# Halyri - Mobile client 1.0

Generated by Doxygen 1.8.7

Sat Jun 21 2014 20:11:36

# Contents

1	Nam	nespace Do	ocumentation	1
	1.1	Package I	Halyri	2
		1.1.1 D	etailed Description	3
		1.1.2 E	numeration Type Documentation	3
		1.1.3 F	unction Documentation	3
	1.2	Package I	Halyri.AudioVideoManagers	4
		1.2.1 F	unction Documentation	5
	1.3	Package I	Halyri.Conversion	5
	1.4	Package I	Halyri.ImportedServiceClasses	5
		1.4.1 D	etailed Description	6
	1.5	Package I	Halyri.Measurement	6
		1.5.1 F	unction Documentation	6
	1.6	Package I	Halyri.Resources	7
	1.7	Package I	Halyri.ViewModels	7
2	Clas	s Docume	Intation	7
	2.1	App Class	s Reference	7
		2.1.1 D	etailed Description	7
		2.1.2 C	Constructor & Destructor Documentation	7
		2.1.3 P	roperty Documentation	8
	2.2	AppReso	urces Class Reference	8
		2.2.1 D	etailed Description	11
		2.2.2 P	roperty Documentation	11
	2.3	AudioVide	eoPlaybackManager Class Reference	16
		2.3.1 D	etailed Description	16
		2.3.2 C	onstructor & Destructor Documentation	16
		2.3.3 M	lember Function Documentation	16
	2.4	AudioVide	eoTransmitManager Class Reference	17
		2.4.1 D	etailed Description	18
		2.4.2 C	Constructor & Destructor Documentation	19
		2.4.3 M	lember Function Documentation	19
		2.4.4 P	roperty Documentation	21
	2.5	Compress	sionHelper Class Reference	21
		2.5.1 D	etailed Description	21
		2.5.2 M	lember Function Documentation	21
	2.6	Connecte	dPage Class Reference	23
		2.6.1 D	etailed Description	23
		2.6.2 C	Constructor & Destructor Documentation	23

	2.6.3	Member Function Documentation	24
2.7	Conne	ction Class Reference	24
	2.7.1	Detailed Description	26
	2.7.2	Constructor & Destructor Documentation	26
	2.7.3	Member Function Documentation	26
	2.7.4	Member Data Documentation	30
	2.7.5	Property Documentation	30
	2.7.6	Event Documentation	31
2.8	EditPer	rsonalInfoPage Class Reference	31
	2.8.1	Detailed Description	31
	2.8.2	Constructor & Destructor Documentation	32
2.9	ItemVie	ewModel Class Reference	32
	2.9.1	Detailed Description	32
	2.9.2	Property Documentation	32
	2.9.3	Event Documentation	33
2.10	Localiz	edStrings Class Reference	33
	2.10.1	Detailed Description	33
2.11	MainPa	age Class Reference	33
	2.11.1	Detailed Description	33
	2.11.2	Constructor & Destructor Documentation	34
	2.11.3	Member Function Documentation	34
2.12	MainVi	ewModel Class Reference	34
	2.12.1	Detailed Description	34
	2.12.2	Member Function Documentation	35
	2.12.3	Property Documentation	35
	2.12.4	Event Documentation	35
2.13	Measu	rementInstrumentManager Class Reference	35
	2.13.1	Detailed Description	35
	2.13.2	Constructor & Destructor Documentation	36
	2.13.3	Member Function Documentation	36
2.14	Measu	rementManager Class Reference	36
	2.14.1	Detailed Description	36
	2.14.2	Constructor & Destructor Documentation	37
	2.14.3	Member Function Documentation	37
	2.14.4	Property Documentation	37
	2.14.5	Event Documentation	37
2.15	Media	ConfigurationDto Class Reference	37
	2.15.1	Detailed Description	38
	2.15.2	Member Function Documentation	38
	2.15.3	Member Data Documentation	38

	2.15.4 F	Property Documentation		 39
2.16	NotUrge	entPage Class Reference		 39
	2.16.1	Detailed Description		 39
	2.16.2	Constructor & Destructor Documentation	•	 40
2.17	Previewl	<pre>/ImageCapturer Class Reference</pre>		 40
	2.17.1	Detailed Description		 40
	2.17.2	Constructor & Destructor Documentation	•	 41
	2.17.3 N	Member Function Documentation	•	 41
	2.17.4 N	Member Data Documentation	•	 41
	2.17.5 F	Property Documentation		 41
2.18	Settings	s Class Reference	•	 42
	2.18.1	Detailed Description	•	 42
	2.18.2	Constructor & Destructor Documentation	•	 43
	2.18.3 N	Member Function Documentation		 43
	2.18.4 N	Member Data Documentation		 43
2.19	Settings/	sAndTutorialsPage Class Reference		 44
	2.19.1	Detailed Description	•	 45
	2.19.2 N	Member Enumeration Documentation		 45
	2.19.3	Constructor & Destructor Documentation		 45
	2.19.4 N	Member Function Documentation		 45
2.20	SpeexCo	CompressionHelper Class Reference		 45
	2.20.1	Detailed Description		 45
	2.20.2 M	Member Function Documentation		 46
2.21	UrgentPa	Page Class Reference		 46
	2.21.1	Detailed Description		 47
	2.21.2	Constructor & Destructor Documentation		 47
	2.21.3 M	Member Function Documentation		 47
2.22	WavAudi	dioSampleCapturer Class Reference		 47
	2.22.1	Detailed Description		 48
	2.22.2	Constructor & Destructor Documentation		 48
	2.22.3 M	Member Function Documentation		 48
	2.22.4 N	Member Data Documentation		 49
	2.22.5 F	Property Documentation		 49
2.23	WaveFile	leHeader Class Reference		 49
	2.23.1	Detailed Description		 49
	2.23.2 N	Member Function Documentation		 49

# 1 Namespace Documentation

# 1.1 Package Halyri

#### Namespaces

- package AudioVideoManagers
- package Conversion
- package ImportedServiceClasses
- package Measurement
- package Resources
- package ViewModels

#### Classes

class App

The class is mostly generated code, but contains little fix when exiting app. It disconnects connection and also stores *Connection* while app is running.

class AudioVideoTransmitManager

The class is used to managing audio and picture transmission from XNA Microphone and AudioVideoCapture⇔ Device. Media is captured using PreviewImageCapturer and WavAudioSampleCapturer classes. A capture configuration is taken from MediaConfigurationDto published by the server and available on NetworkConnection's Media⇔ ConfigurationUpdatedEvent. The network operations are performed by NetworkConnection class.

class ConnectedPage

The class ConnectedPage is a view and it is used when connection to the emergency exchange have been established and processed by emergency exchange handler. If NoSound is enabled by the user, the view opens up with a chat. The emergency exchange can open a map for the user to pinpoint his location on the map and send the location to the emergency exchange. Handler can also open video recording. By taping cant talk button it enables the chat also.

class Connection

The class handles all connection actions to the server. A user can connect, update information about the device, the personal info, the medical info or the location. It uses SignalR for receiving and WCF for sending.

class EditPersonalInfoPage

The class creates a view where the user can edit his personal info and save them.

class LocalizedStrings

The class provides access to string resources used for localization.

class MainPage

The applications main view. It contains information about the application and a user can navigate to other pages like a settings and urgetn and not urgent views. When the page is loaded it sets the language to the default culture of the device if not set in settings different.

class NotUrgentPage

In the page the user will determine what kind of help he needs by choosing the right option from listbox. The user can also navigate to urgent page if he needs to.

class Settings

The class can store any objects to the application storage. The user have to always save by using save function or the modifications do not take place in the storage.

class SettingsAndTutorialsPage

The class is used to view and change the application settings and watch the first aid tutorials.

class UrgentPage

In this page the connection happens visualy. When the connection is taken to process by the emergency exchange handler navigates to the Connected page. The user can turn GPS on and toggle NoSound.

Enumerations

- enum ConnectionStateDto {
- Arrived, InProgress, InTransfer, Processed,

Hold }

Connection states used to specify the current connection state such as arrived, hold and in progress.

Functions

delegate void PacketThrottlingStarted (object sender, EventArgs e)

The class delegate is for the packet send and throttling started events.

delegate void PacketSendStalled (object sender, EventArgs e)

The class delegate is for the packet sending stalled events.

- delegate void MediaSendFinishedDelegate (object sender, AsyncCompletedEventArgs e)
   The function is invoked when sending of the media is finished.
- delegate void UdpMediaDataReceived (object sender, MediaPacket media)
   The function is invoked when Udp media is received.
- delegate void MediaSocketHasFailed (object sender)

The function is invoked when mediaSocket has failed.

1.1.1 Detailed Description

Version 1.0.0 Last modified 24.5.2014

Copyright (C) 2014 Veli-Mikko Puupponen

Halyri is a prototype of an emergency call system. Its purpose is to demonstrate the use of the advanced capabilities available in the current generation smartphones in facilitating the emergency service dispatcher's capability to determine the nature of the emergency and to dispatch help.

For more information, see the README file of this package.

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN ← CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.1.2 Enumeration Type Documentation

#### 1.1.2.1 enum ConnectionStateDto

Connection states used to specify the current connection state such as arrived, hold and in progress.

<author>Veli-Mikko Puupponen</author> Simple state enum.

#### 1.1.3 Function Documentation

1.1.3.1 delegate void Halyri.MediaSendFinishedDelegate ( object sender, AsyncCompletedEventArgs e )

The function is invoked when sending of the media is finished.

1.1.3.2 delegate void Halyri.MediaSocketHasFailed ( object sender )

The function is invoked when mediaSocket has failed.

# 1.1.3.3 delegate void PacketSendStalled ( object sender, EventArgs e )

The class delegate is for the packet sending stalled events.

Delegate for packet send stalled events.

## Parameters

sender	The sending AudioVideoTransmitManager instance
е	EventArgs, normally empty.

## Parameters

sender	The sending AudioVideoTransmitManager instance
е	EventArgs, normally empty.

## 1.1.3.4 delegate void PacketThrottlingStarted (object sender, EventArgs e)

The class delegate is for the packet send and throttling started events.

Delegate for packet send throttling started events.

#### Parameters

sender	The sending AudioVideoTransmitManager instance
е	EventArgs, normally empty.

#### Parameters

sender	The sending AudioVideoTransmitManager instance
е	EventArgs, normally empty.

#### 1.1.3.5 delegate void Halyri.UdpMediaDataReceived ( object sender, MediaPacket media )

The function is invoked when Udp media is received.

#### **Parameters**

sender	The sender object.
media	The recived media packet.

# 1.2 Package Halyri.AudioVideoManagers

# Classes

class AudioVideoPlaybackManager

The class handles playback of incoming audio and picture data. It currently only performs the playback of speex encoded audio using XNA sound effect instances.

class PreviewImageCapturer

The class is used for capturing preview images from AudioVideoCaptureDevice and compressing them to the jpeg format. It enables the compressed frames to have lower resolution and frame rate than the AudioVideoCaptureDevice preview buffer generates.

class WavAudioSampleCapturer

The class is used for capturing PCM audio fragments from XNA microphone and processing them as WAVE files. • class WaveFileHeader

The class generates canonical WAVE header to an array of PCM samples.

# Functions

• delegate void JpgFrameCaptured (object sender, EventArgs e, byte[] jpgData)

The class delegate is for jpg frame captured events.

- delegate void WaveSegmentCaptured (object sender, EventArgs e, byte[] audioFrame)
   The class delegate is for wave file captured events.
- delegate void PcmSegmentCaptured (object sender, EventArgs e, byte[] audioFrame) The class delegate is for pcm data samples captured events.

#### 1.2.1 Function Documentation

1.2.1.1 delegate void Halyri.AudioVideoManagers.JpgFrameCaptured ( object sender, EventArgs e, byte[] jpgData )

The class delegate is for jpg frame captured events.

Parameters

sender	The sending object.
е	The eventargs, normally empty.
jpgData	The jpeg image data bytes.

1.2.1.2 delegate void Halyri.AudioVideoManagers.PcmSegmentCaptured ( object sender, EventArgs e, byte[] audioFrame )

The class delegate is for pcm data samples captured events.

#### **Parameters**

sender	The sending object.
е	The eventargs, normally empty.
audioFrame	The PCM data samples.

1.2.1.3 delegate void Halyri.AudioVideoManagers.WaveSegmentCaptured ( object sender, EventArgs e, byte[] audioFrame )

The class delegate is for wave file captured events.

#### Parameters

sender	The sending object.
е	The eventargs, normally empty.
audioFrame	The wve audio file fragment.

# 1.3 Package Halyri.Conversion

# Classes

class CompressionHelper

The class contains static compression methods.

class SpeexCompressionHelper

The class providies static methods for compression and decompression of speex encoded audio segments.

# 1.4 Package Halyri.ImportedServiceClasses

#### Classes

class MediaConfigurationDto

The Media configuration container class is manually cloned from the server project.

## 1.4.1 Detailed Description

Version 1.0.0 Last modified 24.5.2014

Copyright (C) 2014 Veli-Mikko Puupponen

Halyri is a prototype of an emergency call system. Its purpose is to demonstrate the use of the advanced capabilities available in the current generation smartphones in facilitating the emergency service dispatcher's capability to determine the nature of the emergency and to dispatch help.

For more information, see the README file of this package.

The MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, IN ← CLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# 1.5 Package Halyri.Measurement

#### Classes

class MeasurementInstrumentManager

The mock implementation of the class that manages the measurement instruments connected to the mobile device. The implementation contains a specification for a single measurement instrument and provides it on a list.

class MeasurementManager

The mock of the class that manages an active measurement session on a measurement instrument connected to the mobile device. It currently provides data conforming to the data structure parameters specified in the provided MeasurementInstrumentDto and imitating nommal ECG.

# Functions

delegate void MeasurementDataAvailable (MeasurementInstrumentDto instrument, byte[] data)

Is invoked when measurement data is available with the instrument given.

## 1.5.1 Function Documentation

1.5.1.1 delegate void Halyri.Measurement.MeasurementDataAvailable (MeasurementInstrumentDto instrument, byte[] data )

Is invoked when measurement data is available with the instrument given.

# Parameters

instrument	The instrument.
data	The data available.

# 1.6 Package Halyri.Resources

Classes

class AppResources

A strongly-typed resource class, for looking up localized strings, etc.

# 1.7 Package Halyri.ViewModels

Classes

class ItemViewModel

The class is mostly generated code. It is a static ViewModel used by the views to bind against.

- class MainViewModel
  - The class is mostly generated code. It is a static ViewModel used by the views to bind against.

# 2 Class Documentation

# 2.1 App Class Reference

The class is mostly generated code, but contains little fix when exiting app. It disconnects connection and also stores Connection while app is running.

Inherits Application.

# **Public Member Functions**

#### • App ()

The function is the constructor for the Application object.

#### **Properties**

- static MainViewModel ViewModel [get]
  - A static ViewModel used by the views to bind against.
- Connection Connection [get, set]

The function to get and set connection that app uses for connection management.

• static PhoneApplicationFrame RootFrame [get, set]

The function provides easy access to the root frame of the phone application.

#### 2.1.1 Detailed Description

The class is mostly generated code, but contains little fix when exiting app. It disconnects connection and also stores Connection while app is running.

#### 2.1.2 Constructor & Destructor Documentation

## 2.1.2.1 App()

The function is the constructor for the Application object.

# 2.1.3 Property Documentation

2.1.3.1 Connection Connection [get], [set]

The function to get and set connection that app uses for connection management.

2.1.3.2 PhoneApplicationFrame RootFrame [static], [get], [set]

The function provides easy access to the root frame of the phone application.

## Returns

The root frame of the phone application.

2.1.3.3 MainViewModel ViewModel [static], [get]

A static ViewModel used by the views to bind against.

# Returns

The MainViewModel object.

The documentation for this class was generated from the following file:

App.xaml.cs

# 2.2 AppResources Class Reference

A strongly-typed resource class, for looking up localized strings, etc.

## **Properties**

•	static
	global::System.Resources.ResourceManager ResourceManager [get]
	Returns the cached ResourceManager instance used by this class.
•	static
	global::System.Globalization.CultureInfo Culture [get, set]
	Overrides the current thread's CurrentUICulture property for all resource lookups using this strongly typed resource class.

• static string About [get]

Looks up a localized string similar to About.

- static string AllowGPS [get]
  - Looks up a localized string similar to Allow GPS.
- static string Ambulance [get]

Looks up a localized string similar to Ambulance.

- static string AppBarButtonText [get]
  - Looks up a localized string similar to add.
- static string AppBarMenuItemText [get]
  - Looks up a localized string similar to Menu Item.
- static string AppTitle [get]

Looks up a localized string similar to Hälyri.

• static string Attention [get]

Looks up a localized string similar to Attention!.

• static string Bigger [get]

Looks up a localized string similar to Bigger. • static string Biggest [get] Looks up a localized string similar to Biggest. • static string Cancel [get] Looks up a localized string similar to Cancel. static string CanMakeNoiseButtonText [get] Looks up a localized string similar to CAN speak. • static string CannotConnect [get] Looks up a localized string similar to Cannot connect.. static string CantMakeNoiseButtonText [get] Looks up a localized string similar to CANNOT speak. static string CantMakeNoiseDescription [get] Looks up a localized string similar to If you are in a situation that you cannot make any noise.. • static string ChangingToUrgentButtonDescription [get] Looks up a localized string similar to You can change this to urgent matter:. • static string ChatInfoText [get] Looks up a localized string similar to Type your message to white box and press send button.. static string ChooseLocationFirstText [get] Looks up a localized string similar to Please choose first your location.. • static string City [get] Looks up a localized string similar to City. • static string Connect [get] Looks up a localized string similar to Connect. static string Connected [get] Looks up a localized string similar to Connected. static string ConnectingToEmergencyExchange [get] Looks up a localized string similar to Connecting to emergency exchange. static string ConnectionClosed [get] Looks up a localized string similar to Connection closed. • static string Default [get] Looks up a localized string similar to Default. • static string Description [get] Looks up a localized string similar to Contact emergency exchange.. • static string DifferentKindsOfAssigments [get] Looks up a localized string similar to General kind of situations. static string Edit [get] Looks up a localized string similar to Edit. • static string EditPersonalInfo [get] Looks up a localized string similar to Edit personal information. static string EmergencyExchange [get] Looks up a localized string similar to Emergency exchange. • static string English [get] Looks up a localized string similar to English. • static string Finnish [get] Looks up a localized string similar to Finnish. • static string FireDepartment [get] Looks up a localized string similar to Fire Department. • static string Flood [get]

Looks up a localized string similar to Flood.

• static string Fontsize [get]

Looks up a localized string similar to Fontsize.

```
    static string Language [get]

      Looks up a localized string similar to Language.

    static string LocationDisabledContent [get]

      Looks up a localized string similar to Enable location in phone settings and applications settings.

    static string LocationDisabledTitle [get]

      Looks up a localized string similar to Location is disabled in phone settings...
• static string MapInfoText [get]
     Looks up a localized string similar to Tap your location and then send..
• static string Me [get]
     Looks up a localized string similar to Me.
• static string Name [get]
     Looks up a localized string similar to Name.
• static string Normal [get]
     Looks up a localized string similar to Normal.

    static string NotUrgentButtonDescription [get]

     Looks up a localized string similar to If you are NOT in hurry to get help tap not-urgent to choose what kind of help
     you are needing ..
• static string NotUrgentButtonText [get]
      Looks up a localized string similar to Not Urgent.
• static string NotUrgentDescription [get]
      Looks up a localized string similar to Let's define what exactly is the problem. Choose whose help you are needing:.

    static string PersonalInformation [get]

     Looks up a localized string similar to Personal infomation.
• static string PhoneNumber [get]
     Looks up a localized string similar to Phone Number.
• static string Police [get]
     Looks up a localized string similar to Police.

    static string PostalCode [get]

      Looks up a localized string similar to Postal code.
• static string PostalCodeCannotBeNegative [get]
      Looks up a localized string similar to Postalcode was not saved because it cannot be negative number.
• static string PracticeMode [get]
      Looks up a localized string similar to Practice mode.
• static string Processing [get]
     Looks up a localized string similar to Processing.

    static string ProcessingInfoText [get]

      Looks up a localized string similar to You have been taken for processing. Please take phone on your ear..

    static string RebootNeededForLanguageChange [get]

     Looks up a localized string similar to You have to reboot application to change language.

    static string ResourceFlowDirection [get]

      Looks up a localized string similar to LeftToRight.
• static string ResourceLanguage [get]
      Looks up a localized string similar to en-US.

    static string SampleProperty [get]

      Looks up a localized string similar to Sample Runtime Property Value.
• static string SavedText [get]
      Looks up a localized string similar to Saving was succesful..
• static string Send [get]
```

Looks up a localized string similar to Send.

• static string Settings [get] Looks up a localized string similar to Settings. • static string Smaller [get]

Looks up a localized string similar to Smaller.

• static string Smallest [get]

Looks up a localized string similar to Smallest.

• static string StormDamage [get]

Looks up a localized string similar to Storm damage.

static string StreetNameAndNumber [get]

Looks up a localized string similar to Street name and number.

• static string Tutorials [get]

Looks up a localized string similar to First aid tutorials.

• static string UrgentButtonDescription [get]

Looks up a localized string similar to To connect emergency exhance tap 112-logo:.

static string UrgentButtonText [get]

Looks up a localized string similar to URGENT.

static string VehicleBreakdown [get]

Looks up a localized string similar to Vehicle breakdown.

• static string WaitingForProcessInfoText [get]

Looks up a localized string similar to Waiting for connection to be processed ...

- static string YouAreLeavingNotice [get]
  - Looks up a localized string similar to Are you sure you want to leave?.

# 2.2.1 Detailed Description

A strongly-typed resource class, for looking up localized strings, etc.

# 2.2.2 Property Documentation

2.2.2.1 string About [static], [get]

Looks up a localized string similar to About.

2.2.2.2 string AllowGPS [static], [get]

Looks up a localized string similar to Allow GPS.

2.2.2.3 string Ambulance [static], [get]

Looks up a localized string similar to Ambulance.

2.2.2.4 string AppBarButtonText [static], [get]

Looks up a localized string similar to add.

2.2.2.5 string AppBarMenultemText [static], [get]

Looks up a localized string similar to Menu Item.

2.2.2.6 string AppTitle [static], [get]

Looks up a localized string similar to Hälyri.

2.2.2.7 string Attention [static], [get]

Looks up a localized string similar to Attention!.

2.2.2.8 string Bigger [static], [get]

Looks up a localized string similar to Bigger.

2.2.2.9 string Biggest [static], [get]

Looks up a localized string similar to Biggest.

2.2.2.10 string Cancel [static], [get]

Looks up a localized string similar to Cancel.

2.2.2.11 string CanMakeNoiseButtonText [static], [get]

Looks up a localized string similar to CAN speak.

2.2.2.12 string CannotConnect [static], [get]

Looks up a localized string similar to Cannot connect..

2.2.2.13 string CantMakeNoiseButtonText [static], [get]

Looks up a localized string similar to CANNOT speak.

2.2.2.14 string CantMakeNoiseDescription [static], [get]

Looks up a localized string similar to If you are in a situation that you cannot make any noise..

2.2.2.15 string ChangingToUrgentButtonDescription [static], [get]

Looks up a localized string similar to You can change this to urgent matter:.

2.2.2.16 string ChatInfoText [static], [get]

Looks up a localized string similar to Type your message to white box and press send button..

2.2.2.17 string ChooseLocationFirstText [static], [get]

Looks up a localized string similar to Please choose first your location..

2.2.2.18 string City [static], [get]

Looks up a localized string similar to City.

2.2.2.19 string Connect [static], [get]

Looks up a localized string similar to Connect.

2.2.2.20 string Connected [static], [get]

Looks up a localized string similar to Connected.

2.2.2.21 string ConnectingToEmergencyExchange [static], [get]

Looks up a localized string similar to Connecting to emergency exchange.

2.2.2.22 string ConnectionClosed [static],[get]

Looks up a localized string similar to Connection closed.

2.2.2.23 global.System.Globalization.CultureInfo Culture [static], [get], [set]

Overrides the current thread's CurrentUICulture property for all resource lookups using this strongly typed resource class.

2.2.2.24 string Default [static], [get]

Looks up a localized string similar to Default.

2.2.2.25 string Description [static], [get]

Looks up a localized string similar to Contact emergency exchange..

2.2.2.26 string DifferentKindsOfAssignments [static], [get]

Looks up a localized string similar to General kind of situations.

2.2.2.27 string Edit [static], [get]

Looks up a localized string similar to Edit.

2.2.2.28 string EditPersonalInfo [static], [get]

Looks up a localized string similar to Edit personal information.

2.2.2.29 string EmergencyExchange [static], [get]

Looks up a localized string similar to Emergency exchange.

2.2.2.30 string English [static], [get]

Looks up a localized string similar to English.

2.2.2.31 string Finnish [static], [get]

Looks up a localized string similar to Finnish.

2.2.2.32 string FireDepartment [static], [get]

Looks up a localized string similar to Fire Department.

2.2.2.33 string Flood [static], [get]

Looks up a localized string similar to Flood.

2.2.2.34 string Fontsize [static], [get]

Looks up a localized string similar to Fontsize.

2.2.2.35 string Language [static], [get]

Looks up a localized string similar to Language.

2.2.2.36 string Location Disabled Content [static], [get]

Looks up a localized string similar to Enable location in phone settings and applications settings...

2.2.2.37 string LocationDisabledTitle [static], [get]

Looks up a localized string similar to Location is disabled in phone settings..

2.2.2.38 string MapInfoText [static], [get]

Looks up a localized string similar to Tap your location and then send..

2.2.2.39 string Me [static], [get]

Looks up a localized string similar to Me.

2.2.2.40 string Name [static], [get]

Looks up a localized string similar to Name.

2.2.2.41 string Normal [static], [get]

Looks up a localized string similar to Normal.

2.2.2.42 string NotUrgentButtonDescription [static], [get]

Looks up a localized string similar to If you are NOT in hurry to get help tap not-urgent to choose what kind of help you are needing.

2.2.2.43 string NotUrgentButtonText [static], [get]

Looks up a localized string similar to Not Urgent.

2.2.2.44 string NotUrgentDescription [static], [get]

Looks up a localized string similar to Let's define what exactly is the problem. Choose whose help you are needing:.

2.2.2.45 string PersonalInformation [static], [get]

Looks up a localized string similar to Personal infomation.

2.2.2.46 string PhoneNumber [static], [get]

Looks up a localized string similar to Phone Number.

2.2.2.47 string Police [static], [get]

Looks up a localized string similar to Police.

2.2.2.48 string PostalCode [static], [get]

Looks up a localized string similar to Postal code.

2.2.2.49 string PostalCodeCannotBeNegative [static], [get]

Looks up a localized string similar to Postalcode was not saved because it cannot be negative number..

**2.2.2.50** string PracticeMode [static], [get]

Looks up a localized string similar to Practice mode.

2.2.2.51 string Processing [static], [get]

Looks up a localized string similar to Processing.

2.2.2.52 string ProcessingInfoText [static], [get]

Looks up a localized string similar to You have been taken for processing. Please take phone on your ear..

2.2.2.53 string RebootNeededForLanguageChange [static], [get]

Looks up a localized string similar to You have to reboot application to change language..

2.2.2.54 string ResourceFlowDirection [static], [get]

Looks up a localized string similar to LeftToRight.

2.2.2.55 string ResourceLanguage [static], [get]

Looks up a localized string similar to en-US.

2.2.2.56 global.System.Resources.ResourceManager ResourceManager [static], [get]

Returns the cached ResourceManager instance used by this class.

2.2.2.57 string SampleProperty [static], [get]

Looks up a localized string similar to Sample Runtime Property Value.

2.2.2.58 string SavedText [static], [get]

Looks up a localized string similar to Saving was succesful..

2.2.2.59 string Send [static], [get]

Looks up a localized string similar to Send.

2.2.2.60 string Settings [static], [get]

Looks up a localized string similar to Settings.

2.2.2.61 string Smaller [static], [get]

Looks up a localized string similar to Smaller.

2.2.2.62 string Smallest [static], [get]

Looks up a localized string similar to Smallest.

2.2.2.63 string StormDamage [static], [get]

Looks up a localized string similar to Storm damage.

2.2.2.64 string StreetNameAndNumber [static], [get]

Looks up a localized string similar to Street name and number.

2.2.2.65 string Tutorials [static], [get]

Looks up a localized string similar to First aid tutorials.

2.2.2.66 string UrgentButtonDescription [static], [get]

Looks up a localized string similar to To connect emergency exhance tap 112-logo:.

2.2.2.67 string UrgentButtonText [static], [get]

Looks up a localized string similar to URGENT.

2.2.2.68 string VehicleBreakdown [static], [get]

Looks up a localized string similar to Vehicle breakdown.

2.2.2.69 string WaitingForProcessInfoText [static], [get]

Looks up a localized string similar to Waiting for connection to be processed..

2.2.2.70 string YouAreLeavingNotice [static], [get]

Looks up a localized string similar to Are you sure you want to leave?.

The documentation for this class was generated from the following file:

AppResources.Designer.cs

#### 2.3 AudioVideoPlaybackManager Class Reference

The class handles playback of incoming audio and picture data. It currently only performs the playback of speex encoded audio using XNA sound effect instances.

**Public Member Functions** 

AudioVideoPlaybackManager ()

The function initializes a new AudioVideoPlaybackManager instance.

• void Enable ()

The function enables the media playback using the instance. Currently only the playback of speex compressed audio fragments is available.

• void Disable ()

The function disables the media playback at the instance.

void HandleIncomingMedia (MediaPacket mediaPacket)

The function handles audio and video data received from the remote client. It currently handles only speex encoded audio.

2.3.1 Detailed Description

The class handles playback of incoming audio and picture data. It currently only performs the playback of speex encoded audio using XNA sound effect instances.

<author>Veli-Mikko Puupponen</author> Note: The class should be converted to work with a MediaStream Source rather than playing the samples using XNA audio effect instances.

2.3.2 Constructor & Destructor Documentation

2.3.2.1 AudioVideoPlaybackManager ()

The function initializes a new AudioVideoPlaybackManager instance.

2.3.3 Member Function Documentation

2.3.3.1 void Disable ()

The function disables the media playback at the instance.

#### 2.3.3.2 void Enable ()

The function enables the media playback using the instance. Currently only the playback of speex compressed audio fragments is available.

2.3.3.3 void HandleIncomingMedia (MediaPacket mediaPacket)

The function handles audio and video data received from the remote client. It currently handles only speex encoded audio.

Parameters

mediaPacket	The media packet containing audio or video data.

The documentation for this class was generated from the following file:

AudioVideoPlaybackManager.cs

# 2.4 AudioVideoTransmitManager Class Reference

The class is used to managing audio and picture transmission from XNA Microphone and AudioVideoCapture Device. Media is captured using PreviewImageCapturer and WavAudioSampleCapturer classes. A capture configuration is taken from MediaConfigurationDto published by the server and available on NetworkConnection's Media ConfigurationUpdatedEvent. The network operations are performed by NetworkConnection class.

## **Public Member Functions**

delegate void CaptureConfigurationUpdated (object sender, EventArgs e, MediaConfigurationDto config)

The class delegate is for the capture configuration updated events.

AudioVideoTransmitManager (Connection networkConnection)

The function instantializes a Audio Video Transmit Manager that uses the provided network Connection. The network  $\leftarrow$  Connection has to be connected to a server with a valid session.

void MediaConfigurationUpdateHandler (object sender, EventArgs e, MediaConfigurationDto config)

The function handles the media uploading configurations received by networkConnection from the server. The method only disbles audio and picture transmission if they are currently active and the new configuration suggests that they should be disabled. It also handles the configuration changes to the active audio and picture capturer classes.

void StopAllCaptureTransmit ()

The function disables media capture and transmission from any device.

void ResetThrottlingState ()

The function resets the ECF transfer channel throttling state of the instance. If the packet transmission has been stalled due to throttling, the packet transmission will resume.

 void ExternalSourceDeviceConfigurationChanged (AudioVideoCaptureDevice videoCaptureDevice, Microphone audioCaptureDevice, MediaConfigurationDto mediaConfiguration)

The function handles changes to the external media source devices. The method should be called with a valid source device(s) when the media uploading configuration has transitioned from disabled to enabled state for either of the media types.

- delegate void CaptureConfigurationUpdated (object sender, EventArgs e, MediaConfigurationDto config)
- AudioVideoTransmitManager (Connection networkConnection)

Instantializes a AudioVideoTransmitManager that uses the provided NetworkConnection. The NetworkConnection has to be connected to a server with a valid session.

void MediaConfigurationUpdateHandler (object sender, EventArgs e, MediaConfigurationDto config)

Handles media uploading configurations received by NetworkConnection from server. This method only disbles audio and picture transmission if they are currently active and the new configuration suggests that they should be disabled. It also handles configuration changes to active audio and picture capturer classes.

void StopAllCaptureTransmit ()

Disables media capture and transmission from any device.

void ResetThrottlingState ()

Resets the ECF transfer channel throttling state of this instance. If packet transmissiong has been stalled due to throttling, packet transmission will resume.

 void ExternalSourceDeviceConfigurationChanged (AudioVideoCaptureDevice videoCaptureDevice, Microphone audioCaptureDevice, MediaConfigurationDto mediaConfiguration)

Handles changes to external media source devices. This method should be called with a valid source device(s) when the media uploading configuration has transitioned from disabled to enabled state for either of the media types.

# **Public Attributes**

- CaptureConfigurationUpdated CaptureConfigurationUpdatedEvent
- PacketThrottlingStarted PacketThrottlingStartedEvent
- PacketSendStalled PacketSendStalledEvent

#### Properties

• bool PacketThrottlingEnabled [get]

The function sets and returns whether the packet throttling is enabled or not.

• bool AudioEnabled [get]

The function sets and returns whether the audio is enabled or not.

• bool PictureEnabled [get]

The function sets and returns whether the picture is enabled or not.

bool UseUdp [get, set]

The function sets and returns use UDP. It is used to specify whether to use UDP or not.

## 2.4.1 Detailed Description

The class is used to managing audio and picture transmission from XNA Microphone and AudioVideoCapture Device. Media is captured using PreviewImageCapturer and WavAudioSampleCapturer classes. A capture configuration is taken from MediaConfigurationDto published by the server and available on NetworkConnection's Media ConfigurationUpdatedEvent. The network operations are performed by NetworkConnection class.

Class for managing audio and picture transmission from XNA Microphone and AudioVideoCaptureDevice. Media is captured using PreviewImageCapturer and WavAudioSampleCapturer classes. Capture configuration is taken from MediaConfigurationDto published by the server and available on NetworkConnection's MediaConfiguration UpdatedEvent. Network operations are performed by NetworkConnection class.

<author>Veli-Mikko Puupponen</author> The class implements a simple packet level throttling when using WCF connection by measuring packet transmission time and comparing it to a set threshold. For every packet whose time exceeds the threshold, a counter is incremented. Once the counter meets an overdue packet threshold level, set ratio of picture data packets are dropped. If the overdue counter keeps increasing, picture packets are dropped completely. If the counter still increases, all packets are dropped and the PacketSendStalledEvent is fired.

If UseUdp is true, it uses UDP transfer protocol and no packet throttling is performed by the class.

After this AudioVideoTransmitManager instance has been stalled, it has to be manually reset by a call to the Reset ThrottlingState method.

<author>Veli-Mikko Puupponen</author> Implements a simple packet level throttling when using WCF connection by measuring packet transmission time and comparing it to a set threshold. For every packet whose time exceeds the threshold, a counter is incremented. Once the counter meets overdue packet threshold level, set ratio of picture data packets are dropped. If the overdue counter keeps increasing, picture packets are dropped completely. If the counter still increases, all packets are dropped and the PacketSendStalledEvent is fired.

If UseUdp = true, uses UDP transfer protocol and no packet throttling is performed by this class.

After this AudioVideoTransmitManager instance has been stalled, it has to be manually reset by a call to the Reset ThrottlingState method.

## 2.4.2 Constructor & Destructor Documentation

2.4.2.1 AudioVideoTransmitManager ( Connection networkConnection )

The function instantializes a AudioVideoTransmitManager that uses the provided networkConnection. The networkConnection has to be connected to a server with a valid session.

Parameters

network⇔	NetworkConnection to be used to send pictures and audio.
Connection	

## 2.4.2.2 AudioVideoTransmitManager ( Connection networkConnection )

Instantializes a AudioVideoTransmitManager that uses the provided NetworkConnection. The NetworkConnection has to be connected to a server with a valid session.

Parameters

network⇔	NetworkConnection to be used to send pictures and audio
Connection	

## 2.4.3 Member Function Documentation

2.4.3.1 delegate void CaptureConfigurationUpdated ( object sender, EventArgs e, MediaConfigurationDto config )

The class delegate is for the capture configuration updated events.

#### Parameters

sender	The sending object
е	EventArgs, normally empty.
config	The updrated condifiguration .

2.4.3.2 void ExternalSourceDeviceConfigurationChanged ( AudioVideoCaptureDevice videoCaptureDevice, Microphone audioCaptureDevice, MediaConfigurationDto mediaConfiguration )

Handles changes to external media source devices. This method should be called with a valid source device(s) when the media uploading configuration has transitioned from disabled to enabled state for either of the media types.

Parameters

videoCapture⇔	AudioVideoCaptureDevice opened for at least video if video is enabled in the configuration
Device	
audioCapture⇔	XNA Microphone instance for capturing audio
Device	
media⇔	The media uploading configuration used to set the devices in the UI
Configuration	

2.4.3.3 void ExternalSourceDeviceConfigurationChanged ( AudioVideoCaptureDevice videoCaptureDevice, Microphone audioCaptureDevice, MediaConfigurationDto mediaConfiguration )

The function handles changes to the external media source devices. The method should be called with a valid source device(s) when the media uploading configuration has transitioned from disabled to enabled state for either of the media types.

# Parameters

videoCapture⇔	Specifies AudioVideoCaptureDevice opened for at least video if video is enabled in the con-
Device	figuration.
audioCapture⇔	The XNA Microphone instance for capturing audio.
Device	
media⇔	The media uploading configuration used to set the devices in the UI.
Configuration	

2.4.3.4 void MediaConfigurationUpdateHandler ( object sender, EventArgs e, MediaConfigurationDto config )

Handles media uploading configurations received by NetworkConnection from server. This method only disbles audio and picture transmission if they are currently active and the new configuration suggests that they should be disabled. It also handles configuration changes to active audio and picture capturer classes.

If picture or audio is changed from enabled to disabled or vice versa, the configuration event is propagated through CaptureConfigurationUpdatedEvent to to the UI managing the capture devices.

Parameters

sender	Sending object
е	Eventargs, not currently used
config	New media uploading configuration

2.4.3.5 void MediaConfigurationUpdateHandler (object sender, EventArgs e, MediaConfigurationDto config)

The function handles the media uploading configurations received by networkConnection from the server. The method only disbles audio and picture transmission if they are currently active and the new configuration suggests that they should be disabled. It also handles the configuration changes to the active audio and picture capturer classes.

If picture or audio is changed from enabled to disabled or vice versa, the configuration event is propagated through CaptureConfigurationUpdatedEvent to to the UI managing the capture devices.

Parameters

sender	The sending object.
е	Eventargs, not currently used.
config	The new media uploading configuration.

2.4.3.6 void ResetThrottlingState ( )

Resets the ECF transfer channel throttling state of this instance. If packet transmissiong has been stalled due to throttling, packet transmission will resume.

Has no effect if the UDP transfer channel is being used.

2.4.3.7 void ResetThrottlingState ()

The function resets the ECF transfer channel throttling state of the instance. If the packet transmission has been stalled due to throttling, the packet transmission will resume.

The function has no effect if the UDP transfer channel is being used.

2.4.3.8 void StopAllCaptureTransmit ()

Disables media capture and transmission from any device.

2.4.3.9 void StopAllCaptureTransmit ()

The function disables media capture and transmission from any device.

2.4.4 Property Documentation

2.4.4.1 bool AudioEnabled [get]

The function sets and returns whether the audio is enabled or not.

2.4.4.2 bool PacketThrottlingEnabled [get]

The function sets and returns whether the packet throttling is enabled or not.

2.4.4.3 bool PictureEnabled [get]

The function sets and returns whether the picture is enabled or not.

2.4.4.4 bool UseUdp [get], [set]

The function sets and returns use UDP. It is used to specify whether to use UDP or not.

The documentation for this class was generated from the following file:

AudioVideoManagers/AudioVideoTransmitManager.cs

# 2.5 CompressionHelper Class Reference

The class contains static compression methods.

# **Static Public Member Functions**

- static byte[] CompressGZip (byte[] sourceData, bool useOptimalCompression)
   The function compresses the provided byte array using the GZip algorithm provided by System.IO.Compression library.
- static byte[] DecompressGZip (byte[] sourceData)

The function decompresses the provided byte array using the GZip algorithm provided by System.IO.Compression library.

# 2.5.1 Detailed Description

The class contains static compression methods.

<author>Veli-Mikko Puupponen</author> NOTICE: IO.Compression-library will not work, if the solution has Active Config set to any CPU. For phone, this should be ARM and for emulator, x86.

2.5.2 Member Function Documentation

2.5.2.1 static byte [] CompressGZip ( byte[] sourceData, bool useOptimalCompression ) [static]

The function compresses the provided byte array using the GZip algorithm provided by System.IO.Compression library.

Parameters

sourceData	The data to be compressed.
useOptimal⇔	True, if optimal compression method is used otherwise the fastest.
Compression	

Returns

GZip compressed byte data.

2.5.2.2 static byte [] DecompressGZip ( byte[] sourceData ) [static]

The function decompresses the provided byte array using the GZip algorithm provided by System.IO.Compression library.

Parameters

sourceData The data to be decompressed.

#### Returns

GZip decompressed input byte data.

The documentation for this class was generated from the following file:

CompressionHelper.cs

#### 2.6 ConnectedPage Class Reference

The class ConnectedPage is a view and it is used when connection to the emergency exchange have been established and processed by emergency exchange handler. If NoSound is enabled by the user, the view opens up with a chat. The emergency exchange can open a map for the user to pinpoint his location on the map and send the location to the emergency exchange. Handler can also open video recording. By taping cant talk button it enables the chat also.

Inherits PhoneApplicationPage.

#### **Public Member Functions**

ConnectedPage ()

The function constructor takes the connection from that last view saved to the phone application service with the key "connection". Then it just adds all eventhandlers for it. It also starts xnadispatchtimer and prepares for the media transfering.

# **Protected Member Functions**

override async void OnBackKeyPress (System.ComponentModel.CancelEventArgs e)

The function cancels backey event. It adds a messagebox asking whether the user wants to leave or not. The user can cancel the backkey event here.

override void OnNavigatingFrom (NavigatingCancelEventArgs e)

The function stops capturing and sending data.

override void OnNavigatedTo (NavigationEventArgs e)

The function checks if there is no sound set in the previous view so UI can add a chat for the user. Updates location and starts tracking location so it will update movement.

#### 2.6.1 Detailed Description

The class ConnectedPage is a view and it is used when connection to the emergency exchange have been established and processed by emergency exchange handler. If NoSound is enabled by the user, the view opens up with a chat. The emergency exchange can open a map for the user to pinpoint his location on the map and send the location to the emergency exchange. Handler can also open video recording. By taping cant talk button it enables the chat also.

<author>Atte Söderlund</author>

2.6.2 Constructor & Destructor Documentation

```
2.6.2.1 ConnectedPage ()
```

The function constructor takes the connection from that last view saved to the phone application service with the key "connection". Then it just adds all eventhandlers for it. It also starts xnadispatchtimer and prepares for the media transfering.

2.6.3 Member Function Documentation

2.6.3.1 override async void OnBackKeyPress ( System.ComponentModel.CancelEventArgs e ) [protected]

The function cancels backey event. It adds a messagebox asking whether the user wants to leave or not. The user can cancel the backkey event here.

2.6.3.2 override void OnNavigatedTo (NavigationEventArgs e) [protected]

The function checks if there is no sound set in the previous view so UI can add a chat for the user. Updates location and starts tracking location so it will update movement.

2.6.3.3 override void OnNavigatingFrom (NavigatingCancelEventArgs e) [protected]

The function stops capturing and sending data.

The documentation for this class was generated from the following file:

ConnectedPage.xaml.cs

# 2.7 Connection Class Reference

The class handles all connection actions to the server. A user can connect, update information about the device, the personal info, the medical info or the location. It uses SignalR for receiving and WCF for sending.

**Public Member Functions** 

 delegate void MediaConfigurationRequestUpdated (object sender, EventArgs e, MediaConfigurationDto config)

Is invoked when the media configuration update is requested.

delegate void Connected ()

The function is invoked when connected to the server.

delegate void InProcess ()

The function is invoked when the connection is taken to be processed.

delegate void Processed ()

The function is invoked when the connection is processed.

delegate void OpenMapRequest ()

The function is invoked when a map is requested to be opened.

delegate void CloseMapRequest ()

The function is invoked when the map is requested to be closed.

delegate void TextMessageReceived (String message)

The function is invoked when there is a text message received.

delegate void MeasurementStartRequestReceived (MeasurementInstrumentDto instrument)

The function is invoked when a measurement data is requested to be started with the given instrument.

delegate void MeasurementStopRequestReceived (MeasurementInstrumentDto instrument)

The function is invoked when the measurement sending is requested to be stopped with the given instrument.

- delegate void MeasurementInstrumentListRequestReceived ()
- The function is invoked when request for measurement instrument list is received.
- delegate void ConnectionFailed ()
  - The function is invoked when the connection has failed.
- Connection ()

The function is constructor that does nothing.

- void Connect (ConnectionPriorityDto priority, string emergencyType)
  - Connects with EmergencyType. This should be used when priority is noturgent.

void Connect (ConnectionPriorityDto priority)

The function makes a ui thread that does the connecting. Even without connecting, it sets priority and checks if there is a guid in the settings and tries to reconnect with that.

void StartGpsTracking ()

The function creates a ui thread for the gps tracking to be started.

• void Disconnect ()

The function disconnects from the server.

• void Reconnect ()

The function reconnects with the given priority.

void UdpConnect ()

The function connects to the server with UDP.

void UdpDisconnect ()

The function disconnects the UDP connectio to the server.

bool UdpMediaSend (MediaInformation mediaInfo, byte[] mediaData, int originalLength)

The function sends media with UDP.

- void UpdatePriority (UpdateConnectionPriorityRequest priority)
- The function makes a ui thread that updates the priority.
- void UpdateLocation (LocationTypeDto type)
  - The function creates a ui thread for the location update.
- void UpdatePersonalInfo (UpdatePersonalInfoRequest personalInfo)
  - The function creates a ui thread for the personal info update.
- void UpdateDeviceInfo (UpdateDeviceInfoRequest deviceInfo)
  - The function creates a ui thread for updating the device info.
- void UpdateMedicalInfo (UpdateMedicalInfoRequest medicalInfo)

The function creates a ui thread for the medical info update.

- void SendMessage (String message)
  - The function crates a new thread to send the message to the server so it won't block the ui thread.
- void SwitchNoSound (bool value)
  - The function creates a ui thread for switching to no sound.
- void UpdateLocationUserSpecified (System.Device.Location.GeoCoordinate coordinates)
  - The function creates a ui thread for the user specific location update.
- void SendMedia (MediaInformationDto info, byte[] data, object userState)

The function creates a ui thread for the media sending.

- void UpdateMeasurementInstrumentList (ObservableCollection < MeasurementInstrumentDto > instrument List)
  - The function creates a ui thread for the measurement instrument list update.
- void UploadMeasurementInstrumentData (MeasurementInstrumentDto instrument, byte[] data)

The function uploads the provided instrument data from the specified instrument to the server.

void UpdateEmergencyType (string emergencyType)

The function creates a ui thread for the emergencytype update.

## **Public Attributes**

- MediaConfigurationRequestUpdated MediaConfigurationUpdatedEvent
  - The function is a event hanlder for the Media configuration updated event.
- MediaSendFinishedDelegate MediaSendFinishedEvent
  - The function is a event hanlder for the media sending finished event.
- UdpMediaDataReceived UdpMediaDataReceivedEvent
  - The function is a event hanlder for the UDP media received event.
- MediaSocketHasFailed MediaSocketHasFailedEvent

The function is a event hanlder for the media socket has failed event.

#### Properties

- bool isConnected [get]
  - The function returns whether there is a connection established to the server or not.
- bool IsGpsTrackingEnabled [get]
  - The function returns whether the GPS is enabled or not.

## Events

- Connected ConnectedEvent
  - The event for the Connected delegate.
- InProcess InProcessEvent
  - The event for the InProcess delegate.
- Processed ProcessedEvent
  - The event for the Processed delegate.
- OpenMapRequest OpenMapRequestEvent
  - The event for the OpenMapRequest delegate.
- CloseMapRequest CloseMapRequestEvent
   The event for the CloseMapRequest delegate.
- TextMessageReceived TextMessageReceivedEvent
  - The event for the TextMessageReceived delegate.
- MeasurementStartRequestReceived MeasurementStartRequestReceivedEvent
  - The event for the MeasurementStartRequestReceived delegate.
- MeasurementStopRequestReceived MeasurementStopRequestReceivedEvent
  - The event for the MeasurementStopRequestReceived delegate.
- MeasurementInstrumentListRequestReceived MeasurementInstrumentListRequestReceivedEvent
  - The event for the MeasurementInstrumentListRequestReceived delegate.
- ConnectionFailed ConnectionFailedEvent

The event for the ConnectionFailed delegate.

# 2.7.1 Detailed Description

The class handles all connection actions to the server. A user can connect, update information about the device, the personal info, the medical info or the location. It uses SignalR for receiving and WCF for sending.

<author>Atte Söderlund</author>

# 2.7.2 Constructor & Destructor Documentation

2.7.2.1 Connection ()

The function is constructor that does nothing.

- 2.7.3 Member Function Documentation
- 2.7.3.1 delegate void CloseMapRequest ( )

The function is invoked when the map is requested to be closed.

2.7.3.2 void Connect ( ConnectionPriorityDto priority, string emergencyType )

Connects with EmergencyType. This should be used when priority is noturgent.

**Parameters** 

priority	The user specified priority of emergency.
emergency Type	What is happening.

2.7.3.3 void Connect ( ConnectionPriorityDto priority )

The function makes a ui thread that does the connecting. Even without connecting, it sets priority and checks if there is a guid in the settings and tries to reconnect with that.

Parameters

*priority* The user specified priority of emergency.

2.7.3.4 delegate void Connected ( )

The function is invoked when connected to the server.

2.7.3.5 delegate void ConnectionFailed ()

The function is invoked when the connection has failed.

2.7.3.6 void Disconnect ( )

The function disconnects from the server.

2.7.3.7 delegate void InProcess ()

The function is invoked when the connection is taken to be processed.

2.7.3.8 delegate void MeasurementInstrumentListRequestReceived ( )

The function is invoked when request for measurement instrument list is received.

2.7.3.9 delegate void MeasurementStartRequestReceived ( MeasurementInstrumentDto instrument )

The function is invoked when a measurement data is requested to be started with the given instrument.

Parameters

*instrument* | The given instrument.

2.7.3.10 delegate void MeasurementStopRequestReceived (MeasurementInstrumentDto instrument)

The function is invoked when the measurement sending is requested to be stopped with the given instrument. Parameters

*instrument* | The given instrument.

2.7.3.11 delegate void MediaConfigurationRequestUpdated (object sender, EventArgs e, MediaConfigurationDto config)

Is invoked when the media configuration update is requested.

#### Parameters

config | The media configuration.

2.7.3.12 delegate void OpenMapRequest ( )

The function is invoked when a map is requested to be opened.

## 2.7.3.13 delegate void Processed ( )

The function is invoked when the connection is processed.

2.7.3.14 void Reconnect ( )

The function reconnects with the given priority.

Parameters

*priority* | The given priority.

2.7.3.15 void SendMedia ( MediaInformationDto info, byte[] data, object userState )

The function creates a ui thread for the media sending.

<author>Atte Söderlund</author>

Parameters

info	The information of the media.
data	The media data.
userState	The user state object for tracking the corresponding asynchronous operation that have com-
	pleted.

2.7.3.16 void SendMessage ( String message )

The function crates a new thread to send the message to the server so it won't block the ui thread.

#### **Parameters**

message	The message to be sent.

2.7.3.17 void StartGpsTracking ( )

The function creates a ui thread for the gps tracking to be started.

2.7.3.18 void SwitchNoSound ( bool value )

The function creates a ui thread for switching to no sound.

**Parameters** 

value | Value to where to switch.

## 2.7.3.19 delegate void TextMessageReceived ( String message )

The function is invoked when there is a text message received.

Parameters

*message* | The received message.

2.7.3.20 void UdpConnect ( )

The function connects to the server with UDP.

<author>Veli-Mikko Puupponen</author>

2.7.3.21 void UdpDisconnect ( )

The function disconnects the UDP connectio to the server.

<author>Veli-Mikko Puupponen</author>

2.7.3.22 bool UdpMediaSend (MediaInformation mediaInfo, byte[] mediaData, int originalLength )

The function sends media with UDP.

<author>Veli-Mikko Puupponen</author>

Parameters

mediaInfo	The media information.
mediaData	The data.
originalLength	The lenght of the data.

Returns

Whether the media was sent or not.

2.7.3.23 void UpdateDeviceInfo ( UpdateDeviceInfoRequest deviceInfo )

The function creates a ui thread for updating the device info.

Parameters

<i>deviceInfo</i> The device info to be updated.	
--	--

2.7.3.24 void UpdateEmergencyType ( string emergencyType )

The function creates a ui thread for the emergencytype update.

#### Parameters

emergencyType The type of emergency.

2.7.3.25 void UpdateLocation ( LocationTypeDto type )

The function creates a ui thread for the location update.

Parameters

*location* Location to be updated.

2.7.3.26 void UpdateLocationUserSpecified ( System.Device.Location.GeoCoordinate coordinates )

The function creates a ui thread for the user specific location update.

Parameters

coordinates The specified coordinates.

2.7.3.27 void UpdateMeasurementInstrumentList ( ObservableCollection < MeasurementInstrumentDto > instrumentList )

The function creates a ui thread for the measurement instrument list update.

<author>Atte Söderlund</author>

Parameters

2.7.3.28 void UpdateMedicalInfo ( UpdateMedicalInfoRequest medicalInfo )

The function creates a ui thread for the medical info update.

Parameters

medicalInfo	The medical info to be updated.
-------------	---------------------------------

#### 2.7.3.29 void UpdatePersonalInfo ( UpdatePersonalInfoRequest personalInfo )

The function creates a ui thread for the personal info update.

Parameters

personalInfo	The personalinfo to be updated.

2.7.3.30 void UpdatePriority ( UpdateConnectionPriorityRequest priority )

The function makes a ui thread that updates the priority.

Parameters
------------

priority Priority to be updated.

2.7.3.31 void UploadMeasurementInstrumentData (MeasurementInstrumentDto instrument, byte[] data )

The function uploads the provided instrument data from the specified instrument to the server.

<author>Niko Mononen</author>

Parameters

instrument	The source instrument of the measurement data.
data	The data provided by the instrument.

## 2.7.4 Member Data Documentation

2.7.4.1 MediaConfigurationRequestUpdated MediaConfigurationUpdatedEvent

The function is a event hanlder for the Media configuration updated event.

2.7.4.2 MediaSendFinishedDelegate MediaSendFinishedEvent

The function is a event hanlder for the media sending finished event.

2.7.4.3 MediaSocketHasFailed MediaSocketHasFailedEvent

The function is a event handler for the media socket has failed event.

# 2.7.4.4 UdpMediaDataReceived UdpMediaDataReceivedEvent

The function is a event hanlder for the UDP media received event.

2.7.5 Property Documentation

2.7.5.1 bool isConnected [get]

The function returns whether there is a connection established to the server or not.

2.7.5.2 bool IsGpsTrackingEnabled [get]

The function returns whether the GPS is enabled or not.

2.7.6.1 CloseMapRequest CloseMapRequestEvent
The event for the CloseMapRequest delegate.
2.7.6.2 Connected ConnectedEvent
The event for the Connected delegate.

2.7.6.3 ConnectionFailed ConnectionFailedEvent

The event for the ConnectionFailed delegate.

2.7.6.4 InProcess InProcessEvent

2.7.6 Event Documentation

The event for the InProcess delegate.

2.7.6.5 MeasurementInstrumentListRequestReceived MeasurementInstrumentListRequestReceivedEvent

The event for the MeasurementInstrumentListRequestReceived delegate.

2.7.6.6 MeasurementStartRequestReceived MeasurementStartRequestReceivedEvent

The event for the MeasurementStartRequestReceived delegate.

2.7.6.7 MeasurementStopRequestReceived MeasurementStopRequestReceivedEvent

The event for the MeasurementStopRequestReceived delegate.

2.7.6.8 OpenMapRequest OpenMapRequestEvent

The event for the OpenMapRequest delegate.

2.7.6.9 Processed ProcessedEvent

The event for the Processed delegate.

2.7.6.10 TextMessageReceived TextMessageReceivedEvent

The event for the TextMessageReceived delegate.

The documentation for this class was generated from the following file:

Connection.cs

# 2.8 EditPersonalInfoPage Class Reference

The class creates a view where the user can edit his personal info and save them. Inherits PhoneApplicationPage.

**Public Member Functions** 

• EditPersonalInfoPage () The constructor loads the stored personal info.

#### 2.8.1 Detailed Description

The class creates a view where the user can edit his personal info and save them.

<author>Atte Söderlund</author>

2.8.2 Constructor & Destructor Documentation

2.8.2.1 EditPersonalInfoPage ()

The constructor loads the stored personal info.

The documentation for this class was generated from the following file:

EditPersonalInfoPage.xaml.cs

# 2.9 ItemViewModel Class Reference

The class is mostly generated code. It is a static ViewModel used by the views to bind against. Inherits INotifyPropertyChanged.

# **Properties**

• string LineOne [get, set]

The sample ViewModel property is used in the view to display its line value using a Binding.

• string LineTwo [get, set]

The sample ViewModel property is used in the view to display its line value using a Binding.

• string LineThree [get, set]

The sample ViewModel property is used in the view to display its line value using a Binding.

## **Events**

PropertyChangedEventHandler PropertyChanged
 The event to notify that property has changed.

#### 2.9.1 Detailed Description

The class is mostly generated code. It is a static ViewModel used by the views to bind against.

2.9.2 Property Documentation

2.9.2.1 string LineOne [get], [set]

The sample ViewModel property is used in the view to display its line value using a Binding.

2.9.2.2 string LineThree [get], [set]

The sample ViewModel property is used in the view to display its line value using a Binding.

2.9.2.3 string LineTwo [get], [set]

The sample ViewModel property is used in the view to display its line value using a Binding.

#### 2.9.3 Event Documentation

2.9.3.1 PropertyChangedEventHandler PropertyChanged

The event to notify that property has changed.

The documentation for this class was generated from the following file:

ItemViewModel.cs

# 2.10 LocalizedStrings Class Reference

The class provides access to string resources used for localization.

## Properties

• AppResources LocalizedResources [get]

#### 2.10.1 Detailed Description

The class provides access to string resources used for localization.

The documentation for this class was generated from the following file:

· LocalizedStrings.cs

# 2.11 MainPage Class Reference

The applications main view. It contains information about the application and a user can navigate to other pages like a settings and urgetn and not urgent views. When the page is loaded it sets the language to the default culture of the device if not set in settings different.

Inherits PhoneApplicationPage.

#### **Public Member Functions**

- MainPage ()
  - The function sets language and builds localized application bar.

#### **Protected Member Functions**

override void OnNavigatedTo (NavigationEventArgs e)

The function loads the data for the ViewModel Items.

override async void OnBackKeyPress (System.ComponentModel.CancelEventArgs e)

The function cancels backey event. It adds a messagebox to confirm whether the user wants to leave the page or not. The user can cancel backkey event here.

# 2.11.1 Detailed Description

The applications main view. It contains information about the application and a user can navigate to other pages like a settings and urgetn and not urgent views. When the page is loaded it sets the language to the default culture of the device if not set in settings different.

<author>Atte Söderlund</author>

#### 2.11.2 Constructor & Destructor Documentation

2.11.2.1 MainPage()

The function sets language and builds localized application bar.

2.11.3 Member Function Documentation

2.11.3.1 override async void OnBackKeyPress (System.ComponentModel.CancelEventArgs e) [protected]

The function cancels backey event. It adds a messagebox to confirm whether the user wants to leave the page or not. The user can cancel backkey event here.

2.11.3.2 override void OnNavigatedTo (NavigationEventArgs e) [protected]

The function loads the data for the ViewModel Items.

The documentation for this class was generated from the following file:

MainPage.xaml.cs

# 2.12 MainViewModel Class Reference

The class is mostly generated code. It is a static ViewModel used by the views to bind against. Inherits INotifyPropertyChanged.

**Public Member Functions** 

```
• void LoadData ()
```

The function creates and adds 16 ItemViewModel objects into the Items collection.

#### Properties

- · ObservableCollection
  - < ItemViewModel > Items [get, set]

The get and set functions for the collection of ItemViewModel objects.

- string SampleProperty [get, set]
  - The sample ViewModel property is used in the view to display its value using a Binding.
- string LocalizedSampleProperty [get]

The sample property that returns a localized string.

• bool IsDataLoaded [get, set]

The function gets and sets whether the data is loaded or not.

# **Events**

• PropertyChangedEventHandler PropertyChanged The event to notify that property has changed.

2.12.1 Detailed Description

The class is mostly generated code. It is a static ViewModel used by the views to bind against.

2.12.2 Member Function Documentation

2.12.2.1 void LoadData ( )

The function creates and adds 16 ItemViewModel objects into the Items collection.

2.12.3 Property Documentation

2.12.3.1 bool IsDataLoaded [get], [set]

The function gets and sets whether the data is loaded or not.

2.12.3.2 ObservableCollection<ItemViewModel>Items [get], [set]

The get and set functions for the collection of ItemViewModel objects.

2.12.3.3 string LocalizedSampleProperty [get]

The sample property that returns a localized string.

2.12.3.4 string SampleProperty [get], [set]

The sample ViewModel property is used in the view to display its value using a Binding.

## 2.12.4 Event Documentation

2.12.4.1 PropertyChangedEventHandler PropertyChanged

The event to notify that property has changed.

The documentation for this class was generated from the following file:

MainViewModel.cs

# 2.13 MeasurementInstrumentManager Class Reference

The mock implementation of the class that manages the measurement instruments connected to the mobile device. The implementation contains a specification for a single measurement instrument and provides it on a list.

# **Public Member Functions**

#### MeasurementInstrumentManager ()

The function initializes a new MeasurementInstrumentManager that provides always a single mock ECG measurement instrument.

- ObservableCollection
  - < MeasurementInstrumentDto > GetAvailableMeasurementInstrumentList ()

The function provides list of measurement instruments available at this mobile device. It currently provides a list with a single mock ECG instrument.

#### 2.13.1 Detailed Description

The mock implementation of the class that manages the measurement instruments connected to the mobile device. The implementation contains a specification for a single measurement instrument and provides it on a list.

<author>Veli-Mikko Puupponen</author>

2.13.2 Constructor & Destructor Documentation

2.13.2.1 MeasurementInstrumentManager ( )

The function initializes a new MeasurementInstrumentManager that provides always a single mock ECG measurement instrument.

2.13.3 Member Function Documentation

2.13.3.1 ObservableCollection<MeasurementInstrumentDto> GetAvailableMeasurementInstrumentList()

The function provides list of measurement instruments available at this mobile device. It currently provides a list with a single mock ECG instrument.

Returns

A list of sing mock ECG instrument.

The documentation for this class was generated from the following file:

MeasurementInstrumentManager.cs

# 2.14 MeasurementManager Class Reference

The mock of the class that manages an active measurement session on a measurement instrument connected to the mobile device. It currently provides data conforming to the data structure parameters specified in the provided MeasurementInstrumentDto and imitating nomrmal ECG.

**Public Member Functions** 

MeasurementManager (MeasurementInstrumentDto instrument)

The function initializes a new mock MeasurementManager instance that provides the mock samples conforming to the data parameters specified by the provided instrument.

void StartMeasurement ()

The function starts generating mock ECG data. MeasurementDataAvailableEvent is fired every time a new measurement data sample is available.

void StopMeasurement ()

The function stops the measurement.

#### **Properties**

• bool Active [get]

The function returns whether the measurement managaer is active or not.

#### **Events**

MeasurementDataAvailable MeasurementDataAvailableEvent

The event for the MeasurementDataAvailable delegate.

# 2.14.1 Detailed Description

The mock of the class that manages an active measurement session on a measurement instrument connected to the mobile device. It currently provides data conforming to the data structure parameters specified in the provided MeasurementInstrumentDto and imitating nomrmal ECG.

<author>Veli-Mikko Puupponen</author> The class assumes the DataSampleSize to be 1 byte.

#### 2.14.2 Constructor & Destructor Documentation

2.14.2.1 MeasurementManager (MeasurementInstrumentDto instrument)

The function initializes a new mock MeasurementManager instance that provides the mock samples conforming to the data parameters specified by the provided instrument.

#### **Parameters**

instrument	The instrument used as the data source.

#### 2.14.3 Member Function Documentation

2.14.3.1 void StartMeasurement ( )

The function starts generating mock ECG data. MeasurementDataAvailableEvent is fired every time a new measurement data sample is available.

2.14.3.2 void StopMeasurement ( )

The function stops the measurement.

2.14.4 Property Documentation

```
2.14.4.1 bool Active [get]
```

The function returns whether the measurement managaer is active or not.

#### 2.14.5 Event Documentation

2.14.5.1 MeasurementDataAvailable MeasurementDataAvailableEvent

The event for the MeasurementDataAvailable delegate.

The documentation for this class was generated from the following file:

MeasurementManager.cs

# 2.15 MediaConfigurationDto Class Reference

The Media configuration container class is manually cloned from the server project.

**Public Member Functions** 

void PerformConstraintCheck ()

The function validates AudioComressionQuality and PictureCompressionQuality.

#### **Public Attributes**

- int AudioCompressionQualityMin = 0
  - The minimum quality of the audio compression.
- int AudioCompressionQualityMax = 10
  - The maximum quality of the audio compression.
- const float PictureFpsMin = 0.0F

The minimum fps for the picture.

int PictureCompressionQualityMin = 0

The minimum quality of the picture compression.

- int PictureCompressionQualityMax = 100
  - The maximum quality of the picture compression.

# Properties

- bool EnablePicture [get, set]
- The function gets and sets whether the picture is enabled or not.
- bool EnableAudio [get, set]

The function gets and sets whether the audio is enabled or not.

• int AudioCompressionQuality [get, set]

The function sets and gets audio compression quality. The audio compression quality should be an integer between 0-10, while 10 being the best quality.

• float PictureFps [get, set]

The function gets and sets the frames per second to the picture. Values under 1 will result in frame rates with fewer than one frames per second.

• int PictureCompressionQuality [get, set]

The function gets and sets the picture compression quality. It should be an integer between 0-100, while 100 being the best quality.

• int PictureResolution [get, set]

The function gets and sets the picture resolution. The resolution must be an integer between 0-10, while 10 being the largest resolution available.

# 2.15.1 Detailed Description

The Media configuration container class is manually cloned from the server project.

<author>Veli-Mikko Puupponen</author>

- 2.15.2 Member Function Documentation
- 2.15.2.1 void PerformConstraintCheck ()

The function validates AudioComressionQuality and PictureCompressionQuality.

2.15.3 Member Data Documentation

2.15.3.1 int AudioCompressionQualityMax = 10

The maximum quality of the audio compression.

2.15.3.2 int AudioCompressionQualityMin = 0

The minimum quality of the audio compression.

2.15.3.3 int PictureCompressionQualityMax = 100

The maximum quality of the picture compression.

2.15.3.4 int PictureCompressionQualityMin = 0

The minimum quality of the picture compression.

2.15.3.5 const float PictureFpsMin = 0.0F

The minimum fps for the picture.

2.15.4 Property Documentation

2.15.4.1 int AudioCompressionQuality [get], [set]

The function sets and gets audio compression quality. The audio compression quality should be an integer between 0-10, while 10 being the best quality.

2.15.4.2 bool EnableAudio [get], [set]

The function gets and sets whether the audio is enabled or not.

2.15.4.3 bool EnablePicture [get], [set]

The function gets and sets whether the picture is enabled or not.

2.15.4.4 int PictureCompressionQuality [get], [set]

The function gets and sets the picture compression quality. It should be an integer between 0-100, while 100 being the best quality.

2.15.4.5 float PictureFps [get], [set]

The function gets and sets the frames per second to the picture. Values under 1 will result in frame rates with fewer than one frames per second.

2.15.4.6 int PictureResolution [get], [set]

The function gets and sets the picture resolution. The resolution must be an integer between 0-10, while 10 being the largest resolution available.

The documentation for this class was generated from the following file:

MediaConfigurationDto.cs

# 2.16 NotUrgentPage Class Reference

In the page the user will determine what kind of help he needs by choosing the right option from listbox. The user can also navigate to urgent page if he needs to.

Inherits PhoneApplicationPage.

**Public Member Functions** 

NotUrgentPage ()

The function is constructor that does nothing.

2.16.1 Detailed Description

In the page the user will determine what kind of help he needs by choosing the right option from listbox. The user can also navigate to urgent page if he needs to.

<author>Atte Söderlund</author>

#### 2.16.2 Constructor & Destructor Documentation

## 2.16.2.1 NotUrgentPage ( )

The function is constructor that does nothing.

The documentation for this class was generated from the following file:

NotUrgentPage.xaml.cs

# 2.17 PreviewImageCapturer Class Reference

The class is used for capturing preview images from AudioVideoCaptureDevice and compressing them to the jpeg format. It enables the compressed frames to have lower resolution and frame rate than the AudioVideoCapture Device preview buffer generates.

**Public Member Functions** 

PreviewImageCapturer (AudioVideoCaptureDevice source)

The function instantializes a new PreviewImageCapturer using the provided AudioVideoCaptureDevice as it's preview image source. The class will not open the AudioVideoCaptureDevice for capture. Before calling StartCapture, the AudioVideoCaptureDevice has to be opened for video or audiovideo capture.

• void StartCapture ()

The function starts capturing preview buffer frames. Before calling StartCapture, the AudioVideoCaptureDevice has to be opened for video or audiovideo capture.

void StopCapture ()

The function stops listening for the preview frames from the AudioVideoCaptureDevice.

void SetQualityParameters (float fps, int compressionQuality, int compressedWidth, int compressedHeight)

The function sets parameters related to the jpeg compression quality, the frame capture rate and the size of the compressed frame. The captured frame from the preview buffer is scaled to the provided dimensions on the compression.

# **Public Attributes**

JpgFrameCaptured JpgFrameCapturedEvent

The class delegate is for JPG image captured event.

## Properties

• double SourceImageRatio [get]

The function to set and get source image ratio.

• int imageWidth [get]

The images width

• int imageHeight [get] The images height.

## 2.17.1 Detailed Description

The class is used for capturing preview images from AudioVideoCaptureDevice and compressing them to the jpeg format. It enables the compressed frames to have lower resolution and frame rate than the AudioVideoCapture Device preview buffer generates.

<author>Veli-Mikko Puupponen</author> TODO: it should also count the time between the captured frames on the captureDevice\_PreviewFrameAvailable. If the AudioVideoCaptureDevice providing the frames starts slowing down, the problem would be detected and the outgoing frame rate would not drop proportionally.

## 2.17.2 Constructor & Destructor Documentation

#### 2.17.2.1 PreviewImageCapturer ( AudioVideoCaptureDevice source )

The function instantializes a new PreviewImageCapturer using the provided AudioVideoCaptureDevice as it's preview image source. The class will not open the AudioVideoCaptureDevice for capture. Before calling StartCapture, the AudioVideoCaptureDevice has to be opened for video or audiovideo capture.

#### Parameters

source THe AudioVideoCaptureDevice to be used as a source of ARGB preview frames.	
---	--

# 2.17.3 Member Function Documentation

2.17.3.1 void SetQualityParameters (float fps, int compressionQuality, int compressedWidth, int compressedHeight)

The function sets parameters related to the jpeg compression quality, the frame capture rate and the size of the compressed frame. The captured frame from the preview buffer is scaled to the provided dimensions on the compression.

Parameters

fps	The number of frames to be captured per second.
compression⇔	The compression quality for the resulting jpgs is between 0 to 100, where 100 is the best
Quality	quality.
compressed⇔	The width of the compressed image. If it is different from the AudioVideoCaptureDevices
Width	preview width, the image will be scaled on compression
compressed⇔	The height of the compressed image. If it is different from the AudioVideoCaptureDevices
Height	preview height, the image will be scaled on compression

# 2.17.3.2 void StartCapture ( )

The function starts capturing preview buffer frames. Before calling StartCapture, the AudioVideoCaptureDevice has to be opened for video or audiovideo capture.

```
2.17.3.3 void StopCapture ()
```

The function stops listening for the preview frames from the AudioVideoCaptureDevice.

2.17.4 Member Data Documentation

2.17.4.1 JpgFrameCaptured JpgFrameCapturedEvent

The class delegate is for JPG image captured event.

2.17.5 Property Documentation

2.17.5.1 int imageHeight [get]

The images height.

2.17.5.2 int imageWidth [get]

The images width

2.17.5.3 double SourceImageRatio [get]

The function to set and get source image ratio.

The documentation for this class was generated from the following file:

PreviewImageCapturer.cs

# 2.18 Settings Class Reference

The class can store any objects to the application storage. The user have to always save by using save function or the modifications do not take place in the storage.

Public Member Functions

- · Settings ()
  - The function initializes settings.
- void AddOrUpdateValue (string Key, Object value)
  - The function to update the value for the specified setting. If the setting does not exist, then it adds the setting.
- bool RemoveSetting (String key)

The function removes setting using the given key.

- T GetValueOrDefault< T > (string Key, T defaultValue)
  - The function gets the current value of the setting. If the key is not found it returns the default value.
- void Save ()

The function saves all settings to the storage.

# **Public Attributes**

const String GUID = "guid"

The key for a GUID identifier used to specify connection.

const String ISGPSENABLED = "isGpsEnabled"

The key for a object that identifies whether the GPS is enabled or disabled.

const String FONTSIZE = "fontSize"

The key for a object that specifies fontsize used in the application.

const String STREETADDRESS = "streetName"

The key for a object which contains the user specified street address.

- const String POSTALCODE = "postalCode"
  - The key for a object which contains the user specified postal code.
- const String LOCALITY = "locality"

The key for a object which contains the user specified locality.

const String OWNERNAME = "ownerName"

The key for a object which contains the user specified name.

const String PHONENUMBERS = "phoneNumbers"

The key for a objec which contains the user specified phonenumbers.

const String CONNECTION = "connection"

The key for a object which contains the connection address.

const String SELECTEDCULTURE = "selectedCulture"

The key for a object which contains the user specified culture.

## 2.18.1 Detailed Description

The class can store any objects to the application storage. The user have to always save by using save function or the modifications do not take place in the storage.

<author>Atte Söderlund</author> Use keys defined here as puplic consts.

#### 2.18.2 Constructor & Destructor Documentation

2.18.2.1 Settings ( )

The function initializes settings.

2.18.3 Member Function Documentation

2.18.3.1 void AddOrUpdateValue ( string Key, Object value )

The function to update the value for the specified setting. If the setting does not exist, then it adds the setting.

Parameters

Key	The key that is used to identify the settings object.
value	The object that you like to store.

#### 2.18.3.2 T GetValueOrDefault< T > ( string Key, T defaultValue )

The function gets the current value of the setting. If the key is not found it returns the default value.

#### **Template Parameters**

T	The type of the item

#### Parameters

Key	The key that is used to identify the settings object.
defaultValue	The value to be returned if the item is not found.

#### Returns

The setting identified with the key in the specified type.

# 2.18.3.3 bool RemoveSetting (String key)

The function removes setting using the given key.

Parameters

Key	The key that is used to identify the settings object.	
-----	---	--

Returns

True if successful otherwise false.

2.18.3.4 void Save ( )

The function saves all settings to the storage.

2.18.4 Member Data Documentation

2.18.4.1 const String CONNECTION = "connection"

The key for a object which contains the connection address.

2.18.4.2 const String FONTSIZE = "fontSize"

The key for a object that specifies fontsize used in the application.

2.18.4.3 const String GUID = "guid"

The key for a GUID identifier used to specify connection.

2.18.4.4 const String ISGPSENABLED = "isGpsEnabled"

The key for a object that identifies whether the GPS is enabled or disabled.

2.18.4.5 const String LOCALITY = "locality"

The key for a object which contains the user specified locality.

2.18.4.6 const String OWNERNAME = "ownerName"

The key for a object which contains the user specified name.

2.18.4.7 const String PHONENUMBERS = "phoneNumbers"

The key for a objec which contains the user specified phonenumbers.

2.18.4.8 const String POSTALCODE = "postalCode"

The key for a object which contains the user specified postal code.

2.18.4.9 const String SELECTEDCULTURE = "selectedCulture"

The key for a object which contains the user specified culture.

2.18.4.10 const String STREETADDRESS = "streetName"

The key for a object which contains the user specified street address.

The documentation for this class was generated from the following file:

· Settings.cs

# 2.19 SettingsAndTutorialsPage Class Reference

The class is used to view and change the application settings and watch the first aid tutorials. Inherits PhoneApplicationPage.

**Public Types** 

 enum FontSizes {
 Biggest, Bigger, Default, Smaller, Smallest }

A simple enum for font size choice. It is used to specify the font size used in the UI.

# **Public Member Functions**

• SettingsAndTutorialsPage ()

The function initializes the page and loads the settings from Settings class.

## **Protected Member Functions**

override void OnNavigatedTo (NavigationEventArgs e)

The function sets the pivot to the first aid when the page is loaded and if the navigationcontext contains "tofirstaid" string.

2.19.1 Detailed Description

The class is used to view and change the application settings and watch the first aid tutorials.

<author>Atte Söderlund</author>

2.19.2 Member Enumeration Documentation

2.19.2.1 enum FontSizes

A simple enum for font size choice. It is used to specify the font size used in the UI.

2.19.3 Constructor & Destructor Documentation

2.19.3.1 SettingsAndTutorialsPage()

The function initializes the page and loads the settings from Settings class.

2.19.4 Member Function Documentation

2.19.4.1 override void OnNavigatedTo (NavigationEventArgs e) [protected]

The function sets the pivot to the first aid when the page is loaded and if the navigationcontext contains "tofirstaid" string.

The documentation for this class was generated from the following file:

SettingsAndTutorialsPage.xaml.cs

# 2.20 SpeexCompressionHelper Class Reference

The class providies static methods for compression and decompression of speex encoded audio segments.

## **Static Public Member Functions**

static byte[] DecompressSpeex (SpeexDecoder decoder, MediaPacket speexPacket)

The function decompresses the provided MediaPacket of payload data using the provided SpeexDecoder. It returns the resulting PCM samples in a byte array or an empty array.

 static int CompressSpeex (byte[] pcmSegment, int sampleSizeBytes, SpeexEncoder encoder, out byte[] compressed)

The function encodes the provided PCM samples using the provided SpeexEncoder. The resulting encoded bytes are put into the compressed array. It returns the count of 16 bit samples that was used as the input of the speex encoder.

#### 2.20.1 Detailed Description

The class providies static methods for compression and decompression of speex encoded audio segments.

<author>Veli-Mikko Puupponen</author>

# 2.20.2 Member Function Documentation

2.20.2.1 static int CompressSpeex ( byte[] pcmSegment, int sampleSizeBytes, SpeexEncoder encoder, out byte[] compressed ) [static]

The function encodes the provided PCM samples using the provided SpeexEncoder. The resulting encoded bytes are put into the compressed array. It returns the count of 16 bit samples that was used as the input of the speex encoder.

If the count of the PCM samples is not a multiple 320, the differece is padded with silence after the samples.

The SpeexEncoder is assumed to be operating in the BandMode.Wide and the sampleSizeBytes is assumed to be 2, i.e. 16bit PCM.

Parameters

pcmSegment	The 16Bit PCM samples to be encoded into speex.
sampleSize⇔	The size of the PCM samples in bytes should be 2.
Bytes	
encoder	SpeexEncoder used to encode the data.
compressed	The target array for the resulting speex encoded audio.

Returns

The count of 16Bit PCM samples compressed.

2.20.2.2 static byte [] DecompressSpeex ( SpeexDecoder decoder, MediaPacket speexPacket ) [static]

The function decompresses the provided MediaPacket of payload data using the provided SpeexDecoder. It returns the resulting PCM samples in a byte array or an empty array.

The SpeexDecodes is assumed to operate in the BandMode.Wide and the encoded data to conform to this format.

If the decompression fails, returns an empty array.

Parameters

decoder	SpeexDecodes instance used to decode the speex encoding.
speexPacket	MediaPacket containing speex encoded audio as its payload.

Returns

Resulting PCM samples or an empty array.

The documentation for this class was generated from the following file:

SpeexCompressionHelper.cs

# 2.21 UrgentPage Class Reference

In this page the connection happens visualy. When the connection is taken to process by the emergency exchange handler navigates to the Connected page. The user can turn GPS on and toggle NoSound.

Inherits PhoneApplicationPage.

**Public Member Functions** 

• UrgentPage ()

The function shows the GPS switch If GPS is not enabled. It creates a new connection in to an own thread so the UI thread does not get blocked.

#### Protected Member Functions

override void OnNavigatedFrom (NavigationEventArgs e)

The function disconnects from the server and then removes the event handlers and sets the connection to the App class.

override void OnBackKeyPress (System.ComponentModel.CancelEventArgs e)

The function cancels the connection with the server.

override void OnNavigatedTo (NavigationEventArgs e)

If navigationservice url contains "NotUrgentPage", it changes the connection priority to not urgent.

#### 2.21.1 Detailed Description

In this page the connection happens visualy. When the connection is taken to process by the emergency exchange handler navigates to the Connected page. The user can turn GPS on and toggle NoSound.

<author>Atte Söderlund</author>

2.21.2 Constructor & Destructor Documentation

2.21.2.1 UrgentPage ()

The function shows the GPS switch If GPS is not enabled. It creates a new connection in to an own thread so the UI thread does not get blocked.

2.21.3 Member Function Documentation

2.21.3.1 override void OnBackKeyPress ( System.ComponentModel.CancelEventArgs e ) [protected]

The function cancels the connection with the server.

2.21.3.2 override void OnNavigatedFrom (NavigationEventArgs e) [protected]

The function disconnects from the server and then removes the event handlers and sets the connection to the App class.

2.21.3.3 override void OnNavigatedTo (NavigationEventArgs e) [protected]

If navigationservice url contains "NotUrgentPage", it changes the connection priority to not urgent.

The documentation for this class was generated from the following file:

UrgentPage.xaml.cs

# 2.22 WavAudioSampleCapturer Class Reference

The class is used for capturing PCM audio fragments from XNA microphone and processing them as WAVE files.

**Public Member Functions** 

WavAudioSampleCapturer (Microphone mic)

The function instantializes a new WavAudioSampleCapturer using the provided Microphone instance. The Microphone must be new and initalized, but not in use or started.

• void StartCapture ()

The function stars capturing from the XNA Microphone. FrameworkDispatcher.Update needs to have been called at least once prior to the operation and needs to be called reqularly during the recording.

• void StopCapture ()

The function stops the capturing from the XNA Microphone.

void UpdateSampleParameters (int sampleLengthMilliSeconds)

The function sets the length of the fragments to be captured from the Microphone. A valid range is 50 to 1000 milliseconds.

# **Public Attributes**

WaveSegmentCaptured WaveSegmentCapturedEvent

The function event handler for wave segment captured event.

PcmSegmentCaptured PcmSegmentCapturedEvent
 The function event handler for pcm segment captured event.

#### **Properties**

int AudioSampleMilliSeconds [get]

The function returns the milliseconds of the audio sample as int.

• int SampleRate [get]

The function returns the sample rate as int.

• int SampleSizeBytes [get] The function returns the sample bytes size as int.

#### 2.22.1 Detailed Description

The class is used for capturing PCM audio fragments from XNA microphone and processing them as WAVE files.

<author>Veli-Mikko Puupponen</author>

# 2.22.2 Constructor & Destructor Documentation

```
2.22.2.1 WavAudioSampleCapturer (Microphone mic)
```

The function instantializes a new WavAudioSampleCapturer using the provided Microphone instance. The Microphone must be new and initialized, but not in use or started.

## Parameters

*mic* The microphone to use.

2.22.3 Member Function Documentation

```
2.22.3.1 void StartCapture ()
```

The function stars capturing from the XNA Microphone. FrameworkDispatcher.Update needs to have been called at least once prior to the operation and needs to be called regularly during the recording.

```
2.22.3.2 void StopCapture ( )
```

The function stops the capturing from the XNA Microphone.

2.22.3.3 void UpdateSampleParameters ( int sampleLengthMilliSeconds )

The function sets the length of the fragments to be captured from the Microphone. A valid range is 50 to 1000 milliseconds.

Parameters

sampleLength⇔	The length of the single captured audio fragment in milliseconds.
MilliSeconds	

2.22.4 Member Data Documentation

2.22.4.1 PcmSegmentCaptured PcmSegmentCapturedEvent

The function event handler for pcm segment captured event.

2.22.4.2 WaveSegmentCaptured WaveSegmentCapturedEvent

The function event handler for wave segment captured event.

2.22.5 Property Documentation

2.22.5.1 int AudioSampleMilliSeconds [get]

The function returns the milliseconds of the audio sample as int.

2.22.5.2 int SampleRate [get]

The function returns the sample rate as int.

2.22.5.3 int SampleSizeBytes [get]

The function returns the sample bytes size as int.

The documentation for this class was generated from the following file:

WavAudioSampleCapturer.cs

## 2.23 WaveFileHeader Class Reference

The class generates canonical WAVE header to an array of PCM samples.

**Static Public Member Functions** 

static byte[] AddWaveHeader (byte[] pcm, int bytesPerSample, int channels, int sampleRate)
 The function generates the WAVE header for the provided PCM sample data. The length of the returned data array is length + 44. The length of the PCM data is limited by the limits of the WAVE header.

## 2.23.1 Detailed Description

The class generates canonical WAVE header to an array of PCM samples.

<author>Veli-Mikko Puupponen</author>

2.23.2 Member Function Documentation

2.23.2.1 static byte [] AddWaveHeader ( byte[] pcm, int bytesPerSample, int channels, int sampleRate ) [static]

The function generates the WAVE header for the provided PCM sample data. The length of the returned data array is length + 44. The length of the PCM data is limited by the limits of the WAVE header.

# Parameters

рст	The array of complete PCM samples with no compression.
bytesPerSample	The number of bytes (bits/8) per sample.
channels	The hannels in the PCM data 1 for mono and 2 for stereo.
sampleRate	The sampling rate of the data.

# Returns

The provided data with wave header.

The documentation for this class was generated from the following file:

WaveFileHeader.cs