

eSDK ICP V200R001C10 开发指南 01(CC, Android)

文档版本 01
发布日期 2017-03-17

华为技术有限公司



版权所有 © 华为技术有限公司 2017。 保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI 和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编：518129

网址： <http://www.huawei.com>

客户服务邮箱： support@huawei.com

客户服务电话： 4008302118

目 录

1 eSDK CC 是什么?	1
2 内容导航	2
3 相关资源	3
3.1 华为开发者社区	3
3.2 SDK 下载路径	3
3.3 Sample Codes	3
3.4 接口参考	4
3.5 免费申请远程实验室	4
3.6 SDK 修订记录	6
3.7 技术支持渠道	7
4 Hello World	8
4.1 概述	8
4.2 准备环境	9
4.3 新建工程项目	10
4.4 设置编码格式	12
4.5 引入 jar 包	13
4.6 新建包	14
4.7 新建类	15
4.8 编码实现	17
4.9 编译及调试	21
5 典型场景开发	23
5.1 场景一：登录与登出	23
5.2 场景二：TP 场景下的呼叫	26
5.3 场景三：MS 场景下的呼叫	28
5.4 场景四：设备控制	32
6 问题定位	34
7 修订记录	37
8 附录	38

8.1 Hello World 源码文件	38
8.1.1 CCApplication.java	38
8.1.2 MainActivity.java	38
8.1.3 build.gradle	42
8.1.4 AndroidMainifest.xml	42
8.1.5 values/strings.xml	44
8.1.6 values-zh/strings.xml	44
8.1.7 values/styles.xml	44
8.1.8 layout/activity_main.xml	45

1 eSDK CC 是什么？

eSDK ICP 解决方案简介

华为 ICP（Integrated Communication Platform，集成通信平台）融合指挥解决方案通过丰富的开放接口，可以与火警、匪警、急救、视频监控、无线电通讯系统、公共电话系统、CAD（计算机辅助调度）业务系统等对接，用于公众报告紧急事件和紧急求助，统一接警，统一指挥，联合行动，为市民提供相应的紧急救援服务，为城市的公共安全提供强有力的保障，大力加强了各警力单位之间的配合与协调，从而对特殊、突发、应急和重要事件做出有序、快速而高效的反应。

ICP 集成通信平台作为融合指挥的核心，实现多语音网络、多终端的语音、多视频系统的全连接，实现固定电话、移动电话、各种集群终端、智真会议等不同通信设备之间的互联互通，通过多种网络融合的语音通信，实现对应急事件处置的统一指挥调度和应急决策信息的快速传达。

CC SDK 是什么

CC SDK 作为 eSDK ICP 解决方案中的一个子模块，提供呼叫，呼叫设备控制等功能。

我们提供给你的 SDK 包内容有以下几部分内容：

- **CC SDK 包**

二次开发使用的 SDK 包，包括软件包和接口参考文档。详情请参见 3.2 SDK 下载路径。

- **Sample Codes**

华为 SDK 提供一系列 Sample codes 演示如何调用接口，帮助您完成 eSDK CC Android 接口相关业务开发。详情请参见 3.3 Sample Codes。

2 内容导航

本开发指南指导您如何安装和配置 eSDK CC Android 环境、如何调用 eSDK CC Android 标准化接口以及华为 eSDK 提供的开发者支持服务。主要内容如下：

1. 3 相关资源：二次开发过程中可能用到的软件、文档资源链接、技术支持。包括如何通过社区网站获取资料，sample codes 下载链接，如何申请远程实验室等。
2. 4 Hello World：如果你仅仅是要把 SDK 运行起来，你应该首先看这部分，它会详细说明如何下载及安装 SDK，以及如何配置你的开发环境。
3. 5 典型场景开发：介绍 eSDK CC 开放性能的典型功能场景，包括开发流程、样例代码以及注意事项等。
4. 6 问题定位：开发过程中常见问题的定位解决方法。

阅读建议

- 如果只是想快速入门，可仅参考 4 Hello World 章节。
- 如果想深入了解 eSDK CC Android 核心业务的二次开发，建议您参考 5 典型场景开发章节。
- 使用 SDK 过程中遇到问题可以参考 6 问题定位章节或 3.7 技术支持渠道。

3 相关资源

- 3.1 华为开发者社区
- 3.2 SDK 下载路径
- 3.3 Sample Codes
- 3.4 接口参考
- 3.5 免费申请远程实验室
- 3.6 SDK 修订记录
- 3.7 技术支持渠道

3.1 华为开发者社区

您可通过[华为开发者社区企业云通信版块](#)，快速体验 eSDK ICP 业务功能，获取 eSDK ICP 的二次开发 SDK 工具包以及技术支持等。

3.2 SDK 下载路径

进入[华为开发者社区资源中心](#)，点击“**SDK > 企业云通信 > ICP > eSDK CC SDK**”，选择对应版本进行下载。

推荐使用最新的版本 V2.1.10。

3.3 Sample Codes

这里建议您使用 Android Studio 1.5 编译执行 Sample Codes。

如果需要下载及安装 SDK，请参考 4 Hello World。

Sample Codes

名称	描述
eSDK_ICP_Demo_V2.1.10_Android.zip	基于 eSDK CC Android Sample 包括 TP 和 MS 环境下呼叫，设备控制。

3.4 接口参考

接口参考主要包含：

概述：明确配套版本、使用背景、场景、前提条件等内容；并指出可以获取哪些信息，完成什么样的功能。

数据类型：详细说明 eSDK 对外提供的自定义的数据类型（包括数据结构、枚举、类等），例如：

- 数据类型名。
- 对有继承/嵌套关系的数据结构进行说明，如结构体嵌套关系，要提供总的嵌套关系表（要包括基本数据类型）。
- 包括的数据成员以及数据成员的含义。

接口详细描述：

- 接口描述：详细描述该接口功能、应用场景和使用方法。
- 使用说明：描述该接口函数使用时需要注意的事项、使用的限制；功能相似的接口或者需要配合使用的接口；前提条件等。
- 方法定义：接口函数的完整声明。
- 参数描述：详细描述参数的含义、取值范围、使用限制、参数之间的依赖等。
- 返回值：说明该接口函数的返回值。
- 使用示例：举例说明该接口函数的使用，关键代码需要注释。

3.5 免费申请远程实验室

华为 eSDK 远程实验室简介

华为远程实验室为开发者提供了 7×24 小时的免费的云化实验室环境，提供真实的华为设备供开发者远程在线开发调试。借助远程实验室自助管理平台，开发者不需要购置华为产品便可基于远程实验室针对相关产品进行二次开发，并实现远程对接测试认证。目前华为远程实验室已发布云计算、SDN、大数据、企业云通信、企业移动安全等 10 个生态圈的 45 个实验室环境。

相关信息请查询[远程实验室首页](#)。

远程实验室有哪些优势

- 低门槛：官网注册用户即可申请使用，环境与预约时长、预约次数受限；
- 分级支持：环境按域划分，重点开发者、合作伙伴可访问特定环境并享受额外的预约环境；
- 全球资源高速互联：建设以苏州远程实验室为中心的实验室分布格局，依托 IT 全球 100ms 高性能骨干网络和 IT 全球端到端应用保障能力。

如何免费申请远程实验室



步骤 1 注册账号。

1. 若您已有华为注册账号，则直接在登录远程实验室首页使用已注册账号登录。
2. 若您还未注册华为账号，请进入华为远程实验室网站：
<http://developer.huawei.com/cn/ict/remotelab>，点击“**进入实验室**”，页面跳转至账号注册页，填写注册信息，并在注册邮箱中激活您的华为账号。



步骤 2 预约环境。

1. 若您已成功预约环境，且环境可用时长在有效期范围内，则请忽略此步骤。
2. 华为账号注册成功后，系统会自动跳转到华为远程实验室首页，状态为已登录。使用 CC 调试环境 Contact Center，点击“预约”按钮，默认预约的环境可用时长为 2 小时，如下图所示。



3. 预约成功后，系统会自动发送 SVN 网关地址、用户名、密码等信息至您的华为账户注册邮箱中，根据邮件中的提示信息下载 SSL-VPN 客户端软件，完成本地安装。后续步骤需要使用该信息登录 SVN 客户端进行远程环境连接。

步骤 3 接入环境。

1. 若您成功已接入华为远程实验室环境，请忽略此步骤。
2. 打开 SVN 客户端，将预约成功后系统发送给您的 SVN 网关地址、用户名、密码等信息，分别输入到客户端对应的输入框内，并点击“登录”。



步骤 4 调测发布。

使用申请成功的 CC 平台账号、密码、IP 地址、Port 等信息，完成登录，调测您的应用程序。详见远程实验室操作引导中的“CC 平台登录信息”。

----结束

3.6 SDK 修订记录

SDK 会逐渐的升级来支持新的服务。您可以参考开发者社区网站历史版本了解不同版本有哪些更新。具体的更新信息包含：

- SDK 名称
- 产品名称: SDK 配套的产品名称。
- 更新时间: SDK 更新发布上线的时间。
- 版本号: SDK 当前的版本号。
- 下载链接: 链接中包含 SDK Demo 及配套文档。
- 更新记录描述: 描述变更的特性。

3.7 技术支持渠道

技术支持渠道

如果您在软件开发过程中碰到任何问题, 开发者社区提供了以下技术支持渠道:

- 可以在[华为开发者论坛](#)发帖求助。
- 可以在 [DevCenter](#) 系统中提单跟踪。

如果您在文档使用过程中有任何疑问, 可以通过以下方式联系我们。

- 华为技术支持热线电话: 400-8828-000
- 华为技术支持邮箱: esdk@huawei.com

4 Hello World

- [4.1 概述](#)
- [4.2 准备环境](#)
- [4.3 新建工程项目](#)
- [4.4 设置编码格式](#)
- [4.5 引入 jar 包](#)
- [4.6 新建包](#)
- [4.7 新建类](#)
- [4.8 编码实现](#)
- [4.9 编译及调试](#)

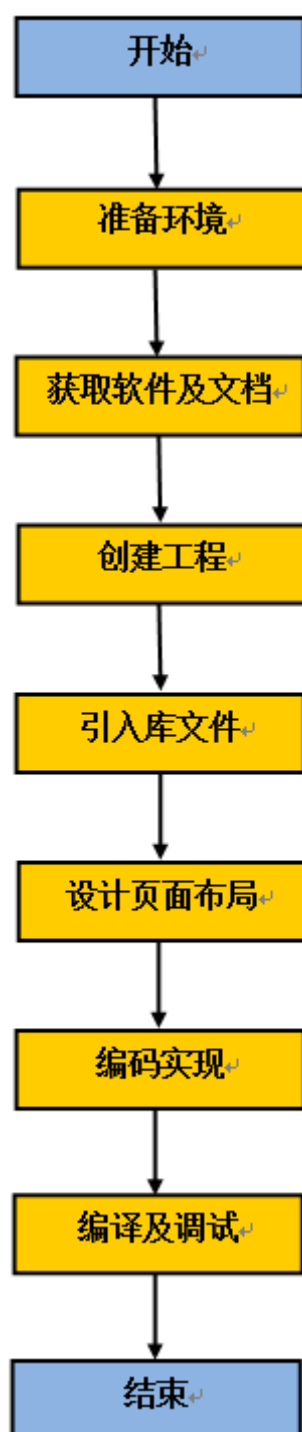
4.1 概述

Hello World 开发流程

本示例以 JAVA 语言简单向您展示如何进行 eSDK CC 的二次集成开发。

开发过程中的故障处理请参考 6 问题定位。

快速入门流程如下：



4.2 准备环境

开发工具

- 操作系统：Windows 7 专业版。

- Android Studio 1.5 及以上版本。
- Java Development Kit 1.7 及以上版本。

SDK 软件包

- SDK 软件包名称：**eSDK_ICP_SDK_V2.1.10_Android.zip**。
- SDK 软件包下载路径：参见 3.2 SDK 下载路径。

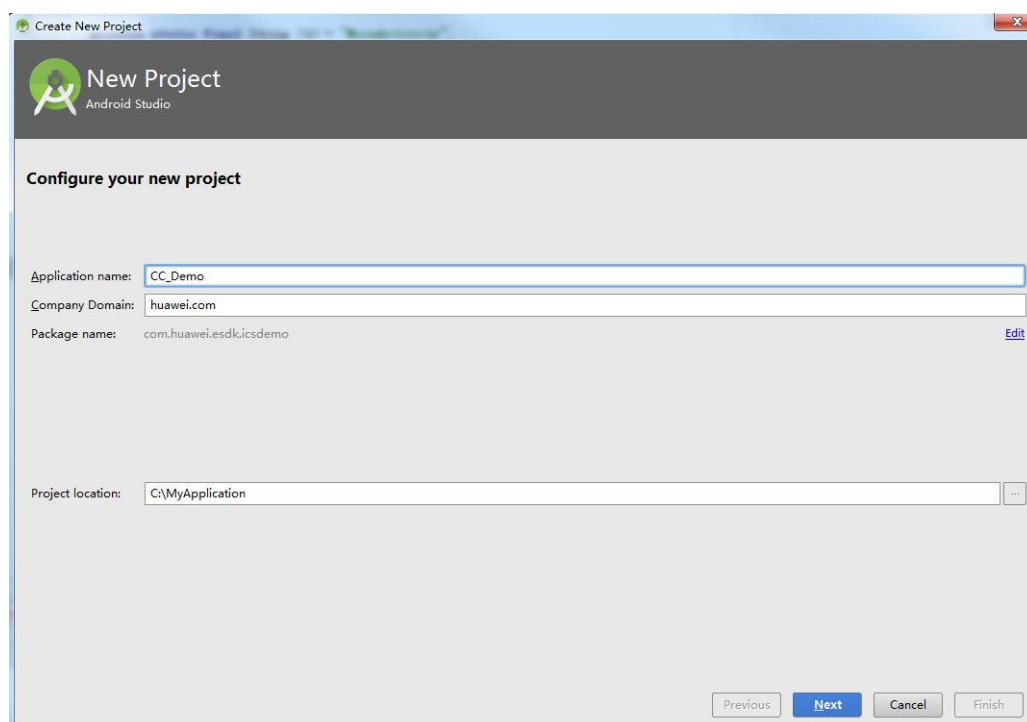
注意：如果需要证书验证功能，需要将证书放在工程中 assets 文件夹下，例如“assets/certs/server.cer”。

使用三星 note4 测试呼叫功能，需要在 AndroidManifest 文件中增加蓝牙权限。网络访问默认使用 https 方式。视频呼叫过程中，当前 Activity 设置为横屏，也就是 android:screenOrientation="landscape"，展现的视频就是横的，Activity 设置为竖屏，也就是 android:screenOrientation="portrait"，展现的视频就是竖的。

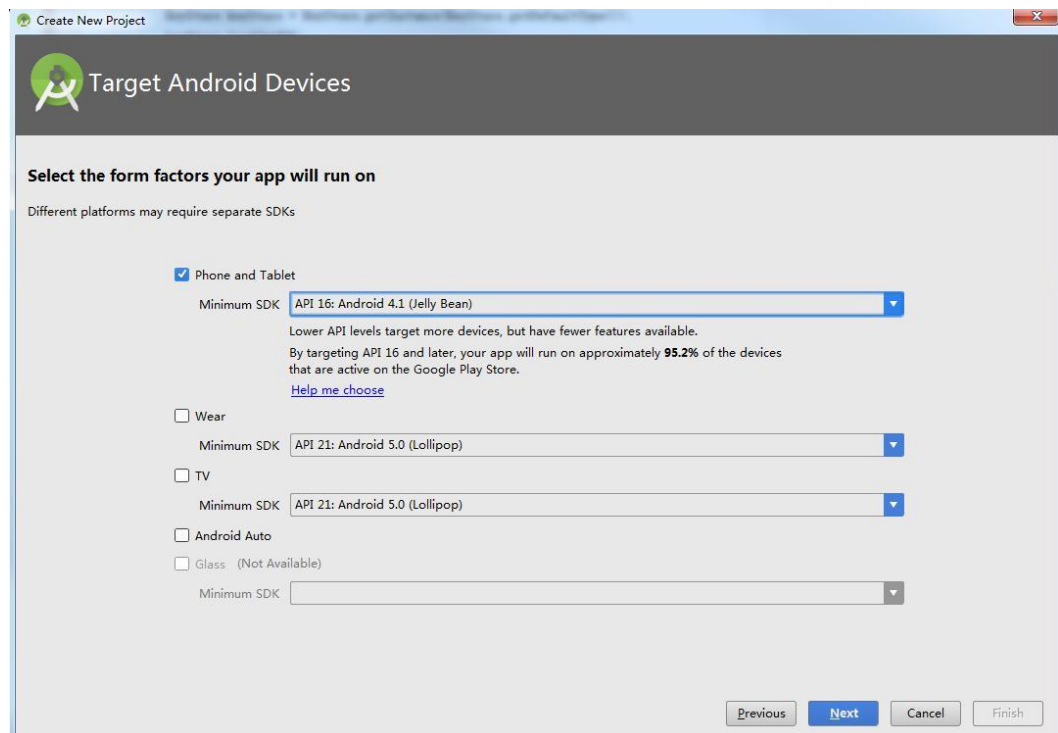
使用本地广播 LocalBroadcastManager，需要在 build.gradle 依赖项里面写上 compile 'com.android.support:support-v4:23.1.1'。

4.3 新建工程项目

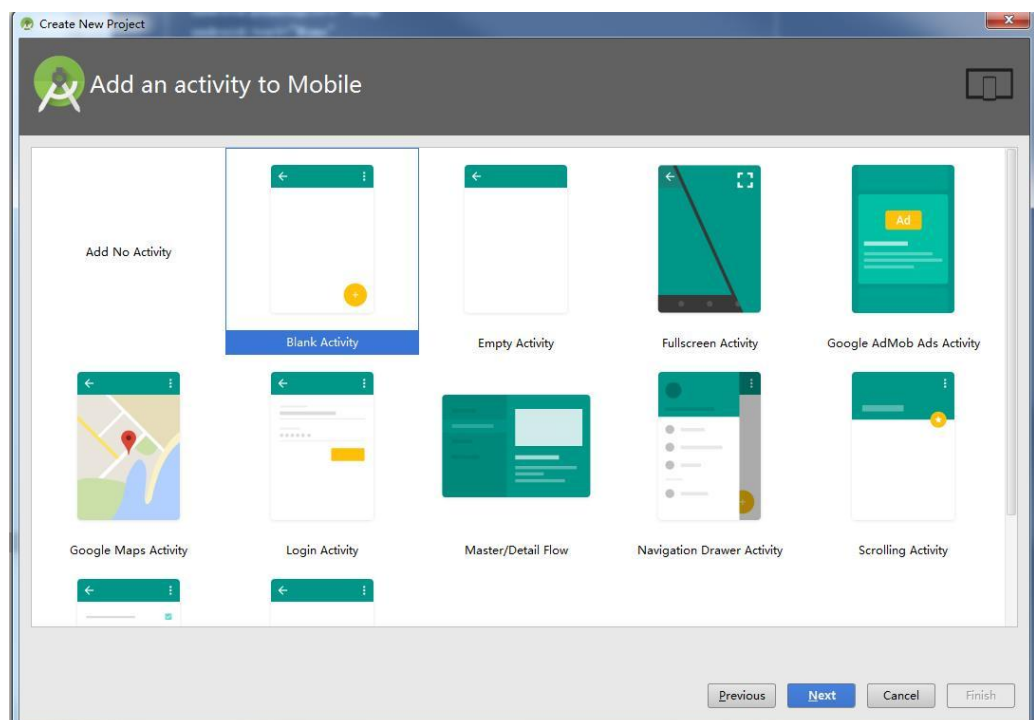
步骤 1 打开 Android Studio，选择“File > New > New Project”。系统显示“Create New Project”界面。



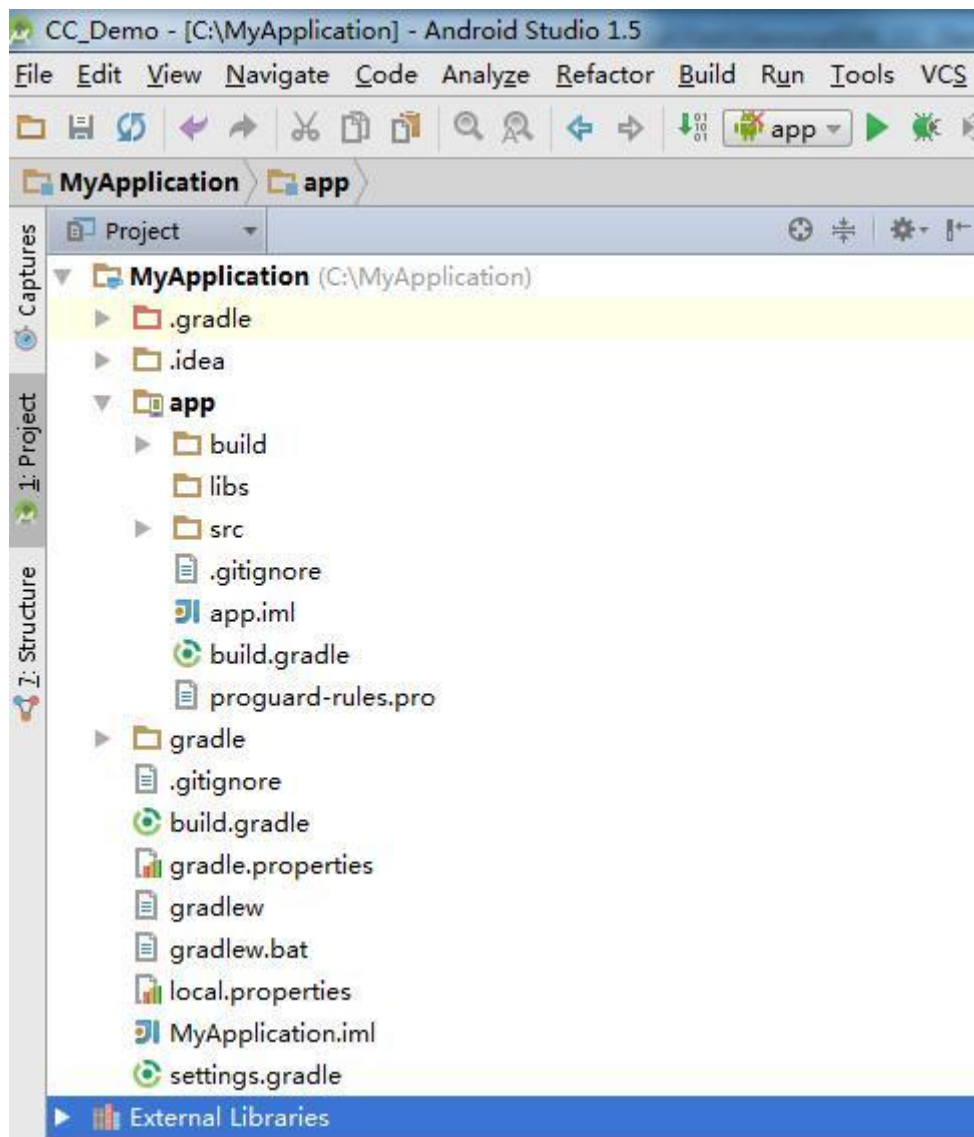
步骤 2 填写 Application name、Company Domain 和 Project location 信息后，点击“Next”，将进入 SDK 版本选择界面。



步骤 3 选取最低支持系统的 SDK 版本后，点击“Next”，将进入 Activity 展示效果界面。



步骤 4 选取“Blank Activity”，点击“Finish”，完成工程创建。

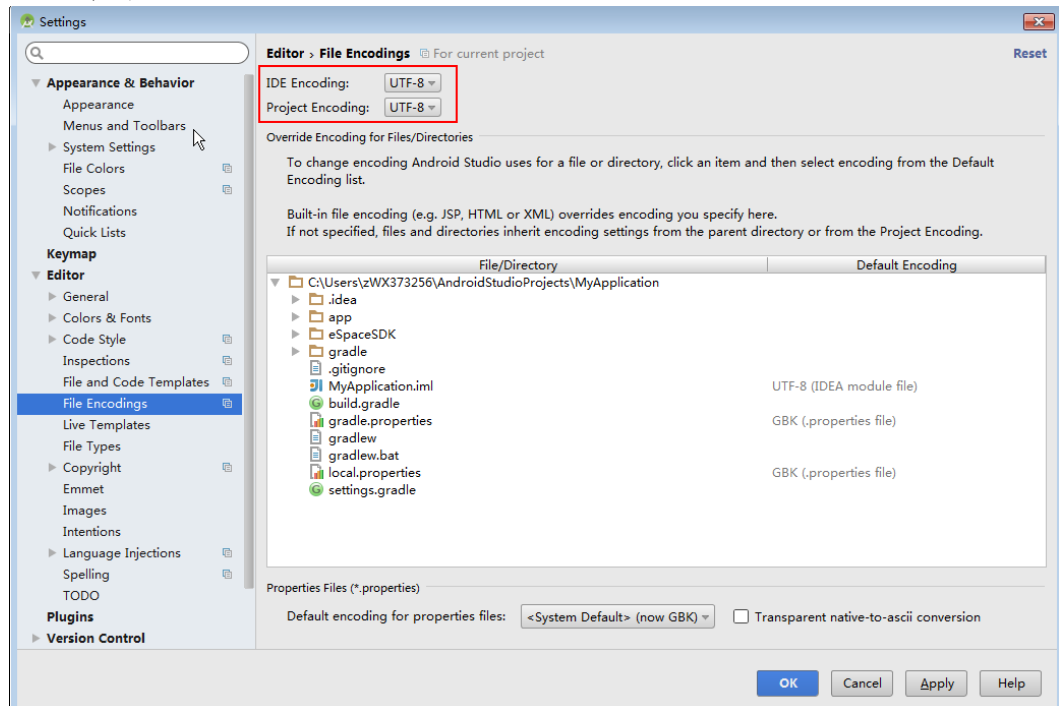


----结束

4.4 设置编码格式

步骤 1 Android Studio1.5 中点击“File > Settings > File Encodings”设置编码格式。

步骤 2 将“IDE Encoding”和“Project Encoding”对应的下拉框中将编码格式设置为“UTF-8”，单击“OK”。



----结束

4.5 引入 jar 包

步骤 1 解压下载的资源“eSDK_ICP_SDK_V2.1.10_Android.zip”，解压缩后将 jar 文件复制到工程目录下的“libs”文件夹中，so 文件复制到“armeabi”文件夹中。


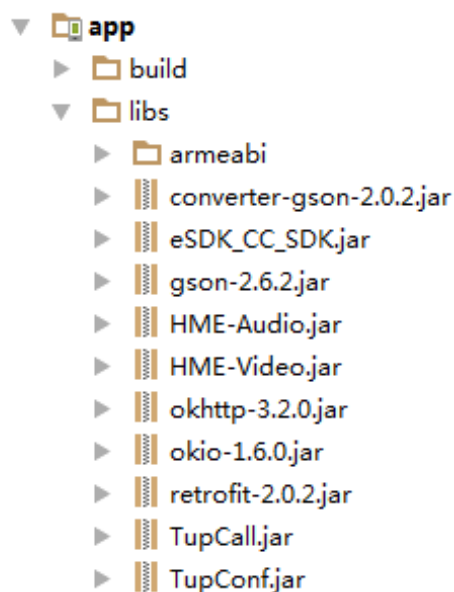
步骤 2 点击一下  图标，刷新工程，jar 包前面有小箭头说明 jar 包被成功导入。

图4-1

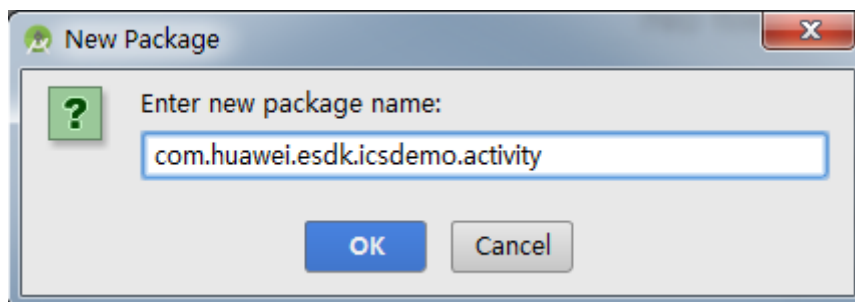


----结束

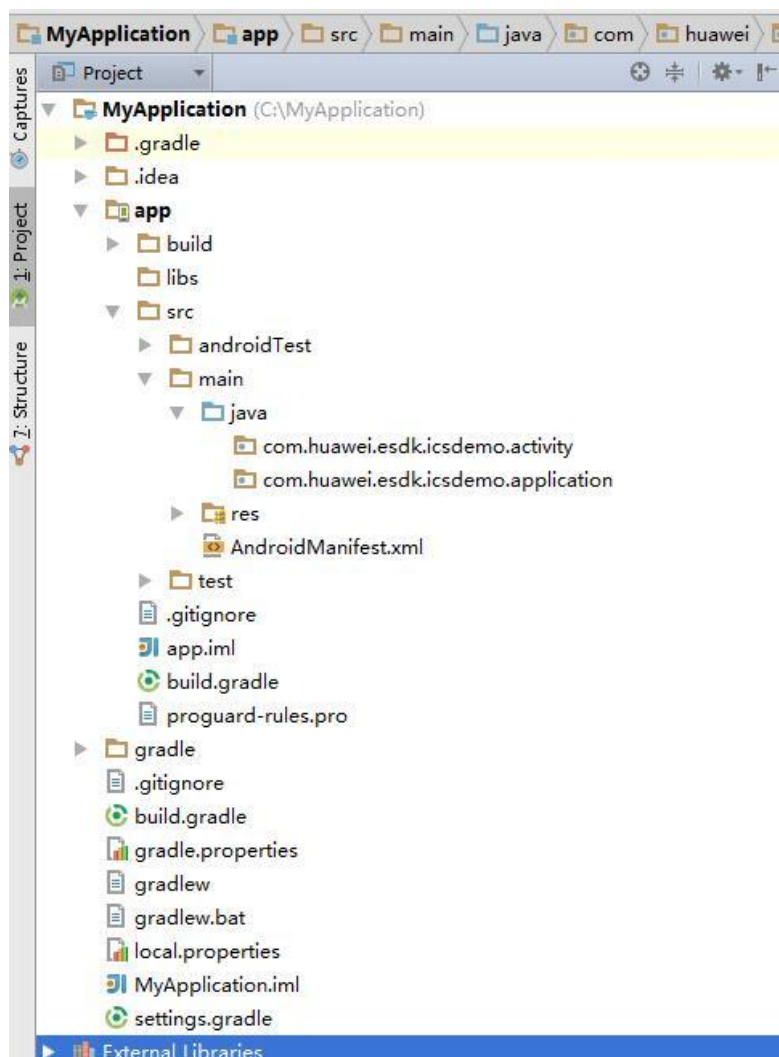
4.6 新建包

步骤 1 点开“App>src>main>java”，右键单击“java”文件夹，右键选择“New > Package”，弹出“New Package”窗口，在“Enter new package name”处输入“com.huawei.esdk.icsdemo.activity”。

图4-2



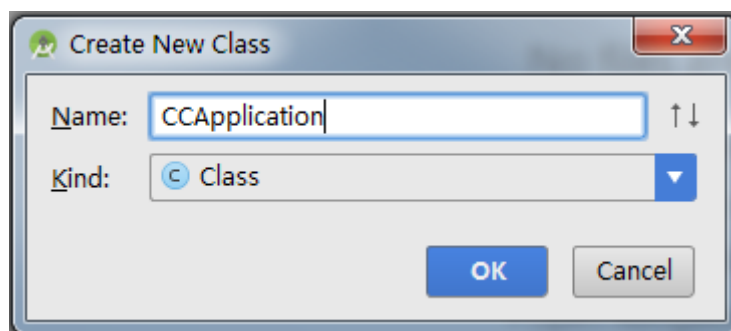
步骤 2 同步骤 1 新建“com.huawei.esdk.icsdemo.application”。



----结束

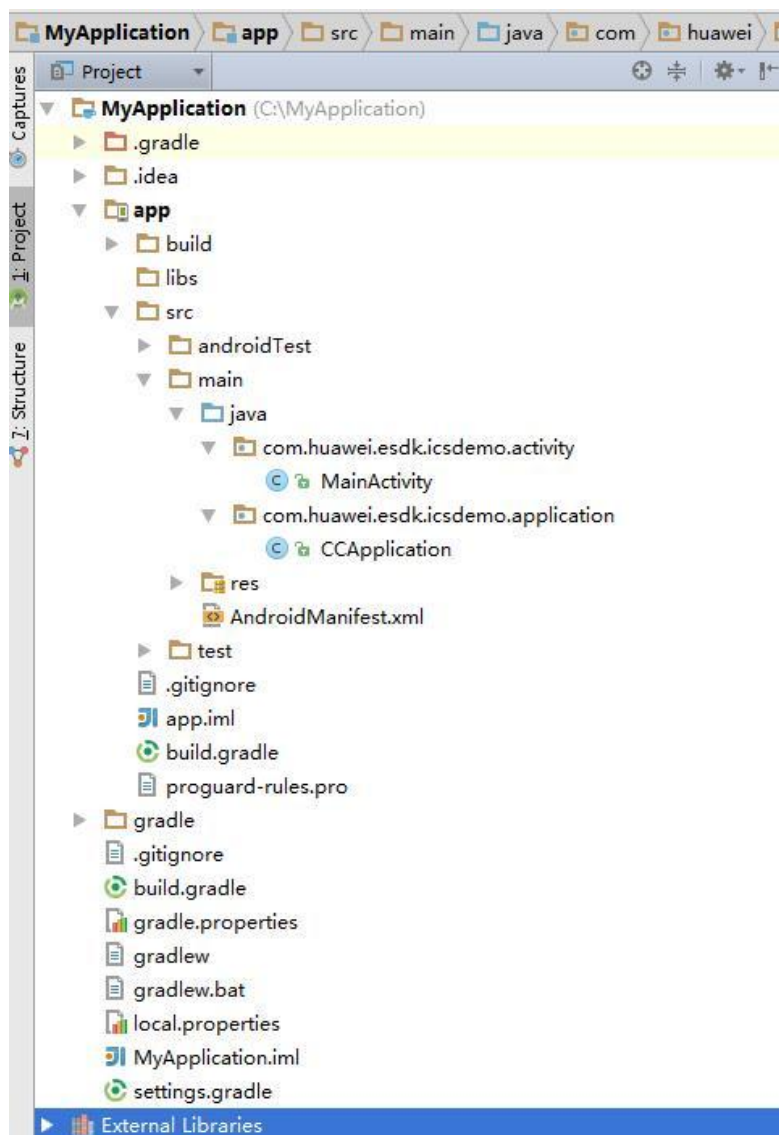
4.7 新建类

步骤 1 右键“com.huawei.esdk.icsdemo.application”包，选择“New”，弹出“Java Class”窗口。



步骤 2 在“Name”处输入“CCApplication”，点击“OK”，完成类的新建。

步骤 3 重复以上步骤，按照所示，在 `com.huawei.esdk.icsdemo.activity` 包下建立 `MainActivity`，已有该文件的工程需要修改 `MainActivity`，并填充相应的代码，详见 8 附录。



步骤 4 在 `res` 目录下，新建 `values-zh` 文件夹，创建 `strings.xml`，并填充代码。

步骤 5 如果 `menu` 文件夹下的文件 `app:showAsAction="never"` 报错，那就把 `app:showAsAction="never"` 替换成 `android:showAsAction="never"`

如果遇到 `styles.xml` 报错，那就用开发指南里附录中的 `styles.xml` 替换。

----结束

4.8 编码实现

总体结构

```
CC Demo
-- src
--com.huawei.esdk.icsdemo.activity
--MainActivity.java
--com.huawei.esdk.icsdemo.application
--CCApplication.java
--res
--layout
--activity main.xml
--values
--strings.xml
--styles.xml
--values-zh
--strings.xml
--AndroidManifest.xml
--build.gradle
```

源码链接：参见 8 附录。

关键代码

部分代关键代码参考如下：

1. MainActivity

```
//初始化 view、监听和 filter
//java code
private void initView()
{
    etIP = (EditText) findViewById(R.id.et_ip);
    etPort = (EditText) findViewById(R.id.et_port);
    etName = (EditText) findViewById(R.id.et_name);
    btnLogin = (Button) findViewById(R.id.btn_login);

    filter = new IntentFilter();
    filter.addAction(NotifyMessage.AUTH_MSG_ON_LOGIN);
    filter.addAction(NotifyMessage.AUTH_MSG_ON_LOGOUT);
}

//注册广播
@Override
protected void onResume()
{
    super.onResume();
    registerReceiver(receiver, filter);
}

@Override
protected void onPause()
{

```

```
        super.onPause();
        unregisterReceiver(receiver);
    }
    //receiver 逻辑

    private BroadcastReceiver receiver = new BroadcastReceiver()
    {
        @Override
        public void onReceive(Context context, Intent intent)
        {
            String action = intent.getAction();
            BroadMsg broadMsg = (BroadMsg) intent
                .getSerializableExtra (NotifyMessage.CC_MSG_CONTENT);

            if (NotifyMessage.AUTH_MSG_ON_LOGIN.equals(action))
            {
                //当 retcode 为空时, 通过 errorcode 提示用户
                if (null == broadMsg.getRetCode())
                {
                    Toast.makeText (MainActivity.this,
                        getString(R.string.login_fail) + broadMsg.getErrorCode(),
                        Toast.LENGTH_SHORT).show();
                }
                //当 retcode 不为空时, 通过服务器返回的错误码来提示用户
                else
                {
                    if ("0".equals(broadMsg.getRetCode()))
                    {
                        //登录成功
                        Toast.makeText (MainActivity.this,
                            getString(R.string.login_success), Toast.LENGTH_SHORT).show();
                    }
                    else
                    {
                        Toast.makeText (MainActivity.this, "登陆失败! " +
                            broadMsg.getRetCode(), Toast.LENGTH_SHORT).show();
                    }
                }
            }
            else if (NotifyMessage.AUTH_MSG_ON_LOGOUT.equals(action))
            {
                if (null == broadMsg.getRetCode())
                {
                    Toast.makeText (MainActivity.this,
                        getString(R.string.logout_fail) + broadMsg.getErrorCode(),
                        Toast.LENGTH_SHORT).show();
                }
                else
                {
                    //注销成功
                    if ("0".equals(broadMsg.getRetCode()))
                    {
                        Toast.makeText (MainActivity.this, "注销成功!
" , Toast.LENGTH_SHORT).show();
                    }
                }
            }
        }
    }
}
```

```

        else
        {
            Toast.makeText(MainActivity.this,
getString(R.string.logout_fail) + broadMsg.getRetCode(),
Toast.LENGTH_SHORT).show();
        }
    }
}
};

```

2. CCApplication

```

//java code
public class CCApplication extends Application
{
    @Override
    public void onCreate()
    {
        super.onCreate();
        MobileCC.getInstance().setLog("eSDK", 3);
        MobileCC.getInstance().initSDK(this);
    }

    @Override
    public void onTerminate()
    {
        super.onTerminate();
        MobileCC.getInstance().unInitSDK();
    }
}

```

3. AndroidManifest

配置文件中的 SDK 版本需要和当前编译环境一致。

```

//xml code
<?xml version="1.0" encoding="utf-8"?>
<manifest package="com.huawei.esdk.icsdemo"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:versionCode="1"
    android:versionName="1.5.70">

    <uses-sdk
        android:minSdkVersion="15"
        android:targetSdkVersion="19"/>

    <uses-permission
android:name="android.permission.ACCESS_WIFI_STATE"></uses-permission>
    <uses-permission android:name="android.permission.RECORD_AUDIO"></uses-
permission>
    <uses-permission android:name="android.permission.INTERNET"></uses-
permission>
    <uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE"></uses-permission>
    <uses-permission

```

```

        android:name="android.permission.MODIFY_AUDIO_SETTINGS"></uses-permission>
        <uses-permission android:name="android.permission.VIBRATE"></uses-
permission>
        <uses-permission android:name="android.permission.GET_TASKS"></uses-
permission>
        <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"></uses-permission>

        <uses-permission android:name="android.permission.RESTART_PACKAGES"></uses-
permission>
        <uses-permission android:name="android.permission.WAKE_LOCK"></uses-
permission>
        <uses-permission android:name="android.permission.BROADCAST_STICKY"></uses-
permission>
        <uses-permission android:name="android.permission.BLUETOOTH"></uses-
permission>
        <uses-permission android:name="android.permission.READ_PHONE_STATE"></uses-
permission>
        <uses-permission
android:name="android.permission.PROCESS_OUTGOING_CALLS"></uses-permission>
        <uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW"/>
        <uses-permission
android:name="android.permission.CHANGE_WIFI_STATE"></uses-permission>
        <uses-permission android:name="android.permission.CAMERA"></uses-
permission>

        <uses-feature
            android:name="android.hardware.camera"
            android:required="true"/>

        <application
            android:name="com.huawei.esdk.icsdemo.application.CCApplication"
            android:allowBackup="true"
            android:icon="@mipmap/ic_launcher"
            android:label="@string/app_name"
            android:hardwareAccelerated="false"
            android:theme="@style/AppTheme">
            <activity
                android:name=".activity.MainActivity"
                android:label="@string/app_name"
                android:theme="@style/AppTheme">
                <intent-filter>
                    <action android:name="android.intent.action.MAIN"/>

                    <category android:name="android.intent.category.LAUNCHER"/>
                </intent-filter>
            </activity>
            <service android:name="com.huawei.AudioDeviceAndroidService"/>
        </application>
    </manifest>

```


4.9 编译及调试


有华为 CC 环境

若您已部署华为 ICP 解决方案，请直接填写登录平台的用户名、密码以及 IP 地址，按照下面介绍的方法[调试运行](#)程序。

无华为 CC 环境

若您还未部署华为 ICP 解决方案，可登录 3.5 免费申请远程实验室，[免费](#)申请 ICP 环境，获取地址和用户名。按照下面介绍的方法[调试运行](#)程序。

调试运行

步骤 1 点击 IDE 工具栏上的绿色箭头，运行程序。

步骤 2 打开应用后，设置网络参数，点击登录，显示“登录成功”。



步骤 3 登录成功后，点击注销，显示“注销成功！”。



----结束

5 典型场景开发

- 5.1 场景一：登录与登出
- 5.2 场景二：TP 场景下的呼叫
- 5.3 场景三：MS 场景下的呼叫
- 5.4 场景四：设备控制

5.1 场景一：登录与登出

功能介绍

通过 eSDK 服务端提供的 REST 接口，完成用户的登录登出操作。

对于登录不稳定的情况，我们需要在服务器操作系统中将 net.ipv4.tcp.timetamps 设置为 0。

示例代码

```
//java code
//设置 filter
private IntentFilter filter;
filter = new IntentFilter();
filter.addAction(NotifyMessage.AUTH_MSG_ON_LOGIN);
filter.addAction(NotifyMessage.AUTH_MSG_ON_LOGOUT);

private LocalBroadcastManager localBroadcastManager =
LocalBroadcastManager.getInstance(this);

//初始化 receiver
private BroadcastReceiver receiver = new BroadcastReceiver()
{
@Override
public void onReceive(Context context, Intent intent)
{
String action = intent.getAction();
BroadMsg broadMsg = (BroadMsg) intent
```

```
.getSerializableExtra (NotifyMessage.CC_MSG_CONTENT);

if (NotifyMessage.AUTH_MSG_ON_LOGIN.equals(action))
{
    if ("0".equals(broadMsg.getRecode()))
    {
        //登录成功
    }
}
else if (NotifyMessage.AUTH_MSG_ON_LOGOUT.equals(action))
{
    if ("0".equals(broadMsg.getRecode()))
    {
        //注销成功
    }
}
else
{
    //注销失败
}
}
};

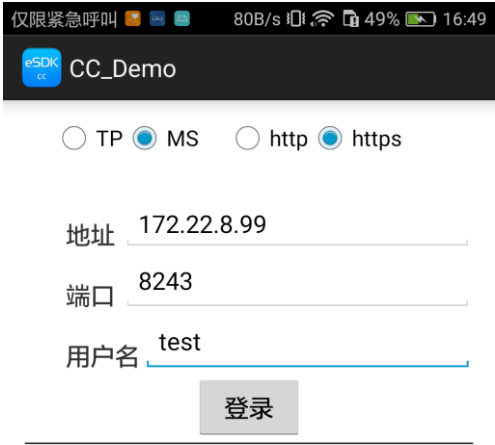
//监听广播
localBroadcastManager.registerReceiver(receiver, filter);

//登录操作
if (0 != MobileCC.getInstance().login("1", etName.getText().toString().trim()))
{
    //登录请求已发出
}
//注销操作
MobileCC.getInstance().logout();
```

Sample 界面

(此处以 MS 平台为例)

- 登录前界面



- 登录后界面



5.2 场景二：TP 场景下的呼叫

功能介绍

发起呼叫：用户发起一次呼叫。

结束呼叫：用户结束当前的呼叫。

示例代码

```
TP 呼叫代码示例：
//java code
private LocalBroadcastManager localBroadcastManager =
LocalBroadcastManager.getInstance(this);
//设置 filter
private IntentFilter filter;
filter = new IntentFilter();
filter.addAction(NotifyMessage.CALL MSG ON CONNECTED);
filter.addAction(NotifyMessage.CALL MSG REFRESH LOCALVIEW);
filter.addAction(NotifyMessage.CALL MSG REFRESH REMOTEVIEW);
filter.addAction(NotifyMessage.CALL MSG ON DISCONNECTED);

//初始化 receiver
private BroadcastReceiver receiver = new BroadcastReceiver()
```

```
{
    @Override
    public void onReceive(Context context, Intent intent)
    {
        String action = intent.getAction();
        BroadMsg broadMsg = (BroadMsg) intent
        .getSerializableExtra(NotifyMessage.CC_MSG_CONTENT);

        if (NotifyMessage.CALL_MSG_ON_CONNECTED.equals(action))
        {
            //与坐席接通
        }
        else if (NotifyMessage.CALL_MSG_REFRESH_LOCALVIEW.equals(action))
        {
            //刷新本地视屏
            MobileCC.getInstance().setVideoContainer(ChatActivity.this, localView, null);
        }
        else if (NotifyMessage.CALL_MSG_REFRESH_REMOTEVIEW.equals(action))
        {
            //刷新远端视屏
            MobileCC.getInstance().setVideoContainer(ChatActivity.this, null,
remoteView);
        }
        else if (NotifyMessage.CALL_MSG_ON_DISCONNECTED.equals(action))
        {
            //与坐席断开连接, 呼叫被释放
        }
    }
}

//注册 receiver

@Override
protected void onResume()
{
    super.onResume();
    localBroadcastManager.registerReceiver(receiver, filter);}

//呼叫操作
MobileCC.getInstance().makeCall("8888", MobileCC.SERVER_TP + "", "data", " ");

//释放呼叫
MobileCC.getInstance().releaseCall();
```

Sample 界面

- TP 呼叫



5.3 场景三：MS 场景下的呼叫

功能介绍

主要介绍在 MS 环境下的文字交谈、语音呼叫、会议，桌面共享等功能。NAT 穿越已经在 sdk 中做好，只要配置相应的参数即可。

如果出现语音连上就立即断开的情况，请到路由器的配置中，在“WAN 设置”中把 SIP ALG 开启。

若不需要验证码，可以在 IcsGateway 服务器中打开 `home/prometheus/tomcat7/webapps/icsgateway/WEB-INF/config/verifycode.properties` 这个文件，修改 `VERIFYCODE_ISUSERFORCALL = false`，然后重启 IcsGateway 服务器即可。

示例代码

- MS 中发送文字代码

```
//java code
MobileCC.getInstance().sendMsg("Help!Help!");
```

- MS 中语音呼叫

```
//java code
MobileCC.getInstance().makeCall(audioAccessCode, MobileCC.AUDIO_CALL, data,
etVerifycode.getText().toString());
```

- MS 中的视屏呼叫代码

```
//java code
private LocalBroadcastManager localBroadcastManager =
LocalBroadcastManager.getInstance(this);
//设置 filter
private IntentFilter filter;
filter = new IntentFilter();
filter.addAction(NotifyMessage.CALL_MSG_ON_CONNECTED);
filter.addAction(NotifyMessage.CALL_MSG_ON_DISCONNECTED);
filter.addAction(NotifyMessage.CALL_MSG_ON_APPLY_MEETING);
```



```
filter.addAction(NotifyMessage.CALL_MSG_USER_STATUS);
filter.addAction(NotifyMessage.CALL_MSG_ON_STOP_MEETING);

//初始化 receiver
private BroadcastReceiver receiver = new BroadcastReceiver()
{
    @Override
    public void onReceive(Context context, Intent intent)
    {
        String action = intent.getAction();
        BroadMsg broadMsg = (BroadMsg) intent
            .getSerializableExtra(NotifyMessage.CC_MSG_CONTENT);

        if (NotifyMessage.CALL_MSG_ON_CONNECTED.equals(action))
        {
            //与坐席接通
        }
        else if (NotifyMessage.CALL_MSG_USER_STATUS.equals(action))
        {
            //会议创建成功
            MobileCC.getInstance().setVideoContainer(MeetingActivity.this,
mLlLocalSurface, mLlRemoteSurface);
        }
        else if (NotifyMessage.CALL_MSG_ON_APPLY_MEETING.equals(action))
        {
            if (null == broadMsg.getRetCode())
            {
                Toast.makeText(MSChatActivity.this, "申请会议时失败, 错误码是: " +
broadMsg.getErrorCode(), Toast.LENGTH_SHORT).show();
            }
            else
            {
                String retcode = broadMsg.getRetCode();

                if (MobileCC.MESSAGE_OK.equals(retcode))
                {
                    Toast.makeText(MSChatActivity.this, "申请会议成功",
Toast.LENGTH_SHORT).show();
                }
                else
                {
                    Toast.makeText(MSChatActivity.this, "申请会议失败, 错误码是: " +
retcode, Toast.LENGTH_SHORT).show();
                }
            }
        }
        else if (NotifyMessage.CALL_MSG_ON_DISCONNECTED.equals(action))
        {
            //与坐席断开连接, 呼叫被释放
        }
    }
}
```

```

        else if (NotifyMessage.CALL_MSG_ON_STOP_MEETING.equals(action))
        {
            //退出会议
        }
    }
}

//注册 receiver

@Override
protected void onResume()
{
    super.onResume();
    localBroadcastManager.registerReceiver(receiver, filter);
}

//视屏会议呼叫操作
MobileCC.getInstance().makeCall("60021", MobileCC.VIDEO_CALL, data. " ");

//释放呼叫
MobileCC.getInstance().releaseCall();

```

● MS 中的桌面共享代码

```

//java code
private LocalBroadcastManager localBroadcastManager =
LocalBroadcastManager.getInstance(this);
//创建 filter
IntentFilter filter = new IntentFilter();
filter.addAction(NotifyMessage.CALL_MSG_USER_RECEIVE_SHARED_DATA);

//初始化 receiver

private BroadcastReceiver receiver = new BroadcastReceiver()
{
    @Override
    public void onReceive(Context context, Intent intent)
    {
        String action = intent.getAction();
        BroadMsg broadMsg = (BroadMsg) intent
            .getSerializableExtra(NotifyMessage.CC_MSG_CONTENT);

        if (NotifyMessage.CALL_MSG_USER_RECEIVE_SHARED_DATA.equals(action))
        {
            //接收到共享通知
        }
    }
}

//注册 receiver

@Override
protected void onResume()
{
    super.onResume();
    localBroadcastManager.registerReceiver(receiver, filter);
}

```

```
}  
  
//加载共享画面  
  
MobileCC.getInstance().setDesktopShareContainer(getBaseContext(),  
        desktopSharedLayout); // 接收共享数据，设置显示容器
```

Sample 界面

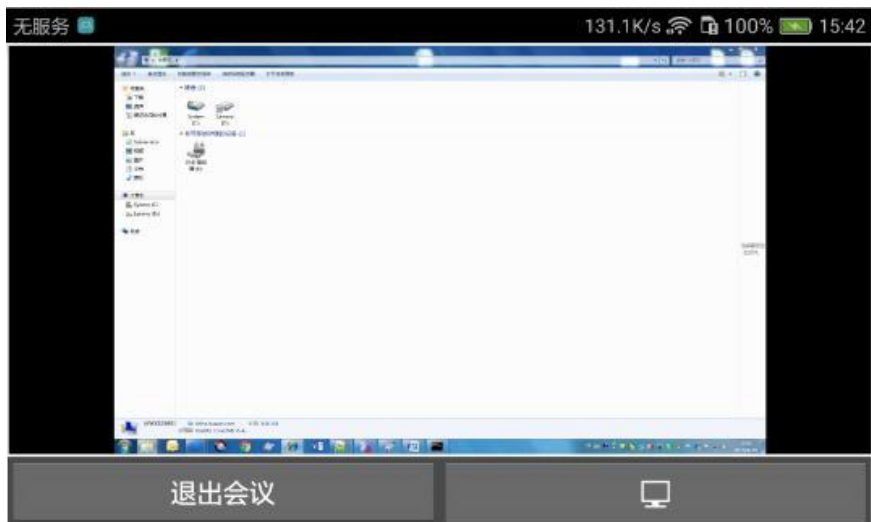
- MS 中的文字链接和语音呼叫



- MS 视屏界面



- MS 中的桌面共享界面



5.4 场景四：设备控制

功能介绍

通话过程中摄像头切换、扬声器/听筒模式切换、麦克风静音、扬声器静音。

代码示例：

- 摄像头切换

```
//java code
MobileCC.getInstance().switchCamera();
```

- 扬声器/听筒模式切换

```
//java code
//扬声器模式
MobileCC.getInstance().changeAudioRoute(MobileCC.getInstance().AUDIO_ROUTE_SPEAKER);

//听筒模式
MobileCC.getInstance().changeAudioRoute(MobileCC.getInstance().AUDIO_ROUTE_RECEIVER);
```

- 麦克风静音

```
//java code
//麦克风静音
MobileCC.getInstance().setMicMute(true);

//恢复麦克风
MobileCC.getInstance().setMicMute(false);
```

- 扬声器静音

```
//java code
//扬声器静音
MobileCC.getInstance().setSpeakerMute(true);
```

```
//恢复扬声器  
MobileCC.getInstance().setSpeakerMute(false);
```

6 问题定位

错误信息查询方法

在接口参考文档里，列出了所有错误码信息。
根据接口返回的错误码，查询相对应的错误描述。

目录(C) 搜索(S)

概述

事件通知

初始化类接口

认证类接口

呼叫类接口

计费类接口

错误码

接口记录

错误码	说明
0	成功。
-1	参数错误。
-2	未登录。
-3	语音未连接。
-4	服务器响应错误。
-5	网络故障。
10-100-002	服务器端Platform.properties中的EVENT_THRESHOLD配置错误。
10-100-003	没有连接到CCS，Waz服务不可用。

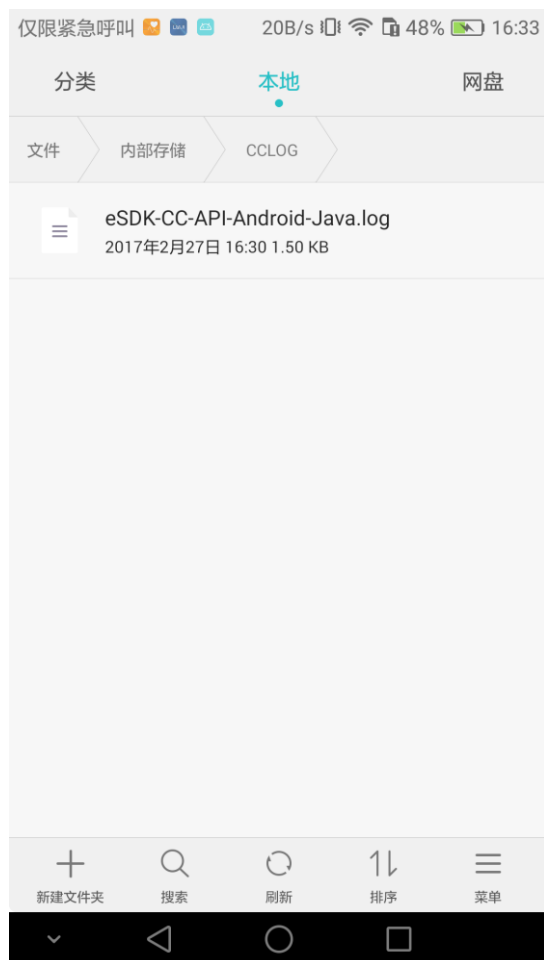
日志获取方法

接口返回

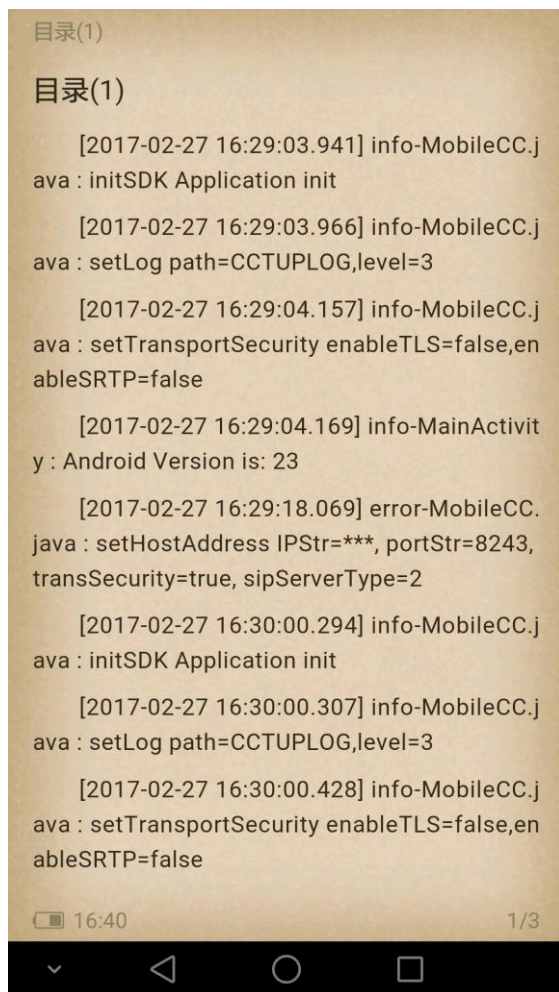
有些接口调用后，无论调用成功还是失败，会有一个返回值或者事件通知。

日志获取

eSDK 的接口返回值可以通过查看接口日志文件获取。
日志文件默认生成在/sdcard/CCLOG/eSDK-CC-API-Android-Java.log 目录下。



打开日志可以查看详细的接口调用结果。



日志分析

以设置网关接口的调用方法为例，全局搜索关键字“setHostAddress”，搜索到对应的记录，在该条记录时间之后出现 **error**，说明该接口调用失败。

```
[2017-02-27 16:29:18.069] error-MobileCC.java : setHostAddress IPStr=***, portStr=8243, transSecurity=true, sipServerType=2
```

如果在该条记录时间之后出现 **info**，说明接口调用失败。

```
[2017-02-27 16:30:05.521] info-MobileCC.java : setHostAddress ipStr=***, portStr=8280, transSecurity=false, sipServerType=2
```

对于非接口日志，比如程序运行的日志，需要根据具体的情况分析即可。

7 修订记录

发布日期	文档版本	描述
2017-3-15	V200R001C10	文档 V200R001C10 版本发布。 全量适配 TP&MS 功能。
2017-3-8	1.1.T1	文档 1.1.T1 版本发布。 增加： <ul style="list-style-type: none">5.3 场景三：MS 场景下的呼叫MS NAT 功能 修改：5.4 场景四：设备控制，
2016-12-31	V2.1.00	第一次正式发布。

8 附录

8.1 Hello World 源码文件

8.1 Hello World 源码文件

8.1.1 CCApplication.java

```
//java code
package com.huawei.esdk.icsdemo.application;
import android.app.Application;
import com.huawei.esdk.cc.MobileCC;
public class CCApplication extends Application
{

    @Override
    public void onCreate()
    {
        super.onCreate();
        MobileCC.getInstance().initSDK(this);
        MobileCC.getInstance().setLog("CCTUPLOG", 3);
    }

    @Override
    public void onTerminate()
    {
        super.onTerminate();
        MobileCC.getInstance().unInitSDK(); // 停止 SDK 服务
    }
}
```

8.1.2 MainActivity.java

```
//java code
package com.huawei.esdk.icsdemo;
import android.app.Activity;
import android.content.BroadcastReceiver;
import android.content.Context;
```

```
import android.content.Intent;
import android.content.IntentFilter;
import android.os.Bundle;
;
import android.support.v4.content.LocalBroadcastManager;
import android.view.View;
import android.view.inputmethod.InputMethodManager;
import android.widget.Button;
import android.widget.EditText;

import android.widget.Toast;

import com.huawei.esdk.cc.MobileCC;
import com.huawei.esdk.cc.common.BroadMsg;
import com.huawei.esdk.cc.common.NotifyMessage;
import com.huawei.esdk.myapplication.R;

public class MainActivity extends Activity implements View.OnClickListener
{
    private EditText etIP;
    private EditText etPort;
    private EditText etName;
    private Button btnLogin;
    private Button btnExit;
    private IntentFilter filter;
    private LocalBroadcastManager localBroadcastManager =
LocalBroadcastManager.getInstance(this);

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        initView();
    }

    /**
     * 初始化 view、监听和 filter
     */
    private void initView()
    {
        etIP = (EditText) findViewById(R.id.et_ip);
        etPort = (EditText) findViewById(R.id.et_port);
        etName = (EditText) findViewById(R.id.et_name);
        btnLogin = (Button) findViewById(R.id.btn_login);
        btnExit = (Button) findViewById(R.id.btn_exit);

        btnLogin.setOnClickListener(this);
        btnExit.setOnClickListener(this);

        filter = new IntentFilter();
        filter.addAction(NotifyMessage.AUTH_MSG_ON_LOGIN);
        filter.addAction(NotifyMessage.AUTH_MSG_ON_LOGOUT);
    }
}
```

```
        localBroadcastManager.registerReceiver(receiver, filter);
    }

    @Override
    public void onClick(View view)
    {
        switch (view.getId())
        {
            case R.id.btn_login:
                login();
                break;

            case R.id.btn_exit:
                MobileCC.getInstance().logout();
                break;

            default:
                break;
        }
    }

    private BroadcastReceiver receiver = new BroadcastReceiver()
    {
        @Override
        public void onReceive(Context context, Intent intent)
        {
            String action = intent.getAction();
            BroadMsg broadMsg = (BroadMsg) intent
                .getSerializableExtra(NotifyMessage.CC_MSG_CONTENT);

            if (NotifyMessage.AUTH_MSG_ON_LOGIN.equals(action))
            {
                //当 retcode 为空时, 通过 errorcode 提示用户
                if (null == broadMsg.getRequestCode().getRetCode())
                {
                    Toast.makeText(MainActivity.this, getString(R.string.login_fail) +
broadMsg.getRequestCode().getErrorCode(), Toast.LENGTH_SHORT).show();
                }
                //当 retcode 不为空时, 通过服务器返回的错误码来提示用户
                else
                {
                    if ("0".equals(broadMsg.getRequestCode().getRetCode()))
                    {
                        //登录成功
                        Toast.makeText(MainActivity.this,
getString(R.string.login_success), Toast.LENGTH_SHORT).show();
                    }
                    else
                    {
                        Toast.makeText(MainActivity.this, "登陆失败!" +
```

```

broadMsg.getRequestCode().getRetCode(), Toast.LENGTH_SHORT).show();
        }
    }
}
else if (NotifyMessage.AUTH_MSG_ON_LOGOUT.equals(action))
{
    if (null == broadMsg.getRequestCode().getRetCode())
    {
        Toast.makeText(MainActivity.this, getString(R.string.logout_fail) +
broadMsg.getRequestCode().getErrorCode(), Toast.LENGTH_SHORT).show();
    }
    else
    {
        //注销成功
        if ("0".equals(broadMsg.getRequestCode().getRetCode()))
        {
            Toast.makeText(MainActivity.this, "注销成功!
", Toast.LENGTH_SHORT).show();
        }
        else
        {
            Toast.makeText(MainActivity.this,
getString(R.string.logout_fail) + broadMsg.getRequestCode().getRetCode(),
Toast.LENGTH_SHORT).show();
        }
    }
}
}
};

private int setHostAddress()
{
    String ipStr = etIP.getText().toString();
    String portStr = etPort.getText().toString();
    return MobileCC.getInstance().setHostAddress(ipStr, portStr, true,
MobileCC.SERVER MS);
}

private void login()
{
    //收起小键盘
    ((InputMethodManager)
getSystemService(INPUT_METHOD_SERVICE)).hideSoftInputFromWindow(MainActivity.this.g
etCurrentFocus().getWindowToken(), InputMethodManager.HIDE_NOT_ALWAYS);
    if (0 == (setHostAddress()))
    {
        if (0 != MobileCC.getInstance().login("1",
etName.getText().toString().trim()))
        {
            Toast.makeText(MainActivity.this, "用户名不合法, 请重新输入",
Toast.LENGTH_SHORT).show();
        }
    }
    else
    {

```

```
        Toast.makeText(MainActivity.this, "网络参数错误",  
        Toast.LENGTH_SHORT).show();  
    }  
}  
@Override  
protected void onDestroy()  
{  
    localBroadcastManager.unregisterReceiver(receiver);  
    android.os.Process.killProcess(android.os.Process.myPid());  
    super.onDestroy();  
}  
}
```

8.1.3 build.gradle

该文件和 AndroidMainifest.xml 需要根据当前的编译环境做相应修改。

```
//java code  
buildscript {  
    repositories {  
        jcenter()  
    }  
    dependencies {  
        classpath 'com.android.tools.build:gradle:1.5.0'  
    }  
}  
  
apply plugin: 'com.android.application'  
  
dependencies {  
    compile fileTree(include: ['*.jar'], dir: 'libs')  
    compile 'com.android.support:support-v4:23.1.1'  
}  
  
android {  
    compileSdkVersion 19  
    buildToolsVersion '21.1.2'  
  
    buildTypes {  
        release {  
            minifyEnabled false  
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-  
rules.pro'  
        }  
    }  
    sourceSets {  
        main {  
            jniLibs.srcDirs = ['libs']  
        }  
    }  
}
```

8.1.4 AndroidMainifest.xml

```
//xml code  
<?xml version="1.0" encoding="utf-8"?>
```

```
<manifest package="com.huawei.esdk.icsdemo"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:versionCode="1"
    android:versionName="1.5.70">

    <uses-sdk
        android:minSdkVersion="15"
        android:targetSdkVersion="19"/>
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"></uses-
permission>
    <uses-permission android:name="android.permission.RECORD_AUDIO"></uses-
permission>
    <uses-permission android:name="android.permission.INTERNET"></uses-permission>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"></uses-
permission>
    <uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS"></uses-
permission>
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"></uses-permission>
    <uses-permission android:name="android.permission.WAKE_LOCK"></uses-permission>
    <uses-permission android:name="android.permission.BROADCAST_STICKY"></uses-
permission>
    <uses-permission android:name="android.permission.READ_PHONE_STATE"></uses-
permission>
    <uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW"/>
    <uses-permission android:name="android.permission.CAMERA"></uses-permission>

    <uses-feature
        android:name="android.hardware.camera"
        android:required="true"/>

    <application
        android:name="com.huawei.esdk.icsdemo.application.CCApplication"
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:hardwareAccelerated="false"
        android:theme="@style/AppTheme">
        <activity
            android:name=".activity.MainActivity"
            android:label="@string/app_name"
            android:theme="@style/AppTheme">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>

                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <service android:name="com.huawei.AudioDeviceAndroidService"/>
    </application>

</manifest>
```

8.1.5 values/strings.xml

```
//xml code
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app name">CC Demo</string>
    <string name="hello world">Hello world!</string>
    <string name="action settings">Settings</string>

    <string name="login fail">Login fail! errorCode is:</string>
    <string name="login success">Login success</string>
    <string name="logout fail">Logout fail! errorCode is:</string>

    <string name="address">Address</string>
    <string name="port">Port</string>
    <string name="name">Name</string>
    <string name="login">Login</string>
    <string name="logout">Logout</string>

</resources>
```

8.1.6 values-zh/strings.xml

```
//xml code
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app name">CC Demo</string>
    <string name="hello world">Hello world!</string>
    <string name="action settings">Settings</string>

    <string name="login fail">登录失败，错误码是：</string>
    <string name="login success">登录成功</string>
    <string name="logout fail">注销失败，错误码是：</string>

    <string name="address">地址</string>
    <string name="port">端口</string>
    <string name="name">用户名</string>
    <string name="login">登录</string>
    <string name="logout">注销</string>

</resources>
```

8.1.7 values/styles.xml

```
//xml code
<resources>
<!--
Base application theme, dependent on API level. This theme is replaced
```



```

by AppBaseTheme from res/values-vXX/styles.xml on newer devices.
-->
<style name="AppBaseTheme" parent="android:Theme.Light">
<!--
Theme customizations available in newer API levels can go in
res/values-vXX/styles.xml, while customizations related to
backward-compatibility can go here.
-->
</style>

<!-- Application theme. -->
<style name="AppTheme" parent="AppBaseTheme">
<!-- All customizations that are NOT specific to a particular API-level can go here.
-->
</style>

</resources>

```

8.1.8 layout/activity_main.xml

```

// xml code
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".activity.MainActivity">

    <TextView
        android:id="@+id/tv_ip"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingLeft="30dp"
        android:paddingTop="70dp"
        android:text="@string/address"
        android:textSize="26dp"/>

    <EditText
        android:id="@+id/et_ip"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@+id/tv_ip"
        android:layout_marginLeft="5dp"
        android:layout_toRightOf="@+id/tv_ip"
        android:singleLine="true"
        android:ellipsize="start"
        android:text="10.170.197.219"

        android:textSize="14dp"/>

    <TextView
        android:id="@+id/tv_port"

```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/tv_ip"
        android:layout_marginTop="20dp"
        android:paddingLeft="30dp"
        android:text="@string/port"
        android:textSize="26dp"/>

<EditText
    android:id="@+id/et_port"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/tv_port"
    android:layout_below="@+id/tv_ip"
    android:layout_marginLeft="5dp"
    android:layout_toRightOf="@+id/tv_port"
    android:singleLine="true"
    android:ellipsize="start"
    android:text="8243"
    android:textSize="14dp"/>

<TextView
    android:id="@+id/tv_name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/tv_port"
    android:layout_marginTop="20dp"
    android:paddingLeft="30dp"
    android:text="@string/name"
    android:textSize="26dp"/>

<EditText
    android:id="@+id/et_name"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/tv_name"
    android:layout_below="@+id/tv_port"
    android:layout_marginLeft="2dp"
    android:layout_toRightOf="@+id/tv_name"
    android:editable="true"
    android:singleLine="true"
    android:ellipsize="start"
    android:textSize="14dp"/>

<Button
    android:id="@+id/btn_login"
    android:layout_width="120dp"
    android:layout_height="wrap_content"
    android:layout_below="@+id/tv_name"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="15dp"
    android:text="@string/login"/>

<Button
```

```
        android:id="@+id/btn_exit"
        android:layout_width="120dp"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:text="@string/logout"/>
    </RelativeLayout>
```