SMS Messaging in Android

Photo by macinate

POSTED BY WEIMENGLEE - 12 MAR 2009

It would be safe to say that nearly every mobile phone sold in the past decade has SMS messaging capabilities. In fact, SMS messaging is one great killer application for the mobile phone and it has created a steady revenue stream for mobile operators. Understanding how to use SMS messaging in your application can provide you with many ideas to create the next killer application.

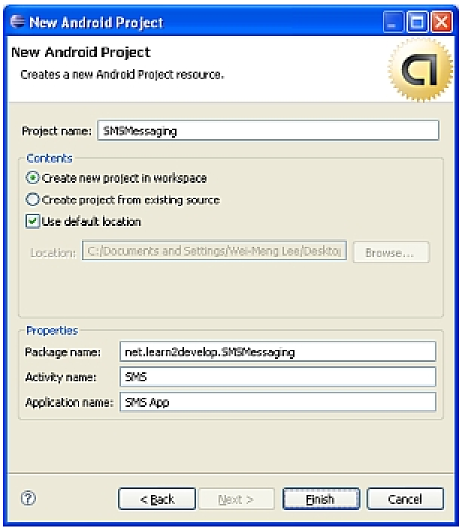
See also...

- Getting Started with Android Development
- Getting Started with Pebble Development
- FIFA World Cup Mobile Site
- Developer Economics: take the survey and win
- Android: platform of the next decade or revenue backwater?

In this article, we take a look at how you can programmatically send and receive SMS messages in your Android applications. The good news for Android developers is that you don't need a real device to test out SMS messaging - the free Android emulator provides the capability to do so.

Sending SMS Messages

To get started, first launch Eclipse and create a new Android project. Name the project as shown in Figure 1.


New Android Project

New Android Project
Creates a new Android Project resource.

Project name: SMSMessaging

Contents
● Create new project in workspace
○ Create project from existing source
☑ Use default location
Location: C:/Documents and Settings/Wei-Meng Lee/Desktop Browse...

Properties
Package name: net.learn2develop.SMSMessaging
Activity name: SMS
Application name: SMS App

< Back Next > Finish Cancel

***Figure 1** Creating a new Android project using Eclipse*

Android uses a permission-based policy where all the permissions needed by an application need to be specified in the AndroidManifest.xml file. By doing so, when the application is installed it will be clear to the user what specific access permissions are required by the application. For example, as sending SMS messages will potentially incur additional cost on the user's end, indicating the SMS permissions in the AndroidManifest.xml file will let the user decide whether to allow the application to install or not.

In the AndroidManifest.xml file, add the two permissions - SEND_SMS and RECEIVE_SMS:

IN THIS SECTION

- HTML5
- Android
- iPhone
- Wearables
- Touch
- Content Adaptation
- Emulators
- Messaging
- Geolocation
- Device Detection
- Responsive Design
- Browsers

MORE >

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="net.learn2develop.SMSMessaging"
      android:versionCode="1"
      android:versionName="1.0.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".SMS"
                  android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-permission android:name="android.permission.SEND_SMS">
    </uses-permission>
    <uses-permission android:name="android.permission.RECEIVE_SMS">
    </uses-permission>
</manifest>
```

In the main.xml file located in the res/layout folder, add the following code so that the user can enter a phone number as well as a message to send:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Enter the phone number of recipient"
        />
    <EditText
        android:id="@+id/txtPhoneNo"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        />
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Message"
        />
    <EditText
        android:id="@+id/txtMessage"
        android:layout_width="fill_parent"
        android:layout_height="150px"
        android:gravity="top"
        />
    <Button
        android:id="@+id/btnSendSMS"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Send SMS"
        />
</LinearLayout>
```

The above code creates the UI shown in Figure 2.

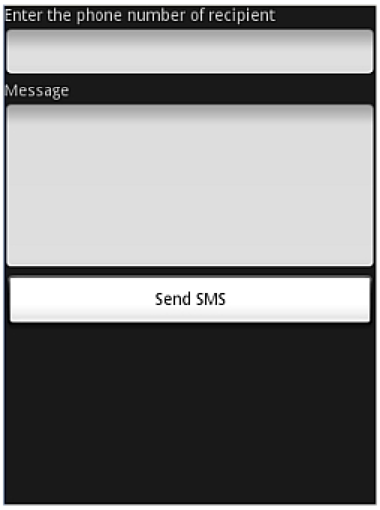


Figure 2 *Creating the UI for sending SMS messages*

Next, in the SMS activity, we wire up the Button view so that when the user clicks on it, we will check to see that the phone number of the recipient and the message is entered before we send the message using the sendSMS() function, which we will define shortly:

```
package net.learn2develop.SMSMessaging;

import android.app.Activity;
import android.app.PendingIntent;
import android.content.Intent;
import android.os.Bundle;
import android.telephony.gsm.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class SMS extends Activity
{
    Button btnSendSMS;
    EditText txtPhoneNo;
    EditText txtMessage;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        btnSendSMS = (Button) findViewById(R.id.btnSendSMS);
        txtPhoneNo = (EditText) findViewById(R.id.txtPhoneNo);
        txtMessage = (EditText) findViewById(R.id.txtMessage);

        btnSendSMS.setOnClickListener(new View.OnClickListener()
        {
            public void onClick(View v)
            {
                String phoneNo = txtPhoneNo.getText().toString();
                String message = txtMessage.getText().toString();
                if (phoneNo.length()>0 && message.length()>0)
                    sendSMS(phoneNo, message);
                else
                    Toast.makeText(getBaseContext(),
                        "Please enter both phone number and message.",
                        Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

The sendSMS() function is defined as follows:

```
public class SMS extends Activity
{
    //...

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        //...
    }

    //---sends an SMS message to another device---
    private void sendSMS(String phoneNumber, String message)
    {
        PendingIntent pi = PendingIntent.getActivity(this, 0,
            new Intent(this, SMS.class), 0);
        SmsManager sms = SmsManager.getDefault();
        sms.sendTextMessage(phoneNumber, null, message, pi, null);
    }
}
```

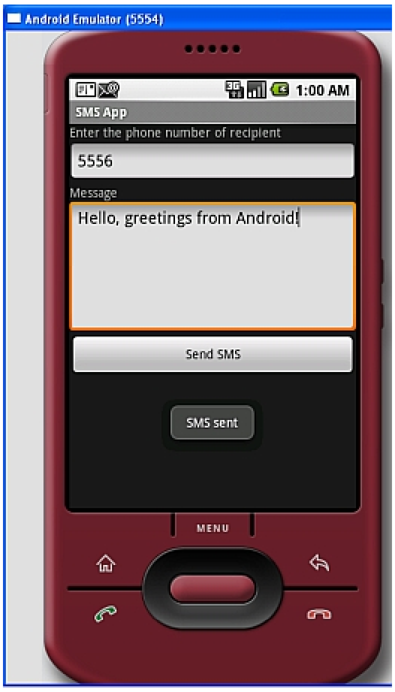
To send an SMS message, you use the SmsManager class. Unlike other classes, you do not directly instantiate this class; instead you will call the getDefault() static method to obtain an SmsManager object. The sendTextMessage() method sends the SMS message with a PendingIntent. The PendingIntent object is used to identify a target to invoke at a later time. For example, after sending the message, you can use a PendingIntent object to display another activity. In this case, the PendingIntent object (pi) is simply pointing to the same activity (SMS.java), so when the SMS is sent, nothing will happen.

If you need to monitor the status of the SMS message sending process, you can actually use two PendingIntent objects together with two BroadcastReceiver objects, like this:

```
//---sends an SMS message to another device---
private void sendSMS(String phoneNumber, String message)
{
    String SENT = "SMS_SENT";
    String DELIVERED = "SMS_DELIVERED";

    PendingIntent sentPI = PendingIntent.getBroadcast(this, 0,
        new Intent(SENT), 0);

    PendingIntent deliveredPI = PendingIntent.getBroadcast(this, 0,
        new Intent(DELIVERED), 0);

    //---when the SMS has been sent---
    registerReceiver(new BroadcastReceiver(){
        @Override
        public void onReceive(Context arg0, Intent arg1) {
            switch (getResultCode())
            {
                case Activity.RESULT_OK:
                    Toast.makeText(getBaseContext(), "SMS sent",
                            Toast.LENGTH_SHORT).show();
                    break;
                case SmsManager.RESULT_ERROR_GENERIC_FAILURE:
                    Toast.makeText(getBaseContext(), "Generic failure",
                            Toast.LENGTH_SHORT).show();
                    break;
                case SmsManager.RESULT_ERROR_NO_SERVICE:
                    Toast.makeText(getBaseContext(), "No service",
                            Toast.LENGTH_SHORT).show();
                    break;
                case SmsManager.RESULT_ERROR_NULL_PDU:
                    Toast.makeText(getBaseContext(), "Null PDU",
                            Toast.LENGTH_SHORT).show();
                    break;
                case SmsManager.RESULT_ERROR_RADIO_OFF:
                    Toast.makeText(getBaseContext(), "Radio off",
                            Toast.LENGTH_SHORT).show();
                    break;
            }
        }
    }, new IntentFilter(SENT));

    //---when the SMS has been delivered---
    registerReceiver(new BroadcastReceiver(){
        @Override
        public void onReceive(Context arg0, Intent arg1) {
            switch (getResultCode())
            {
                case Activity.RESULT_OK:
                    Toast.makeText(getBaseContext(), "SMS delivered",
                            Toast.LENGTH_SHORT).show();
                    break;
                case Activity.RESULT_CANCELED:
                    Toast.makeText(getBaseContext(), "SMS not delivered",
                            Toast.LENGTH_SHORT).show();
                    break;
            }
        }
    }, new IntentFilter(DELIVERED));

    SmsManager sms = SmsManager.getDefault();
    sms.sendTextMessage(phoneNumber, null, message, sentPI, deliveredPI);
}
```

The above code uses a PendingIntent object (sentPI) to monitor the sending process. When an SMS message is sent, the first BroadcastReceiver's onReceive event will fire. This is where you check the status of the sending process. The second PendingIntent object (deliveredPI) monitors the delivery process. The second BroadcastReceiver's onReceive event will fire when an SMS is successfully delivered.

You can now test the application by pressing F11 in Eclipse. To send an SMS message from one emulator instance to another, simply launch another instance of the Android emulator by going to the Tools folder of the SDK and running Emulator.exe.



Android Emulator (5554)

SMS App

Enter the phone number of recipient: 5556

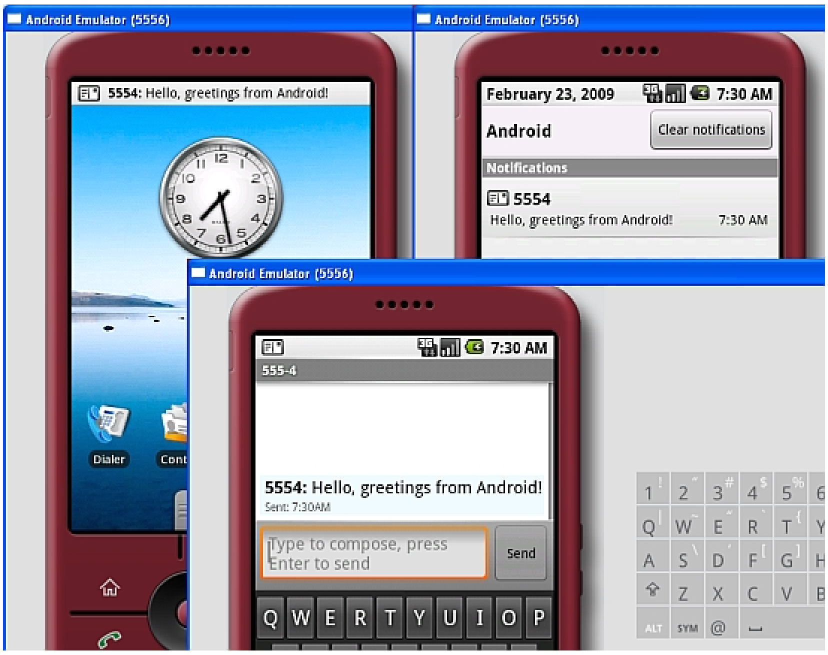
Message: Hello, greetings from Android!

SMS sent

Figure 3 *Sending an SMS message*

Figure 3 shows how you can send an SMS message from one emulator to another; simply use the target emulator's port number (shown in the top left corner of the window) as its phone number. When an SMS is sent successfully, it will display a "SMS sent" message. When it is successfully delivered, it will display a "SMS delivered" message. Note that for testing using the emulator, when an SMS is successfully delivered, the "SMS delivered" message does not appear; this only works for real devices.

Figure 4 shows the SMS message received on the recipient emulator. The message first appeared in the notification bar (top of the screen). Dragging down the notification bar reveals the message received. To view the entire message, click on the message.



Android Emulator (5556)

5554: Hello, greetings from Android!

February 23, 2009 7:30 AM

Android

Notifications

5554
Hello, greetings from Android! 7:30 AM

555-4

5554: Hello, greetings from Android!
Sent: 7:30AM

Figure 4 *The SMS message received by the Android emulator*

If you do not want to go through all the trouble of sending the SMS message yourself, you can use an Intent object to help you send an SMS message. The following code shows how you can invoke the built-in SMS application to help you send an SMS message:

```
Intent sendIntent = new Intent(Intent.ACTION_VIEW);
sendIntent.putExtra("sms_body", "Content of the SMS goes here...");
sendIntent.setType("vnd.android-dir/mms-sms");
startActivity(sendIntent);
```

Figure 5 shows the built-in SMS application invoked to send the SMS message.

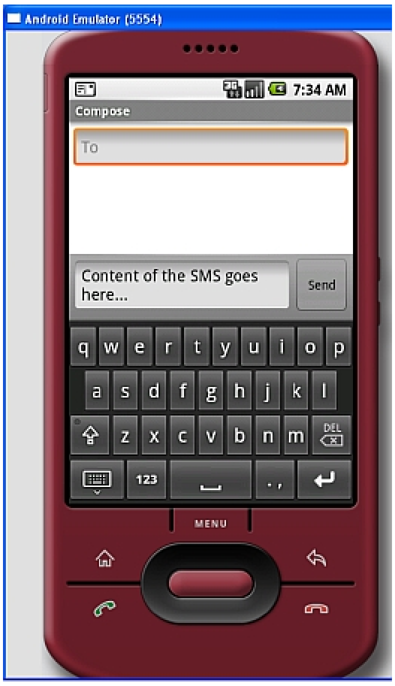


Figure 5 *Invoking the built-in SMS application*

Receiving SMS Messages

Besides programmatically sending SMS messages, you can also intercept incoming SMS messages using a BroadcastReceiver object.

To see how to receive SMS messages from within your Android application, in the AndroidManifest.xml file add the <receiver> element so that incoming SMS messages can be intercepted by the SmsReceiver class:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="net.learn2develop.SMSMessaging"
      android:versionCode="1"
      android:versionName="1.0.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".SMS"
                  android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <receiver android:name=".SmsReceiver">
            <intent-filter>
                <action android:name=
                    "android.provider.Telephony.SMS_RECEIVED" />
            </intent-filter>
        </receiver>

    </application>
    <uses-permission android:name="android.permission.SEND_SMS">
    </uses-permission>
    <uses-permission android:name="android.permission.RECEIVE_SMS">
    </uses-permission>
</manifest>
```

Add a new class file to your project and name it as SmsReceiver.java (see Figure 6).

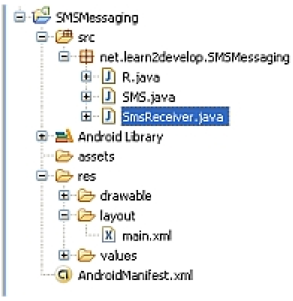


Figure 6*Adding the SmsReceiver.java file to the project*

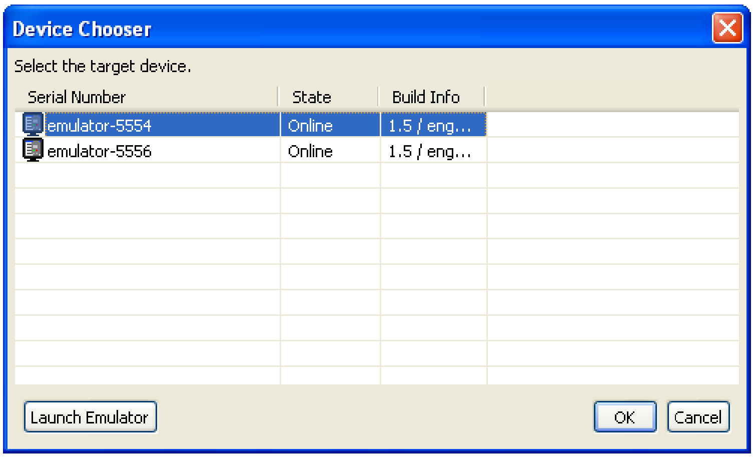
In the SmsReceiver class, extend the BroadcastReceiver class and override the onReceive() method:

```
package net.learn2develop.SMSMessaging;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;

public class SmsReceiver extends BroadcastReceiver
{
 @Override
 public void onReceive(Context context, Intent intent)
       {
 }
}
```

When SMS messages are received, the onCreate() method will be invoked. The SMS message is contained and attached to the Intent object (intent - the second parameter in the onReceive() method) via a Bundle object. The messages are stored in an Object array in the PDU format. To extract each message, you use the static createFromPdu() method from the SmsMessage class. The SMS message is then displayed using the Toast class:

```
package net.learn2develop.SMSMessaging;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.telephony.gsm.SmsMessage;
import android.widget.Toast;

public class SmsReceiver extends BroadcastReceiver
{
    @Override
    public void onReceive(Context context, Intent intent)
    {
        //---get the SMS message passed in---
        Bundle bundle = intent.getExtras();
        SmsMessage[] msgs = null;
        String str = "";
        if (bundle != null)
        {
            //---retrieve the SMS message received---
            Object[] pdus = (Object[]) bundle.get("pdus");
            msgs = new SmsMessage[pdus.length];
            for (int i=0; i<msgs.length; i++){
                msgs[i] = SmsMessage.createFromPdu((byte[])pdus[i]);
                str += "SMS from " + msgs[i].getOriginatingAddress();
                str += " :";
                str += msgs[i].getMessageBody().toString();
                str += "\n";
            }
            //---display the new SMS message---
            Toast.makeText(context, str, Toast.LENGTH_SHORT).show();
        }
    }
}
```

That's it! To test the application, press F11 in Eclipse. Deploy the application to each Android emulator. Figure 7 shows Eclipse showing the emulators currently running. All you need to do is to select each emulator and deploy the application onto each one.



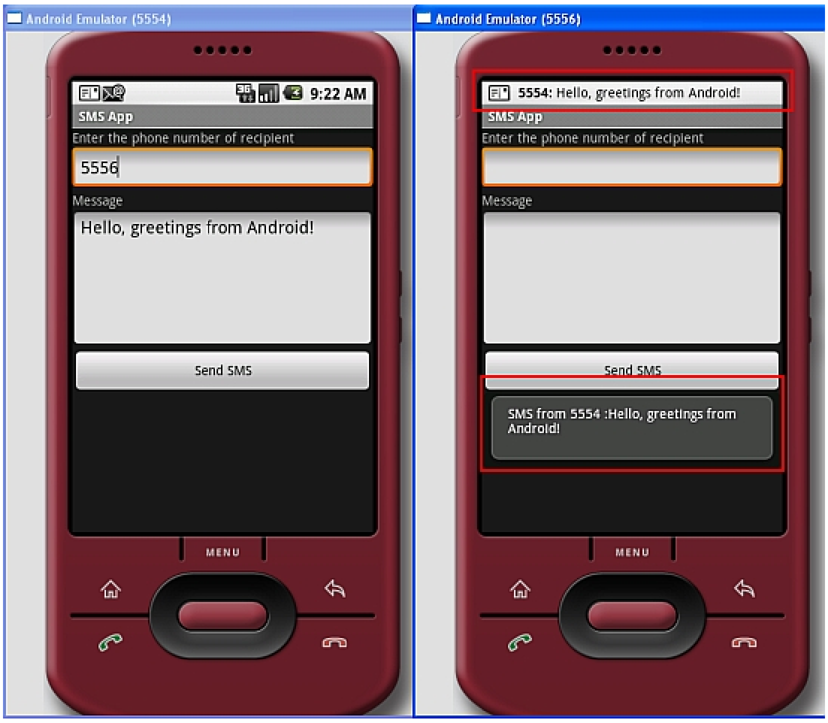
Device Chooser

Select the target device.

Serial Number	State	Build Info
emulator-5554	Online	1.5 / eng…
emulator-5556	Online	1.5 / eng…

Figure 7 *Selecting an emulator/device to deploy the application onto*

Figure 8 shows that when you send an SMS message to another emulator instance (port number 5556), the message is received by the target emulator and displayed via the Toast class.



Android Emulator (5554)
9:22 AM
SMS App
Enter the phone number of recipient
5556
Message
Hello, greetings from Android!
Send SMS
MENU
Android Emulator (5556)
5554: Hello, greetings from Android!
SMS App
Enter the phone number of recipient
Message
Send SMS
SMS from 5554 :Hello, greetings from Android!
MENU

Figure 8 *Sending and receiving SMS messages using the Android emulators*

Summary

In this article, you have seen how you can send and receive SMS messages programmatically from within your Android application. The capability to send and receive SMS messages is very useful as you can build very compelling applications. As an example, you can build a location tracker application where you can send a secret-coded SMS message to a device and when the device receives the secret SMS message it will reply with another SMS message containing its current geographical location using its built-in GPS receiver. How cool is that?!

Click the link below to download project source code

Attachment	Size
SMSMessaging.zip	35.43 KB

POSTED BY WEIMENGLEE - 12 MAR 2009

Wei-Meng Lee is a technologist and the founder of Developer Learning Solutions (**http://www.learn2develop.net**), a company focusing on hands-on training on the latest technology. Wei-Meng specializes in mobile technologies and has written several books on .NET, VB, C#, and .NET Compact Framework and is currently working on an Android book for Wrox. Contact Wei-Meng Lee at **weimenglee@learn2develop.net**.

COMMENTS

POSTED BY D.AUMENTA 5 YEARS AGO

Hi,I want use the intent to send the sms,but where I must put the code?Can you help me?

POSTED BY SCAREDYCAT 5 YEARS AGO

I'm having real problems getting started with android development. I'm trying to learn by trying out some examples making sure they work first. I'm having a couple of problems with your examples here. I've copied the code, manifest etc into a new project but when I attempt to compile I get 4 errors (actually appear to be 2 errors one for each registerReceiver). The errors are:

BroadcastReceiver cannot be resolved to a type line 59 Java Problem

and

IntentFilter cannot be resolved to a type line 86 Java Problem

The appropriate lines are:

registerReceiver(new BroadcastReceiver(){

and
}, new IntentFilter(SENT));

These errors are repeated for the DELIVERED intent too..

Am I missing something really obvious? Can you point me in the general direction of the problem.

Thanks

SC

POSTED BY WEIMENGLEE 5 YEARS AGO

Hi SC:
I am not at my computer right now, but if you can send me your email address, I would be happy to send you the source code tomorrow. My email is **weimenglee@gmail.com**. Meantime, please double check to see if you have keyed in the AndroidManifest.xml file correctly.

Thanks!
Wei-Meng Lee

POSTED BY SCAREDYCAT 5 YEARS AGO

> **weimenglee wrote:**
>
> Hi SC:
> I am not at my computer right now, but if you can send me your email address, I would be happy to send you the source code tomorrow. My email is **weimenglee@gmail.com**. Meantime, please double check to see if you have keyed in the AndroidManifest.xml file correctly.
>
> Thanks!
> Wei-Meng Lee

I've sent you an email, thanks in advance

SC

POSTED BY YOMIOTEBOLAKU 5 YEARS AGO

Hello,

This is an interesting tutorial. I am just getting to know about android and it has been difficult getting a comprehensive tutorial to start with. Finding this tutorials you give here is a big relief. Please, could you kindly make the source code available as I am having problem running these codes here.

Thank you

POSTED BY AHSAN78 5 YEARS AGO

Hi, nice tutorial... thanks..
But i was trying a bit different thing...
is it possible to receive a sms only from a specific application and the sms receive does not show default notification and not stored in inbox?

please let me know if you have any work around for this.

For clarification of the scenario:

A sends sms to B. B has an application x. Only x will get notified about the sms. and B will not show any sms receive notification and will not stor it in inbox. only application x will know about the sms.

POSTED BY ATE 5 YEARS AGO

Hi,
I'am also have the same problems like SC,and i have hard to resolved them for a long time,but the Error still occured;
so,can you help me ?
thanks a lot!

E_mail:henbell2@hotmail.com

Thanks again!

POSTED BY ATE 5 YEARS AGO

Hi.

I have got your SMSMessaging package,and the SMSMessaging App is ok now on emulator;

Thanks!

POSTED BY Q2HEADACHE 5 YEARS AGO

Great tutorial - thanks!

Now that I have this working, I would like to take the sms message received in the BroadcastReceiver, and send it to my Activity so it can update the main screen. Any suggestions on how to do that?

POSTED BY LIFEF 4 YEARS AGO

i use sdk_window_1.5, run app the emulator run normal, but my problem is i can't run emulator.exe in tools folder in sdk 1.5, may its have problem with emulator? so i used the emulator.exe of sdk_1.0 and it run, =)), very interested!!

MrQ

POSTED BY STANLICK 4 YEARS AGO

Great article bro! When would (should) you *not* use the built-in Apps via Intents? Isn't that what they are there for? I see many places where developers are rolling their own solutions and this seems like recreating the wheel to me. If the issue is to change the L&F, this sort of goes against the grain of a common UI too.

Peace,
Scott

POSTED BY KAPNKORE 4 YEARS AGO

is it possible to start a new activity on receive of sms?means i want to startActivity(intent) in SmsReceiver class.but as this class is not a Activity class how to pass context to start new activity?
Please let me know this will be great help for **me.Kapnkore@gmail.com**

POSTED BY APPLEIPHONE 4 YEARS AGO

Hi, congratulation for the tutorial, it's very easy to understand. I am trying to make a little hack to your code. I would like to realize an sms application:
when an sms is received a new thread is lunched, it's an UDP client that should talk whit an UDP Java server (running on a desktop pc) via socket.
The code is attached, please rename the file to .tar.gz.
All the class compiles well and also runs on the emulator, but the client seems to be unable to connect to the server. We tried to install the application on the terminal (htc hero with android 1.5) .
Could you please have a look to say if something is not ok?

Thanks in advance, Tony

Attachment	Size
SMSMessaging.txt	26.05 KB
ParserServer.txt	2.37 KB

POSTED BY HUUCONGIT 4 YEARS AGO

> **kapnkore wrote:**
>
> is it possible to start a new activity on receive of sms?means i want to startActivity(intent) in SmsReceiver class.but as this class is not a Activity class how to pass context to start new activity?
> Please let me know this will be great help for **me.Kapnkore@gmail.com**

Please help me. **jindoman87@yahoo.com**. Thanks

POSTED BY THEMORY 4 YEARS AGO

Hi,

Thanks a lot for this tutorial, that's perfectly work.
Is there any way to get MMS content ?

regards

POSTED BY RIPAN.SEKHON 4 YEARS AGO

Hello Everyone,

This is a very good tutorial site.
Can i anyone tell me, can i call in android automatically without opening its default calling interface. I want to show my application while calling is running in the backgroud.
Sorry for bad english.

Thanks & Regards

POSTED BY GOPAL_M 4 YEARS AGO

Hi
I am unable to start the 2nd instance of the simulator.
I started the application in eclipse but when i click the "**emulator.exe**" noting comes up.
I am using Android 2.1 SDK
Any pointers?

Thanks
GM

POSTED BY GOPAL_M 4 YEARS AGO

Hi
I managed to star the 2nd instance from command line ; with comand

```
    emulator -avd <name>
```

in my case

```
emulator -avd AVD_Android_2_1
```

But i cant see my application "SMS App" icon on the Launcher screen. I can see all other application.

GM

POSTED BY TRUE1 4 YEARS AGO

Hi. I tried to use this code and i have this kind of porblem:
I create their own files for this BroadcastReceiver. Maybe what's why it is work different, but i am sure it is not important.
So. First we register new receivers every time we send a Sms. So if i send 3 sms i will have 6 broadcast receivers in memory. (I noticed it, using log) We need register recievers in onCreate function (or onStart)

And Second. I tried this code on 2 emulators. Sms sends fine, but broadcast receiver, what responsible for delievery is never called. So i don't get delievery toast.
Is this emulator problem? This code work correct on phone?
Also i read something in android developers, what this delivery broadcast use some raw pdu, i don't get how to use it.
Someone can explain me pls?

please answer me. =) Email or here.
microalone@gmail.com

POSTED BY BILAL 4 YEARS AGO

Hi,I want use the Mobile set to send the sms,but where I must put the code?Can you help me?

POSTED BY TRUE1 4 YEARS AGO

> Quote:
>
> Hi,I want use the Mobile set to send the sms,but where I must put the code?Can you help me?

You should have some soft - eclipse + android sdk+ adt.The first qustetion is do you have them?

POSTED BY RCIOVATI 4 YEARS AGO

Thanks for this tutorial!
I need an advice about how to send sms and get the result synchronously. The methos should be like this:

```
.....
int result = myObj.sendSms(....);
....
```

Thanks very much :)

POSTED BY SHUBHAMPATNI86 4 YEARS AGO

Is it posiible to send email like sms from one Emulator to another one.......

now android giving cool air.....

POSTED BY SHUBHAMPATNI86 4 YEARS AGO

How To Delete Recent SMS from Inbox........
I use
context.getContentResolver().delete(Uri.parse("content://sms/conversations/" + (thread_id+1)),null, null);

and for thread_is i use cursor.getCount()/.getInt(0)/getInt(1).

bt it's not working...

now android giving cool air.....

POSTED BY SRILU 4 YEARS AGO

thank!!!!!!!!!!!this code is very useful to me in my project

POSTED BY KHANHTUNGNA 4 YEARS AGO

That is great! Thank you so much!

POSTED BY BILAL 4 YEARS AGO

can any one check my code its not working

```
package net.learn2develop.SMSMessaging1;


import android.app.Activity;
import android.app.PendingIntent;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.os.Bundle;
import android.telephony.gsm.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;


public class SMS extends Activity
{
    Button btnSendSMS;
    EditText txtPhoneNo;
    EditText txtMessage;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        btnSendSMS = (Button) findViewById(R.id.btnSendSMS);
        txtPhoneNo = (EditText) findViewById(R.id.txtPhoneNo);
        txtMessage = (EditText) findViewById(R.id.txtMessage);

        btnSendSMS.setOnClickListener(new View.OnClickListener()
        {
            public void onClick(View v)
            {
                String phoneNo = txtPhoneNo.getText().toString();
                String message = txtMessage.getText().toString();
                if (phoneNo.length()>0 && message.length()>0)
                    sendSMS(phoneNo, message);
                else
                    {Toast.makeText(getBaseContext(),
                        "Please enter both phone number and message.",
                        Toast.LENGTH_SHORT).show();
                    }

            }});
            }
    /**public class SMS extends Activity
    {
        //...

        /** Called when the activity is first created. */

      /*  //---sends an SMS message to another device---
      // private void sendSMS(String phoneNumber, String message)
        {
            PendingIntent pi = PendingIntent.getActivity(this, 0,
                new Intent(this, SMS.class), 0);
            SmsManager SMS = SmsManager.getDefault();
            sms.sendTextMessage(phoneNumber, null, message, pi, null);
      }


        }*/
    //monitor the status of the SMS message sending process
//---sends an SMS message to another device---
private void sendSMS(String phoneNumber, String message)
{
    String SENT = "SMS_SENT";
    String DELIVERED = "SMS_DELIVERED";

    PendingIntent sentPI = PendingIntent.getBroadcast(this, 0,
        new Intent(SENT), 0);

    PendingIntent deliveredPI = PendingIntent.getBroadcast(this, 0,
        new Intent(DELIVERED), 0);

    //---when the SMS has been sent---
    registerReceiver(new BroadcastReceiver(){
        @Override
        public void onReceive(Context arg0, Intent arg1) {
            switch (getResultCode())
            {
                case Activity.RESULT_OK:
                    Toast.makeText(getBaseContext(), "SMS sent",
                            Toast.LENGTH_SHORT).show();
                    break;
                case SmsManager.RESULT_ERROR_GENERIC_FAILURE:
                    Toast.makeText(getBaseContext(), "Generic failure",
                            Toast.LENGTH_SHORT).show();
                    break;
                case SmsManager.RESULT_ERROR_NO_SERVICE:
                    Toast.makeText(getBaseContext(), "No service",
                            Toast.LENGTH_SHORT).show();
                    break;
                case SmsManager.RESULT_ERROR_NULL_PDU:
                    Toast.makeText(getBaseContext(), "Null PDU",
                            Toast.LENGTH_SHORT).show();
                    break;
                case SmsManager.RESULT_ERROR_RADIO_OFF:
                    Toast.makeText(getBaseContext(), "Radio off",
                            Toast.LENGTH_SHORT).show();
                    break;
            }
        }
    }, new IntentFilter(SENT));

    //---when the SMS has been delivered---
    registerReceiver(new BroadcastReceiver(){
        @Override
        public void onReceive(Context arg0, Intent arg1) {
```

```
        public void onReceive(Context arg0, Intent arg1) {
            switch (getResultCode())
            {
                case Activity.RESULT_OK:
                    Toast.makeText(getBaseContext(), "SMS delivered",
                            Toast.LENGTH_SHORT).show();
                    break;
                case Activity.RESULT_CANCELED:
                    Toast.makeText(getBaseContext(), "SMS not delivered",
                            Toast.LENGTH_SHORT).show();
                    break;
            }
        }
    }, new IntentFilter(DELIVERED));

    SmsManager sms = SmsManager.getDefault();
    sms.sendTextMessage(phoneNumber, null, message, sentPI, deliveredPI);
}}// built-in SMS application to help you send an SMS message:
/*
Intent sendIntent = new Intent(Intent.ACTION_VIEW);
sendIntent.putExtra("sms_body", "Content of the SMS goes here...");
sendIntent.setType("vnd.android-dir/mms-sms");
startActivity(sendIntent);*/
```

POSTED BY RANDIKAH 4 YEARS AGO

hi,

this tutorial is good. and it works fine with the emulator. but when i put it in to a real device and use it the reeving device is getting two sms with the same content.

don't no what has gone wrong. even i have done a application by my own still it's same result.

could you look in to this.

this is what i have done.

```
package com.sms;

import android.app.Activity;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Toast;

public class sms extends Activity {
 /** Called when the activity is first created. */
 boolean b;

 @Override
 public void onCreate(Bundle savedInstanceState) {
  super.onCreate(savedInstanceState);

  Button btn = new Button(this);
  setContentView(btn);
  btn.setText("Send msg");
  btn.setOnClickListener(new OnClickListener() {

   @Override
   public void onClick(View v) {
    // TODO Auto-generated method stub
    if (!b) {
     try {
      sendSMS("0772376937", "Randika");
      Toast.makeText(sms.this, "SMS Sent", Toast.LENGTH_LONG)
        .show();
     } catch (Exception e) {
      // TODO Auto-generated catch block
      Toast.makeText(sms.this, e.getMessage(),
        Toast.LENGTH_LONG).show();
     }

    }
   }

  });

 }

 public void sendSMS(String number, String msg) throws Exception {
  if (!b) {
   SmsManager smsManager = SmsManager.getDefault();
   smsManager.sendTextMessage(number, null, msg, null, null);
  }
  b = true;
 }
}
```

mind you in emulator this works fine

regards,
Ranidka

POSTED BY BLUEFAIRY 4 YEARS AGO

Can someone please provide me the source code of this application.
Its really important and urgent to me!
my email is:usp_mkd@hotmail.com

it seems I have problem with the SMS class, where should this part take place:

```
//---sends an SMS message to another device---
private void sendSMS(String phoneNumber, String message)
{
String SENT = "SMS_SENT";
String DELIVERED = "SMS_DELIVERED";

PendingIntent sentPI = PendingIntent.getBroadcast(this, 0,
new Intent(SENT), 0);

PendingIntent deliveredPI = PendingIntent.getBroadcast(this, 0,
new Intent(DELIVERED), 0);

//---when the SMS has been sent---
registerReceiver(new BroadcastReceiver(){
@Override
public void onReceive(Context arg0, Intent arg1) {
switch (getResultCode())
{
case Activity.RESULT_OK:
Toast.makeText(getBaseContext(), "SMS sent",
Toast.LENGTH_SHORT).show();
break;
case SmsManager.RESULT_ERROR_GENERIC_FAILURE:
Toast.makeText(getBaseContext(), "Generic failure",
Toast.LENGTH_SHORT).show();
break;
case SmsManager.RESULT_ERROR_NO_SERVICE:
Toast.makeText(getBaseContext(), "No service",
Toast.LENGTH_SHORT).show();
break;
case SmsManager.RESULT_ERROR_NULL_PDU:
Toast.makeText(getBaseContext(), "Null PDU",
Toast.LENGTH_SHORT).show();
break;
case SmsManager.RESULT_ERROR_RADIO_OFF:
Toast.makeText(getBaseContext(), "Radio off",
Toast.LENGTH_SHORT).show();
break;
}
}
}, new IntentFilter(SENT));

//---when the SMS has been delivered---
registerReceiver(new BroadcastReceiver(){
@Override
public void onReceive(Context arg0, Intent arg1) {
switch (getResultCode())
{
case Activity.RESULT_OK:
Toast.makeText(getBaseContext(), "SMS delivered",
Toast.LENGTH_SHORT).show();
break;
case Activity.RESULT_CANCELED:
Toast.makeText(getBaseContext(), "SMS not delivered",
Toast.LENGTH_SHORT).show();
break;
}
}
}, new IntentFilter(DELIVERED));

SmsManager sms = SmsManager.getDefault();
sms.sendTextMessage(phoneNumber, null, message, sentPI, deliveredPI);
}
```

THANK YOU SO MUCH IN ADVANCE.
PLEASE SEND ME THE SOURCE CODE:
usp_mkd@hotmail.com

POSTED BY BLUEFAIRY 4 YEARS AGO

Can anyone please send me the sourcecode!
Please!
Its urgent!
Mr.Lee is not responding!

usp_mkd@hotmail.com

POSTED BY RANJANARR 4 YEARS AGO

Hello,
I am not able to run two emulators at the same time. Please help , the emulator at the eclispe however, not at the emulator.exe, when I click emulator.exe it would bring up the cmd window for a while and disappear.
reply to
ranjan.rar@gmail.com

POSTED BY LUIBBIO 4 YEARS AGO

in my main.xml i added a textview with id=@+id/txtMessaggio

in the receiver how i can update this textview after updated the TOAST?

thanks

POSTED BY DEBELIZMAJ 3 YEARS AGO

hi,
I have problem with this code. When i send sms, Android default messaging program receive sms! How to use my SMSReceiver???
Using Android 2.1 and Eclipse for developing

POSTED BY POZITIM 3 YEARS AGO

Thanks ! It is exactly what I was looking for and it works. Thanks for sharing !

POSTED BY MIDHUN_610 3 YEARS AGO

thanks

POSTED BY KIMMELJIN 3 YEARS AGO

Hi, thanks for this great tutorial. I got it working on the first try.

POSTED BY HARNESH 3 YEARS AGO

its not working properly so please send me right code its emergency for my collage project
please send me on my email id: **harnesh007@gmail.com**

POSTED BY PRONOUN149 3 YEARS AGO

i am getting an error that public final class R is already defined
I am new to java so i dont want to risk the code that much
if you could please either tell me what to do or email me a working version it would be so much appreciated
Email **pronoun149@gmail.com**

POSTED BY HUZAIFAC 3 YEARS AGO

Hi all...
I would like to know that how to get delivery reports of multipart sms.
we have to pass argument as "sendMultipartTextMessage(newN, null, messages, sentIntents,**deliveryIntentsList**);

so let if my sms contains 3 parts
so the size of delivery report arraylist is 3
so how i will get delivery report?
i will get 1 by 1 or whole as 1?
or what if only part of sms has been delivered than how i can trace that

Regards,
Huzaifa

POSTED BY NIDOSABA 3 YEARS AGO

Hi
Every one!

I need help regarding support of language in SMS program,
i am from Pakistan and Urdu is our language. People's who are using Android want to send and receive message in Urdu..
by default there is no support of Urdu in Android and people cant send and receive message in Urdu language.

I am new in Android development and i need guidance how can i add support of Urdu in SMS Program of Android or how i can add support of Urdu in above application of SMS.
I will be very thank full..

Thanks and Regards

Naveed Ahmad
naveed_2005uet@yahoo.com & **naviduett@gmail.com**

POSTED BY DIXITWADHWANI 2 YEARS AGO

hey great tutorial....

hey i'm facing one simple problem..
when one emulator sending the message to another message through typing in edit box like "hello emu 2", then emulator 2 is receiving the message "hello emu 2"...

but the problem is that when i'm putting the data in edit box from previous activity like "hiii, wass up ???" then emulator 2 is receiving the message in another format/text like "uyioauwop@^"...

please tell me why is happening there..
if you have any solution please suggest me

regards
dixit

Attachment	Size
message send.txt	7.74 KB

POSTED BY DIXITWADHWANI 2 YEARS AGO

> **dixitwadhwani wrote:**
>
> hey great tutorial....
>
> hey i'm facing one simple problem..
> when one emulator sending the message to another message through typing in edit box like "hello emu 2", then emulator 2 is receiving the message "hello emu 2"...
>
> but the problem is that when i'm putting the data in edit box from previous activity like "hiii, wass up ???" then emulator 2 is receiving the message in another format/text like "uyioauwop@^"...
>
> please tell me why is happening there..
> if you have any solution please suggest me
>
> regards
> dixit

got the solution just use the UTF8 unicode during reading the file.....

POSTED BY AKHIL RAVAL 2 YEARS AGO

dear sir,
i want a help that use of class or interface by which i can fetch any of inbuilt application in the mobile like in-box , face book , contact etc.
It means i want that code by which i can fetch all inbox message while my application start run
hopping for quick reply
thank you

"LOVE U ANDROID"

POSTED BY ZOHAIBBROHI 2 YEARS AGO

Thanks Wei-Meng Lee For this piece of code..
I have a question. i want to show the msg in Dialog box . I have shown the msg in dialog box but in only one activity by using INTENT inside SMS Receiver class . i want to show dialog box in current activity. for e.g . if i have receive a msg on ABC activity then the dialog box should appear with msg on ABC activity and if i m on XYZ activity then dialog box should appear on XYZ. Thanks

POSTED BY DANISH894 2 YEARS AGO

i want to make the text encrypted... so that the right reciever can read it.. how and where to add the encryption algorithm??

POSTED BY BARTS49 2 YEARS AGO

dear sir,

Appreciate if you can send me the complete source code so I can test in my eclipse..

Thanks in advance

My email add:
barts49@gmail.com

Regards,
JM

Sign in or **register** to reply.