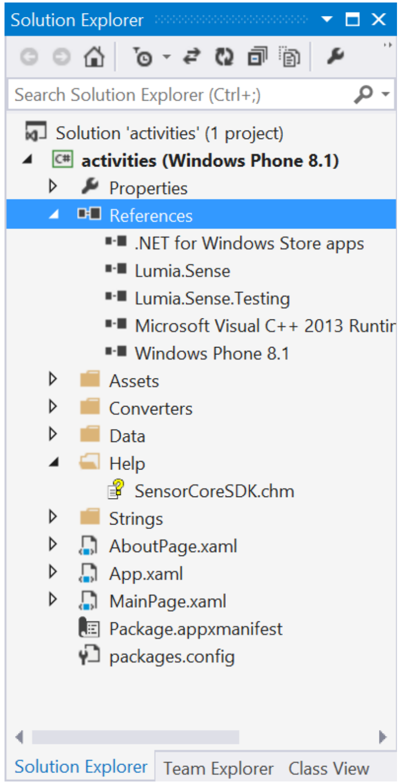
Lumia 930

Lumia SensorCore SDK for 3rd Party Developers

Windows Phone 8.1 library

Available for any 3rd party developers

Supported architectures: ARM on device, x86 on Emulator

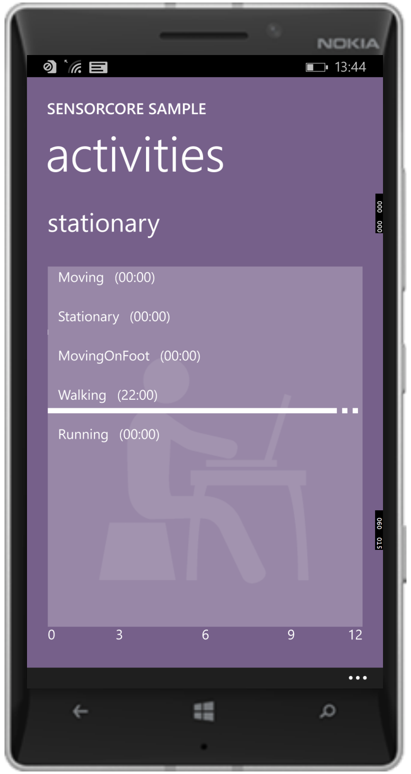


Solution Explorer

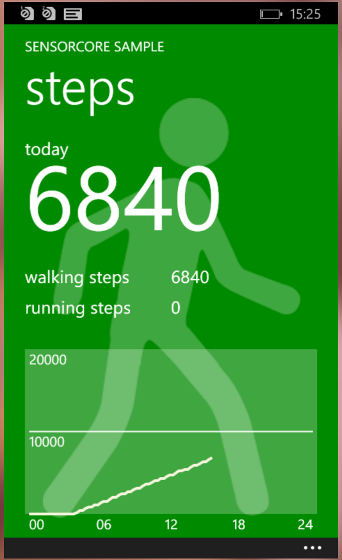
Search Solution Explorer (Ctrl+;)

- Solution 'activities' (1 project)
  - **activities (Windows Phone 8.1)**
    - Properties
    - References
      - .NET for Windows Store apps
      - Lumia.Sense
      - Lumia.Sense.Testing
      - Microsoft Visual C++ 2013 Runtir
      - Windows Phone 8.1
    - Assets
    - Converters
    - Data
    - Help
      - SensorCoreSDK.chm
    - Strings
    - AboutPage.xaml
    - App.xaml
    - MainPage.xaml
    - Package.appxmanifest
    - packages.config

Solution Explorer | Team Explorer | Class View



Activity and Motion Tracking



15:25

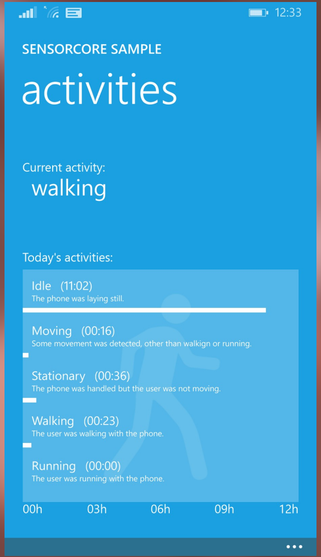
SENSORCORE SAMPLE

steps

today

6840

walking steps	6840
running steps	0



12:33

SENSORCORE SAMPLE

activities

Current activity:
walking

Today's activities:

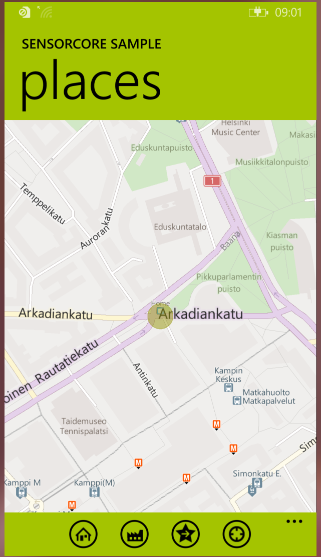
Idle (11:02)
The phone was laying still.

Moving (00:16)
Some movement was detected, other than walkign or running.

Stationary (00:36)
The phone was handled but the user was not moving.

Walking (00:23)
The user was walking with the phone.

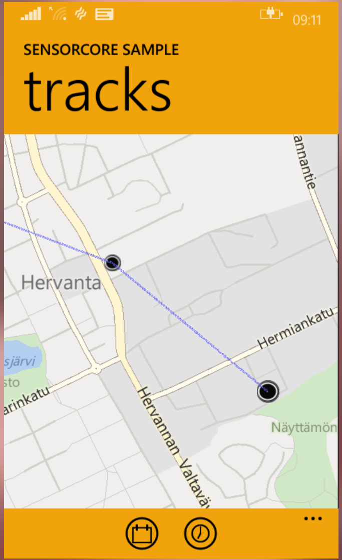
Running (00:00)
The user was running with the phone.



09:01

SENSORCORE SAMPLE

places



09:11

SENSORCORE SAMPLE

tracks

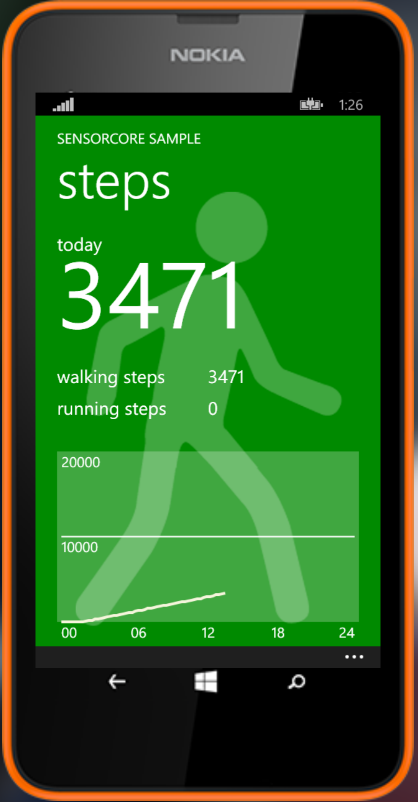
Activity, Motion and Privacy

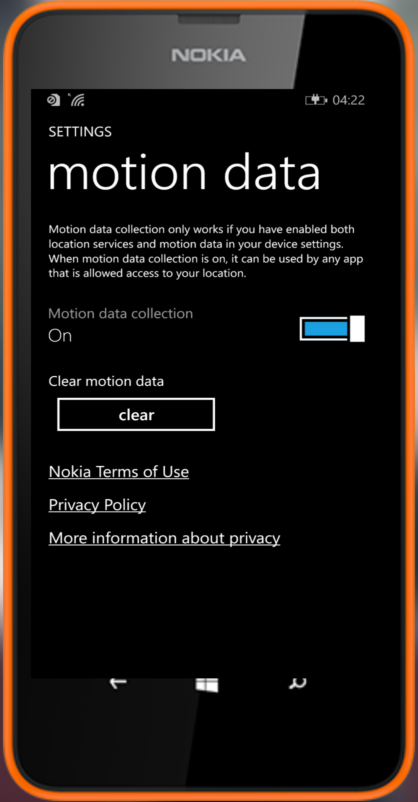
Running constantly in the background

Collecting and preserving data for up to 10 days

Tracking physical activity and motion

Option for disabling tracking and clearing data





motion data

Motion data collection only works if you have enabled both location services and motion data in your device settings. When motion data collection is on, it can be used by any app that is allowed access to your location.

Motion data collection
On

Clear motion data

Nokia Terms of Use
Privacy Policy
More information about privacy

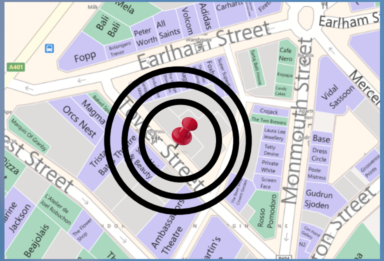
Included APIs:

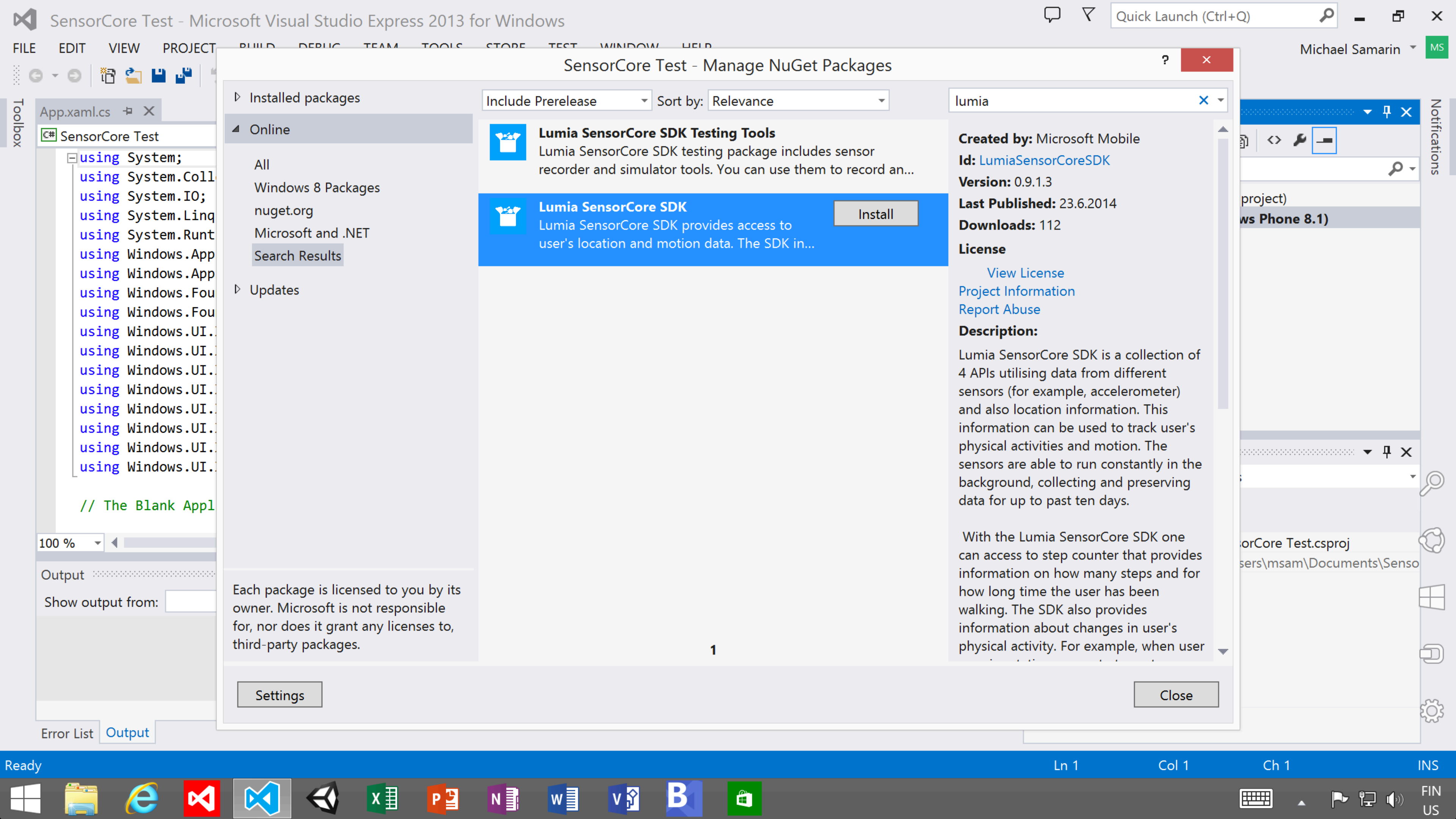
Step Counter API

Activity Monitor API

Place Monitor API

Track Point Monitor API




SensorCore Test - Microsoft Visual Studio Express 2013 for Windows

SensorCore Test - Manage NuGet Packages

Include Prerelease | Sort by: Relevance | lumia

Lumia SensorCore SDK Testing Tools
Lumia SensorCore SDK testing package includes sensor recorder and simulator tools. You can use them to record an...

Lumia SensorCore SDK
Lumia SensorCore SDK provides access to user's location and motion data. The SDK in...

Created by: Microsoft Mobile
Id: LumiaSensorCoreSDK
Version: 0.9.1.3
Last Published: 23.6.2014
Downloads: 112
License
View License
Project Information
Report Abuse
Description:
Lumia SensorCore SDK is a collection of 4 APIs utilising data from different sensors (for example, accelerometer) and also location information. This information can be used to track user's physical activities and motion. The sensors are able to run constantly in the background, collecting and preserving data for up to past ten days.

With the Lumia SensorCore SDK one can access to step counter that provides information on how many steps and for how long time the user has been walking. The SDK also provides information about changes in user's physical activity. For example, when user

Each package is licensed to you by its owner. Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.

```
using System;
using System.Coll
using System.IO;
using System.Linq
using System.Runt
using Windows.App
using Windows.App
using Windows.Fou
using Windows.Fou
using Windows.UI.
using Windows.UI.
using Windows.UI.
using Windows.UI.
using Windows.UI.
using Windows.UI.
using Windows.UI.
using Windows.UI.

// The Blank Appl
```

ApiState Class

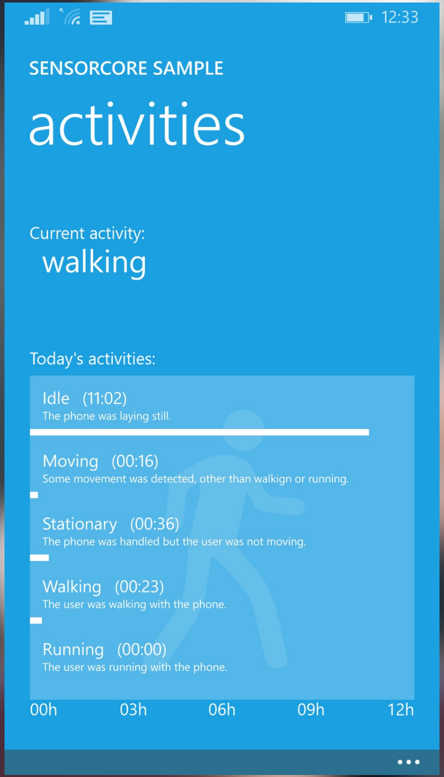
	Name	Description
	LocationEnabled	`true` if the location setting of the phone is enabled, `false` otherwise.
	SenseEnabled	`true` if the motion data setting of the phone is enabled, `false` otherwise.

SenseHelper Class

	Name	Description
S	GetApiStateAsync	Returns Sense and Location API state
S	GetSenseError	Returns Sense error matching the given HResult code from an exception
S	LaunchLocationSettingsAsync	Launches Location settings
S	LaunchSenseSettingsAsync	Launches Sense settings

Sensor Functionality Abstraction

ActivateAsync
DeactivateAsync
GetCurrentReadingAsync
Get<Name>AtAsync
Get<Name>HistoryAsync
IsSupported



12:33

SENSORCORE SAMPLE

activities

Current activity:
walking

Today's activities:

Idle (11:02)
The phone was laying still.

Moving (00:16)
Some movement was detected, other than walkign or running.

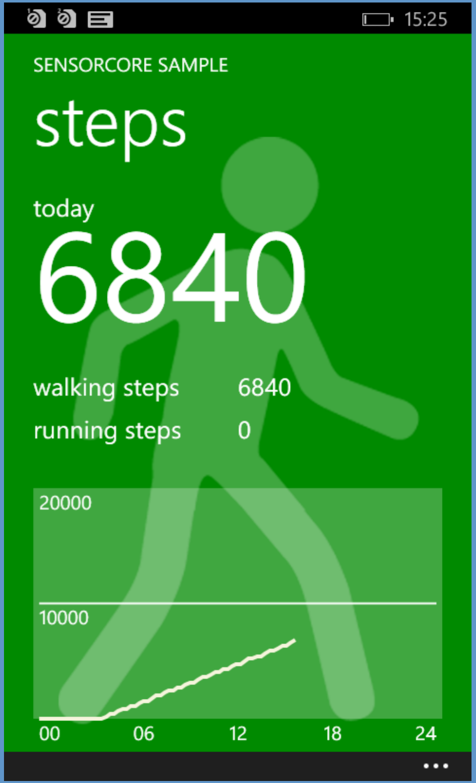
Stationary (00:36)
The phone was handled but the user was not moving.

Walking (00:23)
The user was walking with the phone.

Running (00:00)
The user was running with the phone.

00h 03h 06h 09h 12h

Step CounterAPI


15:25

SENSORCORE SAMPLE

steps

today

6840

walking steps	6840
running steps	0

StepCounter Class

Methods

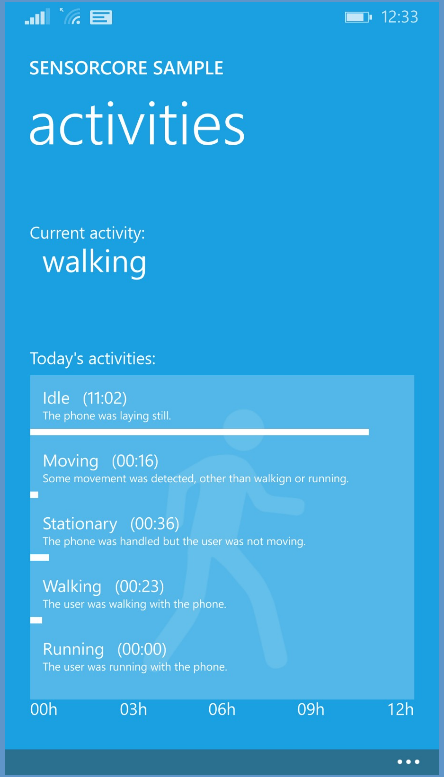
	Name	Description
◆	ActivateAsync	Reestablish the communication channel with underlying sensor, if not already exists
◆	DeactivateAsync	Close the communication with underlying sensor, this explicitly closes the communication channel.
◆	GetCurrentReadingAsync	Gets the current reading.
◆S	GetDefaultAsync	Gets the default implementation.
◆	GetStepCountAtAsync	Gets the step count at given time.
◆	GetStepCountHistoryAsync	Returns time ordered list of step counts during given time period. Data granularity is usually around five minutes.
◆S	IsSupportedAsync	Returns whether the sensor is supported by the device or not.

StepCounterReading Class

Properties

	Name	Description
	RunningStepCount	Gets the number of running steps taken since the motion data was enabled.
	RunTime	Gets the time spent running since the motion data was enabled.
	Timestamp	Gets the creation time of the sensor reading.
	WalkingStepCount	Gets the number of walking steps taken since the motion data was enabled.
	WalkTime	Gets the time spent walking since the motion data was enabled.

Activity Monitor API



12:33

SENSORCORE SAMPLE

activities

Current activity:
walking

Today's activities:

Idle (11:02)
The phone was laying still.

Moving (00:16)
Some movement was detected, other than walkign or running.

Stationary (00:36)
The phone was handled but the user was not moving.

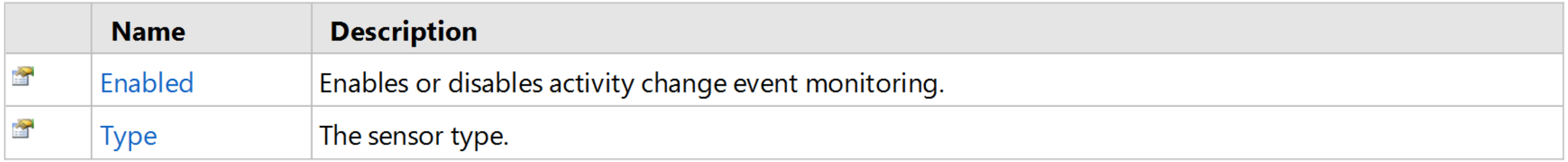
Walking (00:23)
The user was walking with the phone.

Running (00:00)
The user was running with the phone.

00h 03h 06h 09h 12h

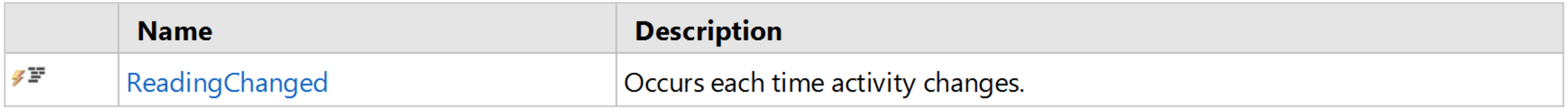
ActivityMonitor Class

Properties



	Name	Description
	Enabled	Enables or disables activity change event monitoring.
	Type	The sensor type.

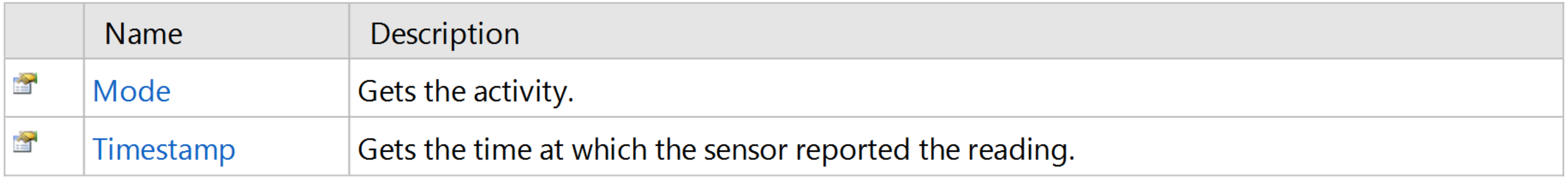
Events



	Name	Description
	ReadingChanged	Occurs each time activity changes.

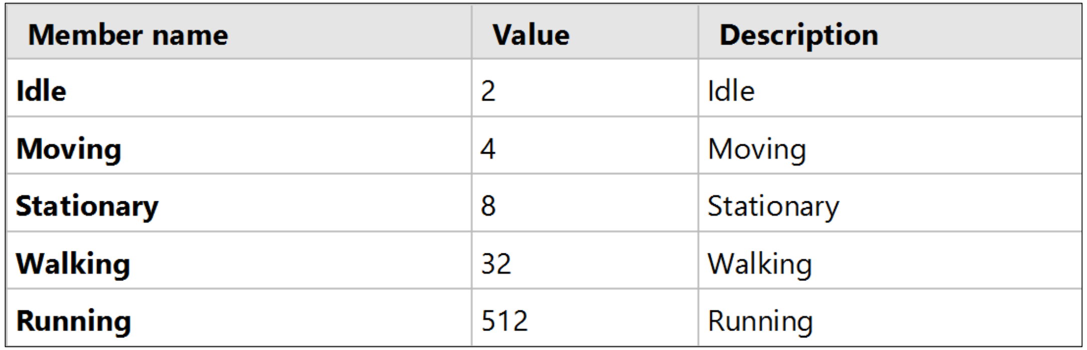
ActivityMonitorReading Class

Properties



	Name	Description
	Mode	Gets the activity.
	Timestamp	Gets the time at which the sensor reported the reading.

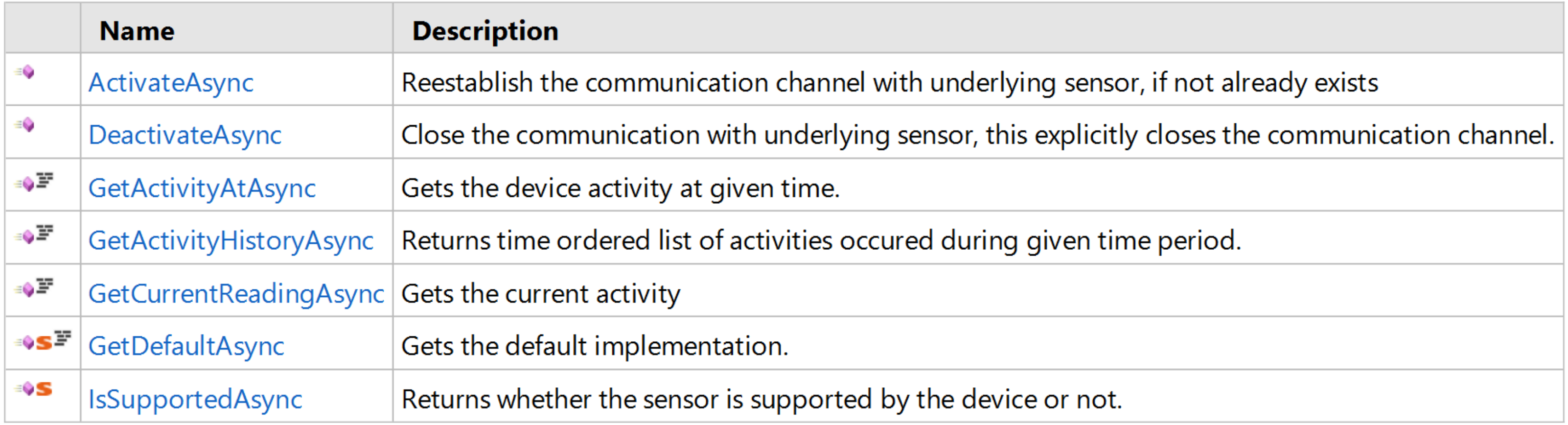
Activity Enumeration



Member name	Value	Description
Idle	2	Idle
Moving	4	Moving
Stationary	8	Stationary
Walking	32	Walking
Running	512	Running

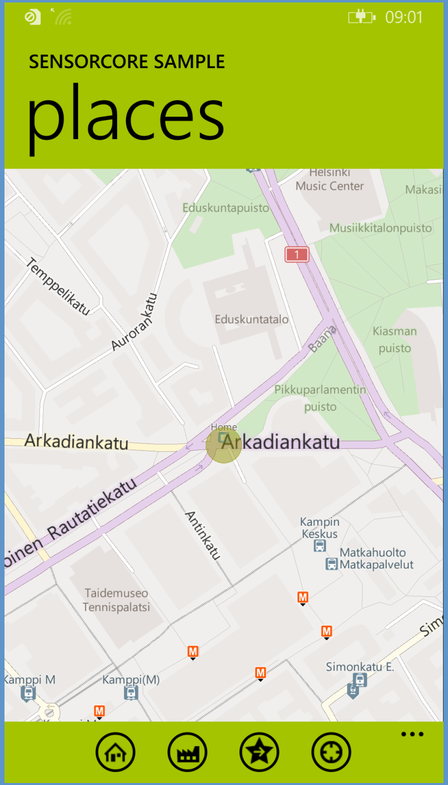
ActivityMonitor Class

Methods



	Name	Description
	ActivateAsync	Reestablish the communication channel with underlying sensor, if not already exists
	DeactivateAsync	Close the communication with underlying sensor, this explicitly closes the communication channel.
	GetActivityAtAsync	Gets the device activity at given time.
	GetActivityHistoryAsync	Returns time ordered list of activities occured during given time period.
	GetCurrentReadingAsync	Gets the current activity
S	GetDefaultAsync	Gets the default implementation.
S	IsSupportedAsync	Returns whether the sensor is supported by the device or not.

Place Monitor API



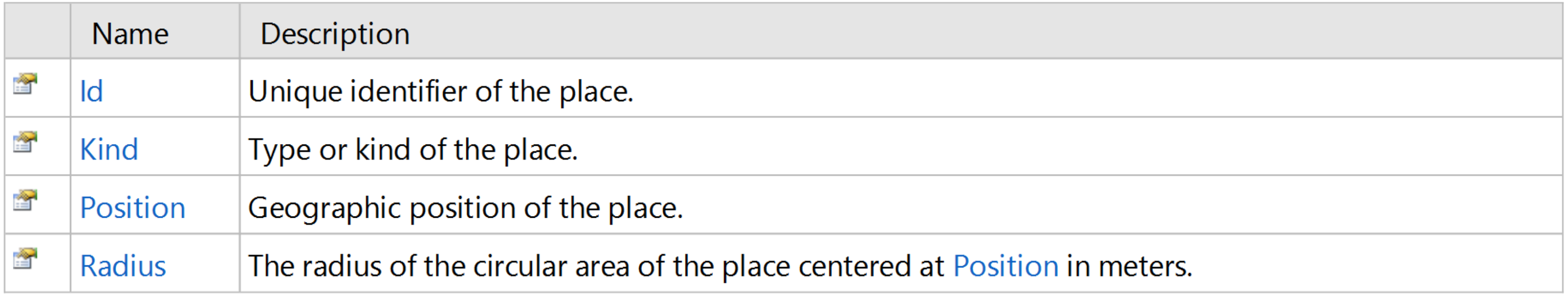
PlaceMonitor Class

Methods

	Name	Description
	ActivateAsync	Reestablish the communication channel with underlying sensor, if not already exists
	DeactivateAsync	Close the communication with underlying sensor, this explicitly closes the communication channel.
S	GetDefaultAsync	Gets the default implementation.
	GetHomeAsync	Gets the home location.
	GetKnownPlaceAsync	Gets place by place id.
	GetKnownPlacesAsync	Gets the set of currently known places.
	GetWorkAsync	Gets the work location.
S	IsSupportedAsync	Returns whether the sensor is supported by the device or not.

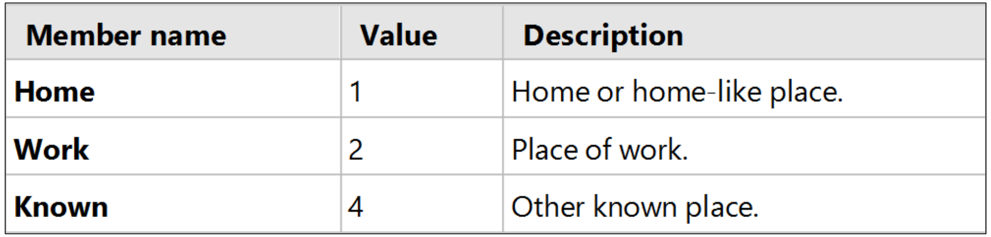
Place Class

Properties



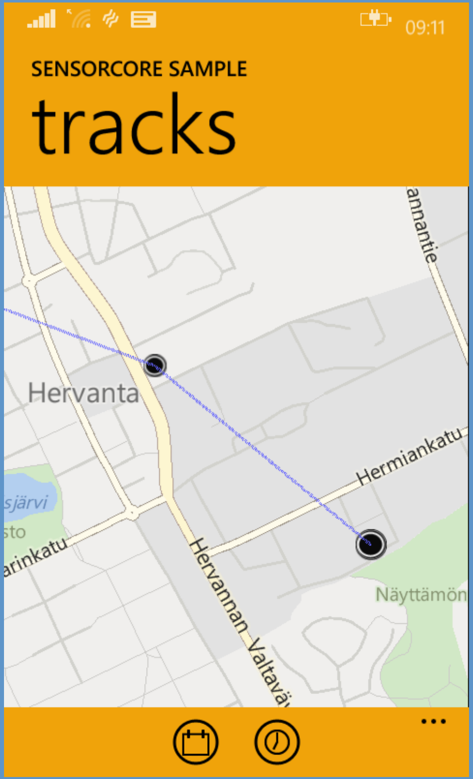
	Name	Description
	Id	Unique identifier of the place.
	Kind	Type or kind of the place.
	Position	Geographic position of the place.
	Radius	The radius of the circular area of the place centered at Position in meters.

PlaceKind Enumeration



Member name	Value	Description
Home	1	Home or home-like place.
Work	2	Place of work.
Known	4	Other known place.

Track Point Monitor API



TrackPointMonitor Class

Methods

	Name	Description
	ActivateAsync	Reestablish the communication channel with underlying sensor, if not already exists
	DeactivateAsync	Close the communication with underlying sensor, this explicitly closes the communication channel.
	GetDefaultAsync	Gets the default implementation.
	GetPointAtAsync	Returns the track point of the device at given time.
	GetTrackPointsAsync	Returns the collected track points the device moved during the given time period.
	IsSupportedAsync	Returns whether the sensor is supported by the device or not.

Properties

	Name	Description
	Type	The sensor type.

TrackPoint Class

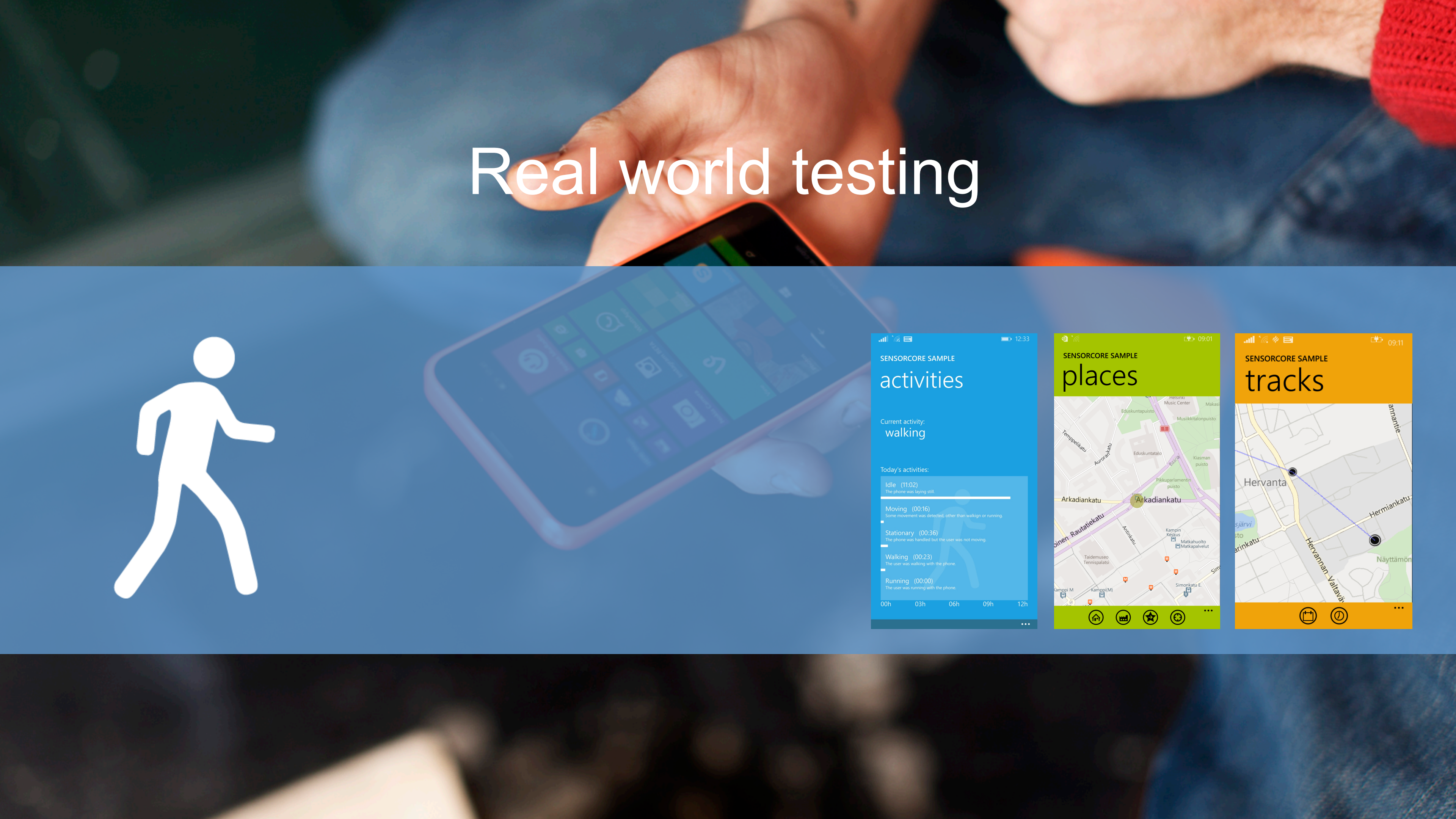
Constructors

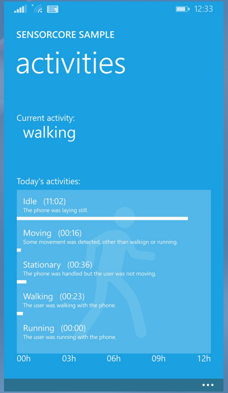
	Name	Description
	TrackPoint(TrackPoint)	Constructor
	TrackPoint(BasicGeoposition, Double, TimeSpan, DateTime)	Constructor

Properties

	Name	Description
	LengthOfStay	Time how long the device stayed at this point.
	Position	Geographic position of the track point.
	Radius	The estimated radius of a circular area around the location which reflects the used positioning technology.
	Timestamp	Time of entry to the location.

Real world testing




SENSORCORE SAMPLE

activities

Current activity:
walking

Today's activities:

Idle (11:02)
The phone was laying still.

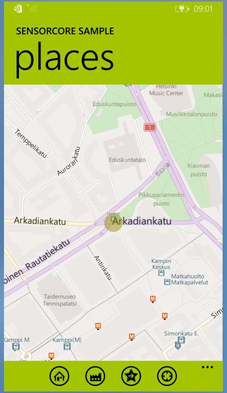
Moving (00:16)
Some movement was detected, other than walkign or running.

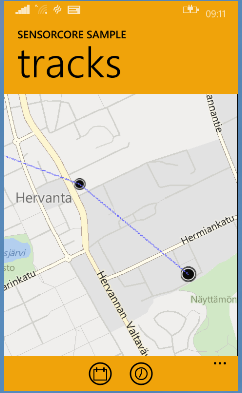
Stationary (00:36)
The phone was handled but the user was not moving.

Walking (00:23)
The user was walking with the phone.

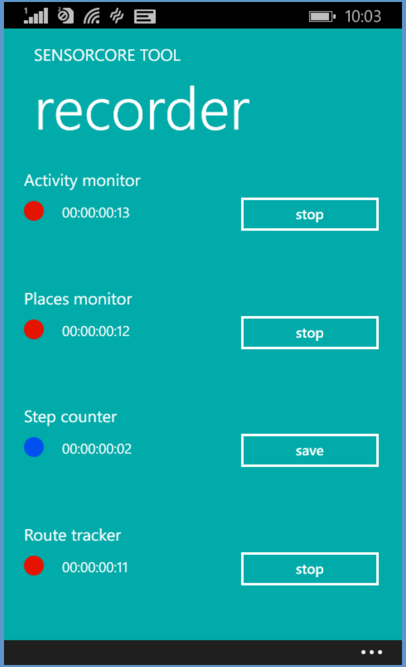
Running (00:00)
The user was running with the phone.

00h 03h 06h 09h 12h



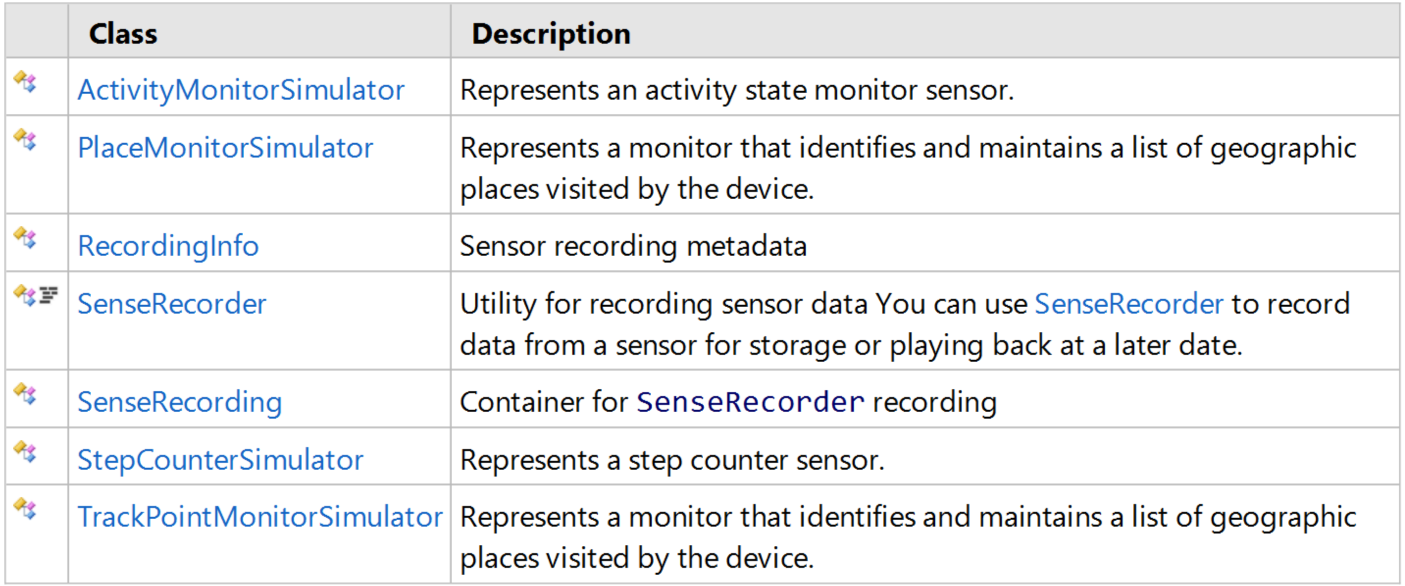


Testing Tools and Recorder



SensorCore SDK Testing Tools

Available classes for simulation



	Class	Description
	ActivityMonitorSimulator	Represents an activity state monitor sensor.
	PlaceMonitorSimulator	Represents a monitor that identifies and maintains a list of geographic places visited by the device.
	RecordingInfo	Sensor recording metadata
	SenseRecorder	Utility for recording sensor data You can use SenseRecorder to record data from a sensor for storage or playing back at a later date.
	SenseRecording	Container for `SenseRecorder` recording
	StepCounterSimulator	Represents a step counter sensor.
	TrackPointMonitorSimulator	Represents a monitor that identifies and maintains a list of geographic places visited by the device.

SenseRecorder Class

Methods

	Name	Description
	GetRecording	Returns the recording
	StartAsync	Starts recording
	StopAsync	Stops recording

Properties

	Name	Description
	IsRecording	Returns whether the recorder is currently recording or not.

SenseRecording Class

Methods

	Name	Description
	LoadFromFileAsync(String)	Loads **SenseRecording** from a file in application's installation directory.
	LoadFromFileAsync(String, UnicodeEncoding)	Loads **SenseRecording** from a file in application's installation directory
	LoadFromText(String)	Loads **SenseRecording** from given text string.
	LoadFromText(String, UnicodeEncoding)	Loads **SenseRecording** from given text string.
	SaveAsync	Prompts user to save the recording in Json format in Documents folder

Properties

	Name	Description
	Description	Description of the recording. Initially empty, can be modified by the developer to describe the recording before saving.
	Duration	Duration of the recording
	StartTime	Date of the recording
	Type	Recorded sensor type

Hands-On

SensorCore SDK

futurice LEAN SERVICE CREATION

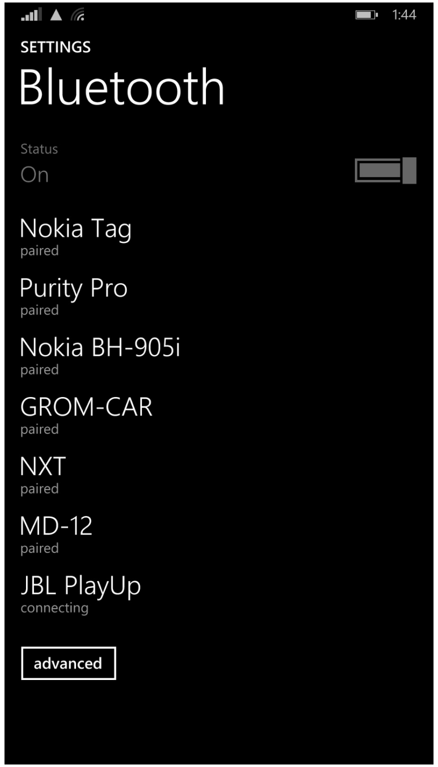
futurice

Bluetooth

RFComm and Lego NXT

Bluetooth Pairing

- For App-to-Device scenarios, the device must have been *paired* with the phone
- A device must be made “discoverable” before pairing can take place
  - Pairing is normally performed via the settings screen on the device
- During the pairing the connection is authenticated
- The user may need to enter a key to validate the connection
  - This may be a fixed key, as in the case of devices such as headsets, or generated by one device and entered on the other


1:44

SETTINGS

Bluetooth

Status

On

Nokia Tag
paired

Purity Pro
paired

Nokia BH-905i
paired

GROM-CAR
paired

NXT
paired

MD-12
paired

JBL PlayUp
connecting

App to Device

- An application running on a Windows Phone 8 device can obtain an enumeration of all the Bluetooth devices that have been paired with the phone
- The application can then attempt to make a connection to the required service on
that device
- For this to work the Bluetooth service on the phone must be turned on
- The ID_CAP_PROXIMITY and ID_CAP_NETWORKING capabilities must be enabled for the application to make use of the Bluetooth communications to a device

futurice LEAN SERVICE CREATION

Hands-On

Bluetooth and Lego NXT

futurice LEAN SERVICE CREATION

Thanks!

Have a great conference!

Michael
Samarin
@MichaelSamarin