
MOST Web Documentation

Release 1.0.1

Healthcare Flows - CRS4

February 08, 2017

1	Visualization Framework	3
1.1	Android Most Visualization API	3
1.2	Examples	13
1.3	License	14
1.4	Authors	14
1.5	Indices and tables	14

The MOST project aims to achieve an open, modular and scalable solution for the creation, execution and management of remote clinical consultations with direct interaction between specialists.

The project consists of a set of frameworks that deal with different aspects and technologies useful for the creation of telemedicine applications.

Visualization Framework

The Visualization Framework allows you to include into your mobile applications visual and interactive widgets capable of rendering, inspecting and changing properties of IStream objects. This framework internally uses the [Android Touch Gallery Library](https://github.com/Dreddik/AndroidTouchGallery) (<https://github.com/Dreddik/AndroidTouchGallery>) for rendering images and the [MOST Streaming Library](https://github.com/crs4/most-streaming) (<https://github.com/crs4/most-streaming>) for managing audio/video streams.

TREE:

1.1 Android Most Visualization API

Contents:

1.1.1 Javadoc

org.crs4.most.visualization

IPtzCommandReceiver

public interface **IPtzCommandReceiver**

An activity must implement this interface to be able to receive notifications from the attached PTZ_ControllerFragment or PTZ_PopupWindow

Methods

onGoHome

public void **onGoHome** ()

Called when the user clicks on the home button of the pan-tilt panel

onPTZstartMove

public void **onPTZstartMove** (PTZ_Direction *dir*)

Called when the user presses one button of the pan-tilt panel

Parameters

- **dir** – the required moving direction

onPTZstartZoom

public void **onPTZstartZoom** (PTZ_Zoom *dir*)

Called when the user presses one button of the zoom panel

Parameters

- **dir** – the required zooming direction

onPTZstopMove

public void **onPTZstopMove** (PTZ_Direction *dir*)

Called when the user releases one button of the pan-tilt panel

Parameters

- **the** – moving direction before this stop command

onPTZstopZoom

public void **onPTZstopZoom** (PTZ_Zoom *dir*)

Called when the user releases one button of the zoom panel

Parameters

- **the** – zooming direction before this stop command

onSnaphot

public void **onSnaphot** ()

Called when the user clicks on the snapshot button

IStreamArrayAdapter

class **IStreamArrayAdapter** extends ArrayAdapter<IStream>

This adapter is internally used from the *StreamInspectorFragment* for representing IStream data.

Constructors

IStreamArrayAdapter

public **IStreamArrayAdapter** (Context *context*, int *viewId*, [List](http://docs.oracle.com/javase/6/docs/api/java/util/List.html) (<http://docs.oracle.com/javase/6/docs/api/java/util/List.html>)<IStream> *objects*, [List](http://docs.oracle.com/javase/6/docs/api/java/util/List.html) (<http://docs.oracle.com/javase/6/docs/api/java/util/List.html>)<StreamProperty> *streamProperties*)

This adapter provides a way of rendering informations about a list of IStream objects.

Parameters

- **context** –
- **viewId** – the view id where to render the informations about each stream
- **objects** – the list of IStream objects.
- **streamProperties** – the properties to render for each stream (a null value renders all the available properties)

Methods

getView

public View **getView** (int *position*, View *convertView*, ViewGroup *parent*)

IStreamFragmentCommandListener

public interface **IStreamFragmentCommandListener**

This interface must be implemented by activities attached to the *StreamViewerFragment* fragment

Methods**onPause**

public void **onPause** ([String](http://docs.oracle.com/javase/6/docs/api/java/lang/String.html) *streamId*)

Callback triggered when the user clicks on the pause button

Parameters

- **streamId** – the id of the stream the *StreamFragment* refer to

onPlay

public void **onPlay** ([String](http://docs.oracle.com/javase/6/docs/api/java/lang/String.html) *streamId*)

Callback triggered after the user clicked on the play button

Parameters

- **streamId** – the id of the stream the *StreamViewerFragment* refer to

onSurfaceViewCreated

public void **onSurfaceViewCreated** ([String](http://docs.oracle.com/javase/6/docs/api/java/lang/String.html) *streamId*, SurfaceView *surfaceView*)

Callback triggered once the surfaceView of the fragment became available

Parameters

- **streamId** – the id of the stream the *StreamViewerFragment* refer to
- **surfaceView** – the surfaceView where to render the stream

onSurfaceViewDestroyed

public void **onSurfaceViewDestroyed** ([String](http://docs.oracle.com/javase/6/docs/api/java/lang/String.html) *streamId*)

Callback triggered after the surfaceView of this fragment has been destroyed

Parameters

- **streamId** – the id of the stream the *StreamViewerFragment* refer to

PTZ_ControllerFragment

public class **PTZ_ControllerFragment** extends DialogFragment implements OnTouchListener

This fragment provides you a set of visual panels and buttons to be used as a GUI frontend for handling remote PTZ webcams. This fragment expects the attached activity implements the *IptzCommandReceiver* interface, because it notifies to this interface all the GUI actions (e.g button clicks)

Methods

newInstance

public static *PTZ_ControllerFragment* **newInstance** ()
 Provides a new instance of this fragment, with all panels visible

Returns the PTZ_ControllerFragment instance

newInstance

public static *PTZ_ControllerFragment* **newInstance** (boolean *panTiltPanelVisible*, boolean *zoomPanelVisible*, boolean *snapshotVisible*)
 Provides a new instance of this fragment, with a selection of desired panels

Parameters

- **panTiltPanelVisible** – set the pan-tilt panel visible or not
- **zoomPanelVisible** – set the zoom panel visible or not
- **snapshotVisible** – set the snapshot button visible or not

onAttach

public void **onAttach** (Activity *activity*)

onCreateView

public View **onCreateView** (LayoutInflater *inflater*, ViewGroup *container*, Bundle *savedInstanceState*)

onTouch

public boolean **onTouch** (View *v*, MotionEvent *event*)

PTZ_ControllerPopupWindowFactory

public class **PTZ_ControllerPopupWindowFactory** implements OnTouchListener

This Factory class provides you an interactive visual panel containing a set of buttons to be used as a GUI frontend for handling remote PTZ webcams. You need to pass a *IPtzCommandReceiver* interface to the factory method of this class, because it notifies to this interface all the GUI actions (e.g button clicks) Note that the created window implements the `android.view.View.OnTouchListener` interface, so you can move it to the desired position on the screen.

Constructors

PTZ_ControllerPopupWindowFactory

public **PTZ_ControllerPopupWindowFactory** (Context *context*, *IPtzCommandReceiver* *ptzReceiver*, boolean *panTiltPanelVisible*, boolean *zoomPanelVisible*, boolean *snapshotVisible*, int *xPos*, int *yPos*)

Creates a new floating popupWindow, containing a set of optional panels to be viewed

Parameters

- **context** – the context where to render the popup Window
- **ptzReceiver** – the remote object to use as the target of all user notifications

- **panTiltPanelVisible** – set the pan-tilt panel visible or not
- **zoomPanelVisible** – set the zoom panel visible or not
- **snapshotVisible** – set the snapshot button visible or not
- **xPos** – the initial X position of the popupWindow
- **yPos** – the initial y position of the popupWindow

Methods

getPopupWindow

public PopupWindow **getPopupWindow** ()

Returns the created popup Window

onTouch

public boolean **onTouch** (View *v*, MotionEvent *event*)

show

public void **show** ()

Show the popupWindow at the current location

StreamInspectorFragment

public class **StreamInspectorFragment** extends Fragment

This fragment provides a way for visually getting and/or updating the video properties of a list of *IStream* objects. Also, you can specify a filter for getting only a subset of stream properties you are interested in. You can attach this fragment to any activity, provided that it implements the *StreamInspectorFragment.IStreamProvider* interface.

Methods

newInstance

public static *StreamInspectorFragment* **newInstance** ()

Provides a new instance of this fragment

Returns the StreamInspectorFragment instance

onActivityCreated

public void **onActivityCreated** (Bundle *bundle*)

onAttach

public void **onAttach** (Activity *activity*)

onCreateView

public View **onCreateView** (LayoutInflater *inflater*, ViewGroup *container*, Bundle *savedInstanceState*)

refreshData

public void **refreshData** ()

Force the reloading of the stream data of the underlying adapter

updateStreamStateInfo

public void **updateStreamStateInfo** (IStream *stream*)

This method would be called for notifying the StreamInspectorFragment that one or more properties of the IStream (specified as argument) has been changed,

Parameters

- **stream** – the modified IStream object

StreamInspectorFragment.IStreamProvider

public interface **IStreamProvider**

This interface is used by the StreamInspector for getting the streams to inspect along with their properties.

Methods

getStreamProperties

public [List](http://docs.oracle.com/javase/6/docs/api/java/util/List.html) (http://docs.oracle.com/javase/6/docs/api/java/util/List.html)<StreamProperty> **getStreamProperties** ()

Provide the list of properties to show for each stream (a null value shows all properties)

getStreams

public [List](http://docs.oracle.com/javase/6/docs/api/java/util/List.html) (http://docs.oracle.com/javase/6/docs/api/java/util/List.html)<IStream> **getStreams** ()

Provides the list of the streams to show in the inspector

Returns the list of the streams to inspect

StreamViewerFragment

public class **StreamViewerFragment** extends Fragment

This fragment represents a visual container for an IStream. It can be attached to any Activity, provided that it implements the *IStreamFragmentCommandListener* interface. This fragment contains a surface where to render the stream along with two image buttons that you can optionally use for sending play or pause stream requests to the attached activity

Fields

FRAGMENT_STREAM_ID_KEY

public static final [String](http://docs.oracle.com/javase/6/docs/api/java/lang/String.html) (http://docs.oracle.com/javase/6/docs/api/java/lang/String.html) **FRAGMENT_STREAM_ID_KEY**

Methods

newInstance

public static *StreamViewerFragment* **newInstance** (*String* (<http://docs.oracle.com/javase/6/docs/api/java/lang/String.html>) *streamId*)

Intances a new *StreamViewerFragment*

Parameters

- **streamId** – the id of the stream to render

Returns a new *StreamViewerFragment* instance

onActivityCreated

public void **onActivityCreated** (*Bundle bundle*)

onAttach

public void **onAttach** (*Activity activity*)

onCreate

public void **onCreate** (*Bundle savedInstanceState*)

onCreateView

public *View* **onCreateView** (*LayoutInflater inflater*, *ViewGroup container*, *Bundle savedInstanceState*)

onDetach

public void **onDetach** ()

setPlayerButtonsVisible

public void **setPlayerButtonsVisible** (*boolean value*)

Set the player buttons visible or not

Parameters

- **value** – `true` set buttons visible; `false` invisible.

org.crs4.most.visualization.image_gallery**EclairMotionEvent**

public class **EclairMotionEvent** extends *WrapMotionEvent*

Constructors**EclairMotionEvent**

protected **EclairMotionEvent** (*MotionEvent event*)

Methods**getPointerCount**

public int **getPointerCount** ()

getPointerId

public int **getPointerId** (int *pointerIndex*)

getX

public float **getX** (int *pointerIndex*)

getY

public float **getY** (int *pointerIndex*)

ImageGalleryFragment

public class **ImageGalleryFragment** extends Fragment

This fragment allows you to embed in your activity an image gallery.

Fields

imageView

LinearLayout **imageView**

Methods

onActivityCreated

public void **onActivityCreated** (Bundle *savedInstanceState*)

onCreateView

public View **onCreateView** (LayoutInflater *inflater*, ViewGroup *container*, Bundle *savedInstanceState*)

Called when the activity is first created.

reloadGalleryImages

public void **reloadGalleryImages** ()

Reloads the images contained in to the internal storage

selectImage

public void **selectImage** (int *imageIndex*)

Select an image from the gallery, by index array

Parameters

- **imageIndex** – the index of the image (the index 0 is the newest image)

ImageGalleryFragment.ImageAdapter

public class **ImageAdapter** extends BaseAdapter

Fields

imageBackground
int **imageBackground**

Constructors

ImageAdapter
public **ImageAdapter** (Context *c*)

Methods

getCount
public int **getCount** ()

getItem
public [Object](http://docs.oracle.com/javase/6/docs/api/java/lang/Object.html) (<http://docs.oracle.com/javase/6/docs/api/java/lang/Object.html>) **getItem** (int *arg0*)

getItemId
public long **getItemId** (int *arg0*)

getView
public View **getView** (int *arg0*, View *arg1*, ViewGroup *arg2*)

TouchImageView

public class **TouchImageView** extends ImageView

Fields

DRAG
static final int **DRAG**

NONE
static final int **NONE**

ZOOM
static final int **ZOOM**

context
Context **context**

matrix
Matrix **matrix**

mid

PointF **mid**

mode

int **mode**

oldDist

float **oldDist**

savedMatrix

Matrix **savedMatrix**

start

PointF **start**

Constructors

TouchImageView

public **TouchImageView** (Context *context*, GestureDetector *gestureDetector*)

Methods

onTouchEvent

public boolean **onTouchEvent** (MotionEvent *event*)

setImage

public void **setImage** (Bitmap *bm*, int *displayWidth*, int *displayHeight*)

WrapMotionEvent

public class **WrapMotionEvent**

Fields

event

protected MotionEvent **event**

Constructors

WrapMotionEvent

protected **WrapMotionEvent** (MotionEvent *event*)

Methods

getAction

```
public int getAction ()
```

getPointerCount

```
public int getPointerCount ()
```

getPointerId

```
public int getPointerId (int pointerIndex)
```

getX

```
public float getX ()
```

getX

```
public float getX (int pointerIndex)
```

getY

```
public float getY ()
```

getY

```
public float getY (int pointerIndex)
```

wrap

```
public static WrapMotionEvent wrap (MotionEvent event)
```

1.2 Examples

1.2.1 Android

All the following examples are located into the folder *examples/android*.

- **StreamInspectorExample** This example explains you:
 - how to play a stream on a *StreamViewerFragment*
 - how to get and/or update the properties of the stream by using a *StreamInspectorFragment*
 - **how to change the Stream Rendering mode. You can choose among the following 3 modalities:**
 - rstp streamning (the continous stream that the user can play or pause by clicking the player buttons provided by the StreamViewer fragment)
 - still-image (you load a snapshot from the renmote camera by clicking on a button)
 - timed still-images (the system loads a jpeg image from the remote camera every # seconds, as specified by the user)
- **PTZ_ControllerExample** This example explains you:
 - how to play a stream on a *StreamViewerFragment*
 - how to get and/or update the properties of the stream by using a *StreamInspectorFragment*

- how to remotely control pan, tilt and zoom values of an Axis PTZ Webcam by using a *PTZ_ControllerFragment*
- how to make snapshots of the stream and save them into the internal storage
- **PTZ_ImageGalleryExample** This example contains all the features of the *PTZ_ControllerExample* example, and in addition, explains you:
- how to open an Image Gallery containing all the stream snapshots, by using a *ImageGalleryFragment*
- how to select an image from the gallery, zoom in/out and move it by touch screen gestures
- how to delete an image from the gallery (simply by a double tap on it)

For running the Android examples, open your preferred IDE (e.g Eclipse) and do the following changes:

- Import the Most-Streaming project library
- Edit the file *jni/Android.mk* and properly change the absolute path of the environment variables `GSTREAMER_SDK_ROOT_ANDROID` and `GSTREAMER_ROOT`
- Import the Android project example located from the *android/examples* folder and add the Most-Streaming and the Most-Visualization projects both as Library and project references
- Create your *uri.properties.default* property file and put it into the *assets* folder. (That folder already contains the *uri.properties* file that you can use as template for your own property file)
- Build the projects (Note that the NDK must be installed and configured on your system in order to build the project)
- Deploy the application on an android device or emulator

1.3 License

```
/*!  
 * Project MOST - Moving Outcomes to Standard Telemedicine Practice  
 * http://most.crs4.it/  
 *  
 * Copyright 2014-15, CRS4 srl. (http://www.crs4.it/)  
 * Dual licensed under the MIT or GPL Version 2 licenses.  
 * See license-GPLv2.txt or license-MIT.txt  
 */
```

GPL2: <https://www.gnu.org/licenses/gpl-2.0.txt>

MIT: <http://opensource.org/licenses/MIT>

1.4 Authors

Code author: Francesco Cabras <francesco.cabras@crs4.it>

Code author: Stefano Leone Monni <stefano.monni@crs4.it>

1.5 Indices and tables

- [genindex](#)

- [modindex](#)
- [search](#)

C

context (Java field), 11

D

DRAG (Java field), 11

E

EclairMotionEvent (Java class), 9

EclairMotionEvent(MotionEvent) (Java constructor), 9

event (Java field), 12

F

FRAGMENT_STREAM_ID_KEY (Java field), 8

G

getAction() (Java method), 13

getCount() (Java method), 11

getItem(int) (Java method), 11

getItemId(int) (Java method), 11

getPointerCount() (Java method), 9, 13

getPointerId(int) (Java method), 10, 13

getPopupWindow() (Java method), 7

getStreamProperties() (Java method), 8

getStreams() (Java method), 8

getView(int, View, ViewGroup) (Java method), 5, 11

getX() (Java method), 13

getX(int) (Java method), 10, 13

getY() (Java method), 13

getY(int) (Java method), 10, 13

I

ImageAdapter (Java class), 10

ImageAdapter(Context) (Java constructor), 11

imageBackground (Java field), 11

ImageGalleryFragment (Java class), 10

imageView (Java field), 10

IPtzCommandReceiver (Java interface), 3

IStreamArrayAdapter (Java class), 4

IStreamArrayAdapter(Context, int, List, List) (Java constructor), 4

IStreamFragmentCommandListener (Java interface), 5

IStreamProvider (Java interface), 8

M

matrix (Java field), 11

mid (Java field), 12

mode (Java field), 12

N

newInstance() (Java method), 6, 7

newInstance(boolean, boolean, boolean) (Java method), 6

newInstance(String) (Java method), 9

NONE (Java field), 11

O

oldDist (Java field), 12

onActivityCreated(Bundle) (Java method), 7, 9, 10

onAttach(Activity) (Java method), 6, 7, 9

onCreate(Bundle) (Java method), 9

onCreateView(LayoutInflater, ViewGroup, Bundle) (Java method), 6, 7, 9, 10

onDetach() (Java method), 9

onGoHome() (Java method), 3

onPause(String) (Java method), 5

onPlay(String) (Java method), 5

onPTZstartMove(PTZ_Direction) (Java method), 3

onPTZstartZoom(PTZ_Zoom) (Java method), 4

onPTZstopMove(PTZ_Direction) (Java method), 4

onPTZstopZoom(PTZ_Zoom) (Java method), 4

onSnapshot() (Java method), 4

onSurfaceViewCreated(String, SurfaceView) (Java method), 5

onSurfaceViewDestroyed(String) (Java method), 5

onTouch(View, MotionEvent) (Java method), 6, 7

onTouchEvent(MotionEvent) (Java method), 12

org.crs4.most.visualization (package), 3

org.crs4.most.visualization.image_gallery (package), 9

P

PTZ_ControllerFragment (Java class), 5

PTZ_ControllerPopupWindowFactory (Java class), [6](#)
PTZ_ControllerPopupWindowFactory(Context, IPTz-
CommandReceiver, boolean, boolean, boolean,
int, int) (Java constructor), [6](#)

R

refreshData() (Java method), [8](#)
reloadGalleryImages() (Java method), [10](#)

S

savedMatrix (Java field), [12](#)
selectImage(int) (Java method), [10](#)
setImage(Bitmap, int, int) (Java method), [12](#)
setPlayerButtonsVisible(boolean) (Java method), [9](#)
show() (Java method), [7](#)
start (Java field), [12](#)
StreamInspectorFragment (Java class), [7](#)
StreamViewerFragment (Java class), [8](#)

T

TouchImageView (Java class), [11](#)
TouchImageView(Context, GestureDetector) (Java con-
structor), [12](#)

U

updateStreamStateInfo(IStream) (Java method), [8](#)

W

wrap(MotionEvent) (Java method), [13](#)
WrapMotionEvent (Java class), [12](#)
WrapMotionEvent(MotionEvent) (Java constructor), [12](#)

Z

ZOOM (Java field), [11](#)