

M2M AR 協作之 Android WebRTC 及繪圖
SDK
使用者手冊

中華電信研究院
中華民國106年8月製作

M2M AR 協作之 Android WebRTC 及繪圖 SDK 使用者手冊

本冊文件非經本院同意，不得翻印

發行機構：中華電信研究院

製作者：匯流服務研究所 843 計畫

楊子文、陳澤民

維護者：匯流服務研究所 843 計畫 陳澤民

核准者：匯流服務研究所 843 計畫 楊子文

文件版本修正履歷表

版本	修改日期	修訂內容	修改人員
1	2017/08/31	1. 新增文件	陳澤民

目 錄

文件版本修正履歷表	II
1 SDK 使用說明	1
1.1 APP PROJECT 需取得的 ANDROID 使用者權限.....	1
1.2 APP 需取得的 AR 協作平台使用者戳記.....	1
1.3 在 APP PROJECT 加入 SDK 檔案.....	2
1.4 在 APP 啟動 WRLSIGNALSERVICE	3
1.5 在 APP 程式使用 SDK.....	5
1.5.1 使用 WebRTC SDK 的方式.....	5
1.5.2 使用繪圖 SDK 的方式.....	14
1.6 在 ACTIVITY 的 LAYOUT 各畫面的疊置	17

1 SDK 使用說明

1.1 App project 需取得的 Android 使用者權限

```
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

1.2 App 需取得的 AR 協作平台使用者戳記

1.開發區驗證授權服務位址

<https://asstar-qa.hinet.net/VideoRoom/webservice/userlogin>

2.驗證授權輸入參數說明

參數	必備參數	描述	備註
op	◎	表示驗證類型	op 固定為 Validation
identifier	◎	iOS: bundle id Android:package name	
id	◎	認證帳號	
password	◎	認證密碼	
Example:			

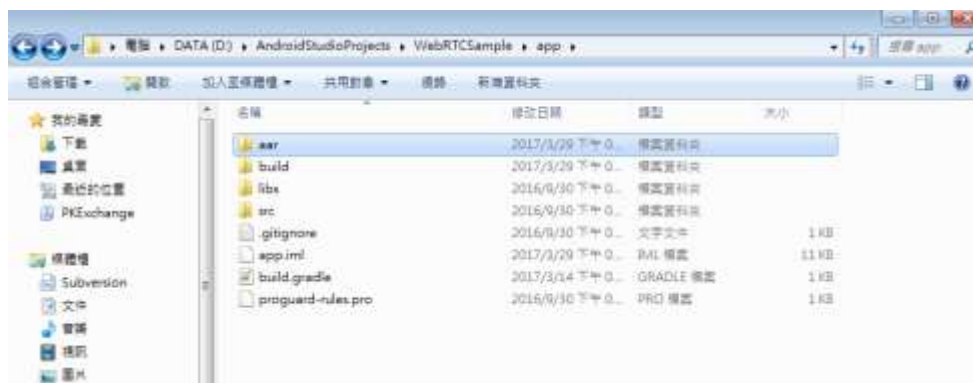
3.驗證授權查詢結果說明

屬性	必備屬性	屬性格式	描述	備註
status_code	◎	數字	驗證授權結果代碼	status_code:1000 驗證成功 status_code:2000 驗證失敗，id 不存在 status_code:3000 驗證失

				敗，password 不對
status_msg		字串	驗證授權結果訊息	status_code>1000 會有訊息
session_id		字串	驗證成功後回傳 session_id	認證 token
general_interval		字串	一般作業定位回傳 區間	單位:分
emergence_inte rvel		字串	緊急作業定位回傳 區間	單位:分
<p>回傳 JSON 範例如下：</p> <pre>{ "status_code": "1000", "status_msg": "XXX 您好，登入成功", "session_id": "ewq7232eweqw32w", "general_interval": "5", "emergence_interval": "1" }</pre>				

1.3 在 App project 加入 SDK 檔案

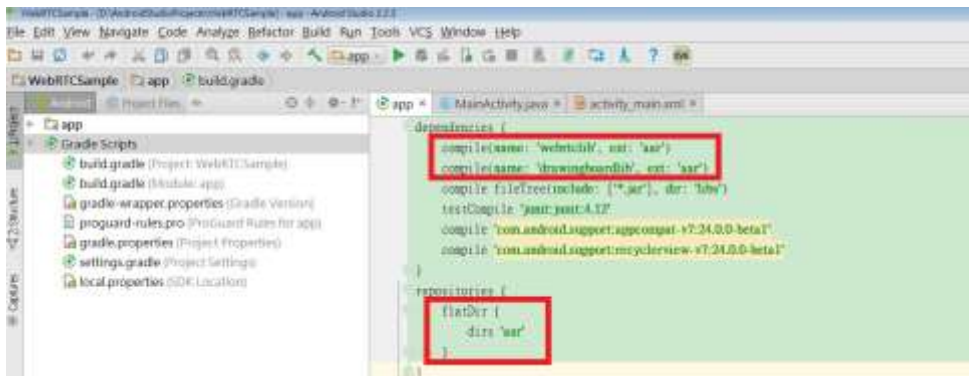
1. 在 project 新增一個 aar 目錄如下圖：



2. 複製本 SDK aar 檔案至此目錄如下圖：



3. 在 project app 的 build.gradle 加入如下圖中紅框的設定：



1.4 在 APP 啟動 WRLSignalService

一. 啟動 WRLSignalService

使用 startService(intent) 呼叫，Intent extended data 定義如下：

名稱	說明
UserId	使用者帳號
TokenName	儲存在 SharedPreferences 的名稱
LoginActionName	Token 無效要呼叫的登入 activity 之 action name
CalleeActionName	有人 call 時要呼叫的受話端 activity 之 action name
ServerUrl	AR 協作平台 Server 的 Url
PreferencesName	SharedPreferences 的名稱
ConnectIcon	WRLSignalService 連線的通知圖式
DisconnectIcon	WRLSignalService 未連線的通知圖式

範列如下：

```
Intent intent = new Intent(LoginActivity.this, WRLSignalService.class);
intent.putExtra("UserId", "cht_007");
intent.putExtra("TokenName", "AsstARToken");
intent.putExtra("LoginActionName", " com.cht.assistantar.qa.login");
intent.putExtra("CalleeActionName", " com.cht.assistantar.qa.callee");
```

```

intent.putExtra("ServerUrl", "https://asstar-qa.hinet.net/AsstARWeb/");
intent.putExtra("PreferencesName", "com.cht.tlcsl.webrtcsample.preferences");
intent.putExtra("ConnectIcon", R.drawable.start);
intent.putExtra("DisconnectIcon", R.drawable.stop);
startService(intent);

```

二. 有人 call 時 WRLSignalService 會呼叫受話端 Activity

WRLSignalService 呼叫受話端 Activity 所使用的 Intent extended data 定義如下：

名稱	說明
CallerId	呼叫者的帳號
CallerName	呼叫者的姓名
ScreenType	呼叫者影像畫面方向，P 指 caller 為直式，L 指 caller 為橫式

受話端 Activity 收到 Intent 後顯示來電及鈴聲，並詢問使用者是否接聽，如接聽再依呼叫者影像畫面方向開啟對應 layout 的受話端視訊 Activity。

在等待使用者回應期間，可向 WRLSignalService 代理程式 WRLSignalAgent，註冊 Listen 對方先掛斷之 Signal 訊息，方法如下：

1. 先 implement WRLHangUpListener interface

```

private class HangUpListener implements WRLHangUpListener {
    @Override
    public void onRemoteHangUp(String RemoteId) {
        //寫收到 HangUp 要做的事
    }
}

```

2. 再 new 一個 WRLSignalAgent 物件，並用 setWRLHangUpListener(Context inContext, WRLHangUpListener inWRLHangUpListener) 方法將 WRLHangUpListener 設定給 WRLSignalAgent，即可 Listen 對方先掛斷之 Signal 訊息

```

mWRLSignalAgent = new WRLSignalAgent();
mWRLSignalAgent.setWRLHangUpListener(this, new HangUpListener());

```

3. 不再需要 Listen 請呼叫 removeWRLHangUpListener(Context inContext) 方法 remove listener

```

mWRLSignalAgent.removeWRLHangUpListener(this);

```


三. 關閉 WRLSignalService

呼叫 M2MFragment 之 static function stopWRLSignalService 如下：

```
static public void stopWRLSignalService(Context inContext)
```

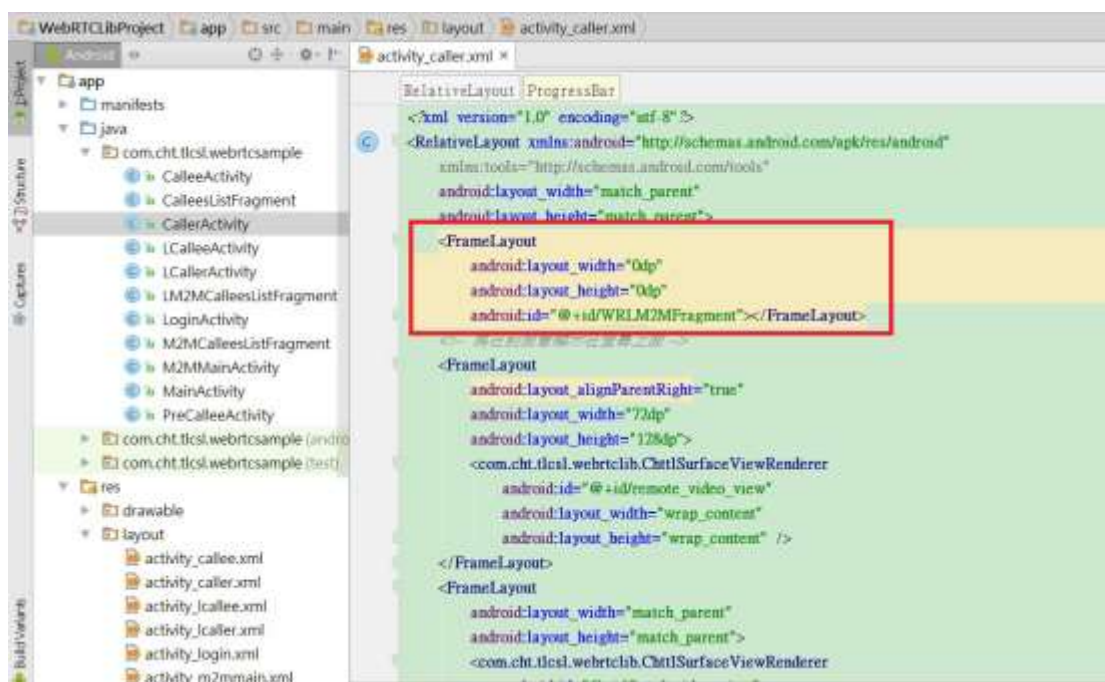
參數 inContext 為 Application Context，範例如下：

```
M2MFragment.stopWRLSignalService(getApplicationContext());
```

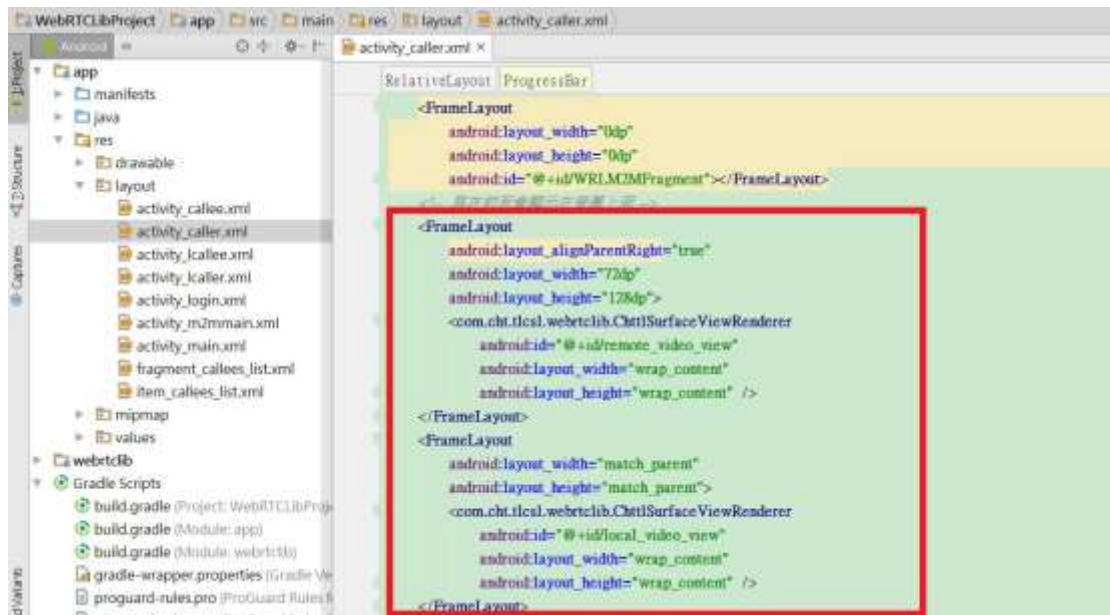
1.5 在 App 程式使用 SDK

1.5.1 使用 WebRTC SDK 的方式

1.在 App 程式欲使用本 SDK 的 activity 之 layout 設定檔案加入空的 FrameLayout，如下圖中紅框的設定：



2.在此 activity 之 layout 欲顯示本機端影像及遠端影像的畫面位置加入含 ChttlSurfaceViewRenderer 的 FrameLayout，如下圖中紅框的設定：



3. 在 activity 程式中加入 M2MFragment 片段：

```
//加入 fragment
mM2MFragment = M2MFragment.newInstance(userId, userToken,
    serverPreUrl);
getFragmentManager().beginTransaction()
    .replace(R.id.WRLM2MFragment,
        mM2MFragment)
    .commit();
```

其中 userToken 為使用者登入系統回覆之戳記，serverPreUrl 為登入系統之前段 URL。

4. 加入 M2MFragment 片段還需要 implements WRLMessageListener，並覆載(Override)以下方法(Method)，並對應到 App 程式欲呈現這些資訊的方式：

4.1 從 WebRTC 接收到訊息之 Callback：

```
@Override
public void onReceiveWRLMessage(int MessageType, String MessageString)
    //MessageType=1 MessageString 為 Text Message
```

//MessageType=2 為白板 image file，MessageString 為白板 image file
 所在位置

//MessageType=4 為凍結 image file，MessageString 為凍結 image file
 所在位置

//MessageType=5 為白板清除 Message，MessageString 固定為 CLEAR
 BOARD

//MessageType=6 為解凍 Message，MessageString 固定為 UNFREEZE

//MessageType=7 為其它檔，MessageString 為其它檔所在位置

4.2 從 WebRTC 接收到已連線之 Callback：

@Override

```
public void onConnect(String inCalleeSeq)
```

//例如可在此改變 Call Button 由不可按變為可掛斷

4.3 從 WebRTC 接收到已斷線之 Callback：

@Override

```
public void onDisconnect()
```

//例如可在此改變 Call Button 變為可撥打

4.4 從 WebRTC 接收到錯誤之 Callback：

@Override

```
public void onError(int errorType, String descriptionString)
```

//ErrorType=11 Failed to create peer connection

//ErrorType=12 Failed to open camera

//ErrorType=13 Failed to add stream

//ErrorType=14 Failed to create SDP

//ErrorType=15 Failed to create ice connection

//ErrorType=16 Failed to set SDP

//ErrorType=17 Failed to initializeAndroidGlobals

//ErrorType=21 Failed to connect WebSocket
 //ErrorType=22 Failed to register WebSocket
 //ErrorType=23 Failed to send WebSocket
 //ErrorType=24 Failed to send PostServer
 //ErrorType=25 無法連接訊息伺服器
 //ErrorType=26 與遠端失去連線
 //ErrorType=31 Failed to parse RoomParameters
 //ErrorType=32 onWebSocketError
 //ErrorType=33 not initiator receive answer
 //ErrorType=34 initiator receive offer
 //ErrorType=35 Unexpected WebSocket message
 //ErrorType=36 WebSocket message JSON parsing error
 //ErrorType=37 WebSocket non connected
 //ErrorType=38 WebSocket send error
 //ErrorType=39 Loopback room is busy
 //ErrorType=101 Permissions not granted
 //ErrorType=103 Failed to connect ARWORK API
 //ErrorType=104 Failed to parse ARWORK API JSON
 //ErrorType=105 使用者取消連線
 //ErrorType=107 對方取消連線
 //ErrorType=108 對方忙線中
 //ErrorType=109 重覆呼叫 SDK

5. M2MFragment 片段所提供的方法如下：

5.1 caller

`void caller(String inCallee, int inLocalVideo, int inRemoteVideo)`

開始撥號連線，參數 `inLocalVideo` 為 activity 之 layout 欲顯示本機端影像的畫面位置 `ChtmlSurfaceViewRenderer` 的 Resource ID，`inRemoteVideo` 為欲顯示遠端影像的畫面位置 `ChtmlSurfaceViewRenderer` 的 Resource ID，`inCallee` 為指定撥打支援端之 ID。

5.2 callee

`void callee(String inCallerId, int inLocalVideo, int inRemoteVideo, boolean inPickUp)`

回復是否接聽，參數 `inCallerId` 為撥號者之 ID，`inLocalVideo` 為 activity 之 layout 欲顯示本機端影像的畫面位置

`ChtmlSurfaceViewRenderer` 的 Resource ID，`inRemoteVideo` 為欲顯示遠端影像的畫面位置 `ChtmlSurfaceViewRenderer` 的 Resource ID，`inPickUp` 為是否接聽，另提供一個不接聽的 static 方法如下。

5.3 calleeAnswerNo

`static public JSONObject calleeAnswerNo(String inUserId, String inUserToken, String inServerPreUrl, String inCallerId)`

其中 `inUserId` 為使用者帳號，`inUserToken` 為使用者登入系統回覆之戳記，`inServerPreUrl` 為登入系統之前段 URL，`inCallerId` 為撥號者之 ID，回覆結果之說明如下：

屬性	必備屬性	屬性格式	描述	備註
status_code	◎	數字	結果代碼	status_code:1000 成功 status_code:2000 失敗，id 不存在 status_code:3000 失敗，查無符合 token status_code:4000 失敗，系統錯誤
status_msg		字串	結果訊息	status_code>1000 會有訊息
回傳 JSON 範例如下： <pre>{ "status_code": "1000", "status_msg": "XXX 您好，成功" }</pre>				

```
}

```

5.4 getCalleesList

static public JSONObject getCalleesList(String inUserId, String inUserPwd, String inServerPreUrl, Context inContext, String inPreferencesName)

為靜態方法取得在線上的支援端清單，回傳查詢結果為 JSONObject，請參考下方表格，參數 userId 為使用者帳號，userToken 為使用者登入系統回覆之戳記，serverPreUrl 為登入系統之前段 URL，inContext 為 Application Context，inPreferencesName 原 APP 與 SignalService 共用之 PreferencesName。

屬性	必備屬性	屬性格式	描述	備註
status_code	◎	數字	查詢結果代碼	status_code:1000 成功 status_code:2000 失敗，id 不存在 status_code:3000 失敗，查無有效 chat room status_code:4000 失敗，系統錯誤 status code:110 失敗，訊息服務尚未啟動完成!
status_msg	◎	字串	查詢結果訊息	status code 非 1000 會有錯誤訊息
callee	◎	Array	接受端資訊	
callee_ID	◎	字串	回傳 online callee id	人員 id

回傳 JSON 範例如下：

```
{
  "status_code": "1000",
  "status_msg": "XXX 您好，查詢成功",
  "callee": [
    { "callee_ID": "cht_mtest002",
      "name": "錢二"
    },
    { "callee_ID": "cht_mtest003",
      "name": "張三"
    }
  ]
}
```

}

5.5 hangup

`void hangup()`

掛斷雙向視訊。

5.6 newInstance

`M2MFragment newInstance(String userId, String userToken, String serverPreUrl)`

取得一個 `M2MFragment` 物件，參數 `userId` 為使用者帳號，`userToken` 為使用者登入系統回覆之戳記，`serverPreUrl` 為登入系統之前段 URL。

5.7 onOffFlashLight

`void onOffFlashLight()`

開關相機的 LED 燈。

5.8 sendClearingBoardRequest

`void sendClearingBoardRequest()`

傳送清除畫板要求訊息。

5.9 sendLocalBoardFile

`void sendLocalBoardFile(String filePathString)`

傳送畫板檔案，參數 `filePathString` 為畫板檔案在手機的絕對路徑。

5.10 sendLocalFreezingFile

`void sendLocalFreezingFile(String filePathString)`

傳送凍結畫面檔案，參數 `filePathString` 為凍結畫面檔案在手機的絕對路徑。

5.11 `sendUnfreezingBoardRequest`

`void sendUnfreezingBoardRequest()`

傳送解除凍結畫面要求訊息。

5.12 `setDebug`

`void setDebug(boolean isDebug)`

設定是否顯示 Debug 訊息，參數 `isDebug true` 為顯示 Debug 訊息。

5.13 `setLocalVideoSize`

`void setLocalVideoSize(int inWidth, int inHeight, boolean isPortrait)`

設定手機端影像的大小及畫面直式或橫式，參數 `inWidth` 為影像寬度，`inHeight` 為影像高度，`isPortrait true` 畫面為直，預設為 720x1280 直式。

5.14 `setWebRTCConnectTimeout`

`void setWebRTCConnectTimeout(int connectTimeout)`

設定雙向視訊建立連線之 timeout，參數 `connectTimeout` 為 timeout 時間單位為毫秒，預設值為 60000 毫秒, 60 秒。

5.15 `noBlockSendLocalFile`

`void noBlockSendLocalFile(String inFilePath)`

傳送本機檔案至遠端，參數 `inFilePath` 為檔案所在之絕對位置。

5.16 registerFileTransferListener

```
void registerFileTransferListener(WRLFileTransferListener
inWRLFileTransferListener)
```

向 M2MFragment 註冊檔案傳輸之 Listener，參數 WRLFileTransferListener 為檔案傳輸之 Listener，此 Listener 需 implement 的 Callback Function 說明如下：

一. 傳送檔案處理進度 Callback Function

```
public void onSendProgressUpdate(String fileName, final long fileSize, int
maxValue, int minValue, int progressValue)
//fileName:欲傳送檔案短檔名  fileSize:檔案大小  maxValue:Progress 的最大
值  minValue:Progress 的最小值  progressVaule:目前 Progress 之值
```

二. 傳送檔案處理完成 Callback Function

```
public void onSendFinish(String fileName)
//完成傳送檔案  fileName:欲傳送檔案短檔名
```

三. 接收檔案處理進度 Callback Function

```
public void onReceiveProgressUpdate(String fileName, long fileSize, int maxValue,
int minValue, int progressValue)
//fileName:接收檔案短檔名  fileSize:檔案大小  maxValue:Progress 的最大值
minValue:Progress 的最小值  progressVaule:目前 Progress 之值
//舊版接收檔案因無檔案大小，所以
fileSize,maxValue,minValue,progressValue 皆為 0
```

四. 接收檔案處理進度 Callback Function

```
public void onReceiveFinish(String fileName, String fullFileName)
//完成接收檔案  fileName:接收檔案短檔名  fullFileName:接收檔案完整的儲
存檔名
```

5.17 unregisterFileTransferListener

```
void unregisterFileTransferListener ()
```

向 M2MFragment 取消檔案傳輸 Listener 之註冊。

5.18 setEnableLocalCamera

```
void setEnableLocalCamera(boolean enableLocalCamera)
```

設定是否開啟 Local Camera，預設為 true 開啟，參數 enableLocalCamera 設為 false 則關閉，需在視訊開始前設定，可用來關閉 callee 的視訊以提升 caller 的視訊品質。

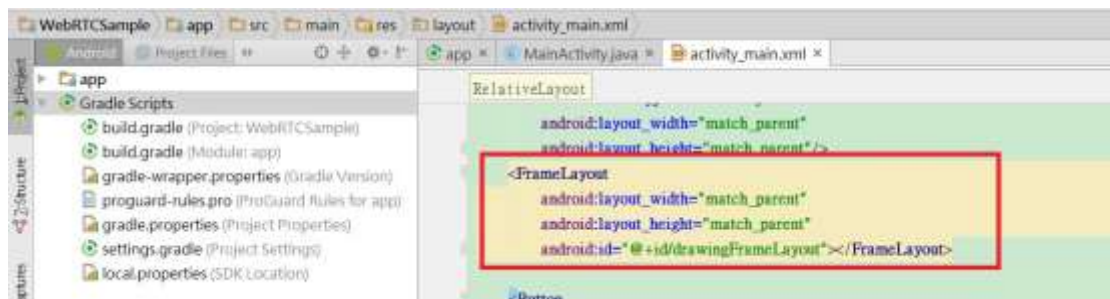
5.19 sendMessage

```
void sendMessage(String inTextMessage)
```

傳送文字訊息之方法，參數 inTextMessag 為傳送之文字訊息。

1.5.2 使用繪圖 SDK 的方式

1. 在 App 程式欲使用本 SDK 的 activity 之 layout 設定檔案加入空的 FrameLayout，如下圖中紅框的設定：



2. 在 activity 程式中加入 DrawingFragment 片段：

```
//將畫板 Fragment 加入 Activity
FragmentManager fragmentManager = getFragmentManager();
FragmentTransaction fragmentTransaction =
    fragmentManager.beginTransaction();
mDrawingFragment = DrawingFragment.newInstance();
fragmentTransaction.replace(R.id.drawingFrameLayout,
    mDrawingFragment, "DrawingFragment");
fragmentTransaction.commit();
```

3. 設定 DrawingFragment 的 DBLSurfaceViewRendererInterface

```
mDrawingFragment.setDBLSurfaceViewRendererInterface(
    (DBLSurfaceViewRendererInterface)findViewById(
        R.id.local_video_view));
```

其中 R.id.local_video_view 為加入 WebRTC SDK 的 activity 之

layout 欲顯示手機端影像的畫面位置 ChtmlSurfaceViewRenderer 的 Resource ID。

4. 加入 DrawingFragment 片段還需要 implements DBLMessageListener，並覆載(Override)以下方法(Method)，並對應到 App 程式欲呈現這些資訊的方式：

4.1 從手機端畫板畫面已儲存完畢之 Callback：

```
@Override
public void onDBLFileSaved(String filePathString)
    //例如可在此傳送畫板畫面檔案至支援端
```

4.2 從手機端使用者按下清除畫板之 Callback：

```
@Override
public void onDBLClear()
    //例如可在此做清除支援端畫板及傳送清除畫板 message 至支援端
```

4.3 從手機端凍結畫面已儲存完畢之 Callback：

```
@Override
public void onDBLFreezeFileSaved(String filePathString)
    //例如可在此做顯示手機端凍結畫面及傳送凍結畫面檔案至支援端
```

4.4 從手機端使用者按下解凍按鈕之 Callback：

```
@Override
public void onDBLUnfreeze()
    //例如可在此做清除手機端凍結畫面及傳送解凍 message 至支援端
```

4.5 從手機端使用者按下文字輸入按鈕之 Callback：

@Override

```
public void onDBLInputButtonPressed ()
```

//例如可在此顯示使用者文字輸入畫面，輸入完成以下方式傳給對方

4.6 從手機端使用者按下傳送檔案按鈕之 Callback：

@Override

```
public void onDBLFileButtonPressed ()
```

//例如可在此顯示使用者選取檔案畫面，輸入完成以下方式傳給對方

4.7 從手機端使用者按下 LED 按鈕之 Callback：

@Override

```
public void onDBLLedButtonPressed()
```

//按 LED 時，呼叫 M2MFragment 的 onOffFlashLight Method

5. DrawingFragment 片段所提供的方法如下：

5.1 clearDrawingBoard

```
public void clearDrawingBoard()
```

清除手機端畫版。

5.2 newInstance

```
public static DrawingFragment newInstance()
```

取得一個 DrawingFragment 物件。

5.3 newInstance

```
public void
```

```
setDBLSurfaceViewRendererInterface(DBLSurfaceViewRendererInterface inDBLSurfaceViewRendererInterface)
```

設定 DrawingFragment 的 DBLSurfaceViewRendererInterface，以便

取得視訊的畫面，參數 `inDBLSurfaceViewRendererInterface` 為加入 WebRTC SDK 的 activity 之 layout 欲顯示手機端影像的畫面位置 `ChtmlSurfaceViewRenderer` 的 `DBLSurfaceViewRendererInterface`。

5.4 setFreeze

```
public void setFreeze()
```

設定目前狀態為凍結。

5.4 setUnfreeze

```
public void setUnfreeze()
```

設定目前狀態為解凍。

5.5 setInitColor

```
public void setInitColor(int inColor)
```

設定畫筆初始顏色。

5.5 removeLEDButton

```
public void removeLEDButton()
```

刪除畫版上的 LED 功能按鈕。

1.6 在 Activity 的 layout 各畫面的疊置

由下而上各畫面建議參考如下：

1. 最底層為手機端影像的畫面

```
<!-- 寫在前面會顯示在螢幕上面 -->
```

```
<FrameLayout
```

```
    android:layout_alignParentRight="true"
```

```
    android:layout_width="72dp"
```

```

        android:layout_height="128dp">
        <com.cht.tlcsl.webrtc.lib.ChtmlSurfaceViewRenderer
            android:id="@+id/remote_video_view"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </FrameLayout>
    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <com.cht.tlcsl.webrtc.lib.ChtmlSurfaceViewRenderer
            android:id="@+id/local_video_view"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </FrameLayout>

```

2. 凍結畫面

```

    <ImageView
        android:background="@android:color/transparent"
        android:id="@+id/freezeImageView"
        android:scaleType="centerCrop"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

```

3. 支援端畫版畫面

```

    <ImageView
        android:background="@android:color/transparent"
        android:id="@+id/boardImageView"
        android:scaleType="centerCrop"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

```

4. 手機端畫版畫面

```

    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/drawingFrameLayout">

```

</FrameLayout>