

Android Workshop Handout

Frank McCown

Working with the emulator

The **Emulator Control** can simulate sending phone calls and SMS messages, changing network transfer rates, and changing the location of the device. To display, select **Window → Show view → Other...**, **Android → Emulator Control**. The control is also visible in the DDMS Perspective: **Window → Open Perspective → Other... → select DDMS option**.

You may also use the command-line interface.

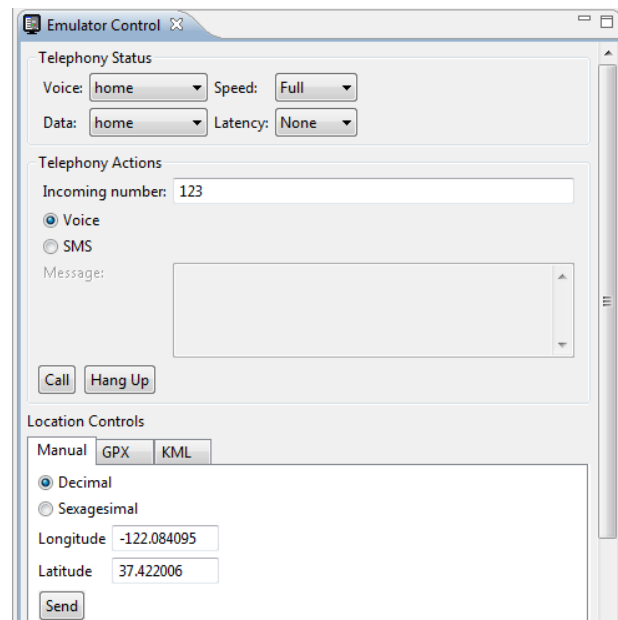
At the command line: `telnet localhost 5554`

Commands:

```
gsm call 123      Place incoming call
gsm cancel 123   Hang up on "123"
```

```
sms send 123 How are you?  Send an SMS
```

```
network speed edge      Simulate Edge transfer rates
network speed full      Full transfer speed
```



More info on emulator command-line options:

<http://developer.android.com/guide/developing/devices/emulator.html>

Install .apk file on Android device using adb

After you have created an Android app, you can deploy it to your Android device.

1. For Windows, download the Android USB drivers from <http://developer.android.com/sdk/win-usb.html>
2. Connect your Android device to your computer using the USB cable.
3. Allow your device to install non-Market apps: **Settings → Applications → check Unknown sources**
4. At the command-line: `adb install path/file.apk`

Install .apk file on Android device using the Web

1. Place the .apk file on the Web.
2. Allow your device to install non-Market apps: **Settings → Applications → check Unknown sources**
3. Enable USB debugging: **Settings → Applications → Development → check USB debugging**
4. Using the Android web browser, access the URL of the .apk file, and your device will prompt you to install the app.

Viewing the LogCat

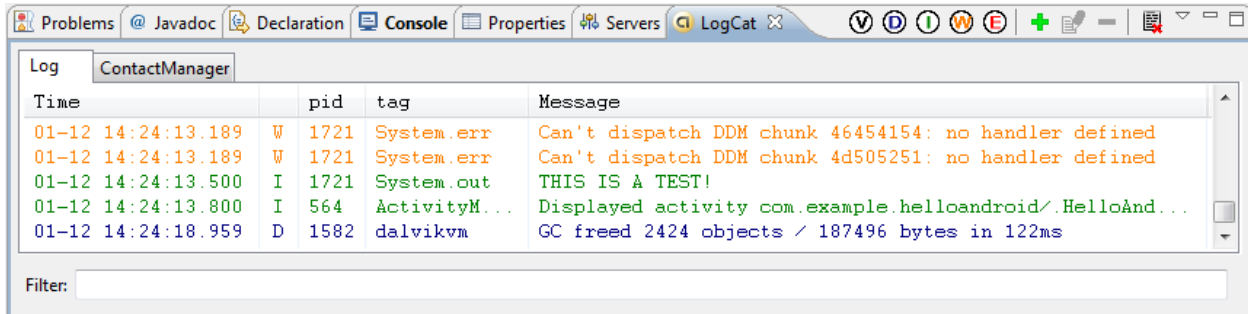
To view output statements like the one below, you need to bring up the LogCat window.

```
System.out.println("THIS IS A TEST!");
```

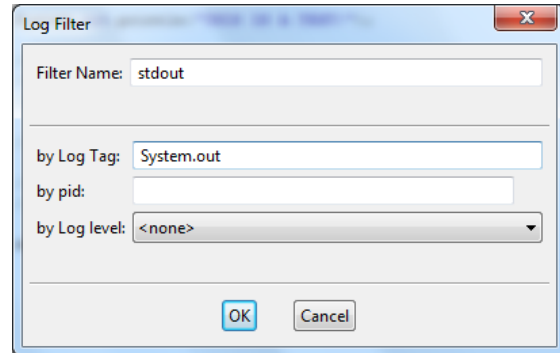


This work is licensed under a [Creative Commons Attribution 3.0 Unported License](http://creativecommons.org/licenses/by/3.0/)

To view the LogCat, select **Window** → **Show view** → **Other...**, **Android** → **LogCat**. Now you'll see the LogCat filled with many different messages, including the one you just output to stdout:



Because there are so many messages, you may want to filter out only those to stdout, so you'll need to create a filter that shows only messages tagged with "System.out". Press the green + which will display the Log Filter dialog box displayed on the right. Enter "stdout" for the filter name and "System.out" for the Log Tag.



Now you will have a tab in LogCat called "stdout" which will only contain messages sent to stdout.

Logging Messages

Android provides a built-in logging facility (`android.util.Log`) which allows for fine-grained logging and filtering. Use the same log tag for each of your messages:

```
private static final String LOG_TAG = "Log_HelloAndroid";
```

To log a message:

```
int x = 2;  
Log.v(LOG_TAG, "x is " + x);
```

There are various types of messages you may want to log:

v = verbose d = debug i = information w = warning e = error

Create a log filter for "Log_HelloAndroid" to see only messages in LogCat that are from this application.



This example application launches a second activity, passing it the value of the `mHelloCount` variable. The second activity sends back a boolean value indicating if the checkbox was clicked or not. The value of `mHelloCount` is also maintained when the app's layout is changed from perspective to landscape.

HelloAndroid.java

```
package com.example.helloandroid;

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

public class HelloAndroid extends Activity {

    private int mHelloCount = 0;

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

        final EditText edittext = (EditText) findViewById(R.id.name);

        final Button button = (Button) findViewById(R.id.hello_button);
        button.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {

                mHelloCount++;
                Toast.makeText(HelloAndroid.this, "Hello, " + edittext.getText() +
                    " (" + mHelloCount + ") ",
                    Toast.LENGTH_SHORT).show();

                // Send the value of mHelloCount to the SecondActivity
                Intent myIntent = new Intent(HelloAndroid.this, SecondActivity.class);
                myIntent.putExtra("mHelloCount", mHelloCount);

                // Get ActivityNotFoundException if activity isn't registered in AndroidManifest.xml
                startActivityForResult(myIntent, 0);
            }
        });
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        if (requestCode == 0) {
            if (resultCode == RESULT_OK) {

                // Get the value returned by the SecondActivity
                boolean ret = (boolean) data.getBooleanExtra("is_checked", false);
                Toast.makeText(HelloAndroid.this, "Value returned: " + ret,
                    Toast.LENGTH_SHORT).show();
            }
        }
    }

    @Override
    protected void onRestoreInstanceState(Bundle savedInstanceState) {
        super.onRestoreInstanceState(savedInstanceState);
        mHelloCount = savedInstanceState.getInt("mHelloCount");
    }

    @Override
    protected void onSaveInstanceState(Bundle outState) {
        super.onSaveInstanceState(outState);
        outState.putInt("mHelloCount", mHelloCount);
    }
}
```



SecondActivity.java

```
package com.example.helloandroid;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Toast;

public class SecondActivity extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.second);

        final CheckBox checkbox = (CheckBox) findViewById(R.id.my_check_box);

        final Button button = (Button) findViewById(R.id.my_button);
        button.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                // Send back to the HelloAndroid activity the checkbox status
                Intent data = new Intent();
                data.putExtra("is_checked", checkbox.isChecked());
                setResult(RESULT_OK, data);
                finish();
            }
        });

        // Get the hello count sent from HelloAndroid
        Intent sourceIntent = getIntent();
        int helloCount = sourceIntent.getIntExtra("mHelloCount", -1);
        Toast.makeText(this, "Hello count: " + helloCount,
            Toast.LENGTH_SHORT).show();
    }
}
```

AndroidManifest.xml

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

        <!-- Added activity -->
        <activity android:name=".SecondActivity"
            android:label="Second Activity" />
    </application>
</manifest>
```



res/layout/main.xml

```
<?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" >

    <EditText
        android:id="@+id/name"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />

    <Button
        android:id="@+id/hello_button"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:text="Press Me" />

</LinearLayout>
```

res/layout/second.xml

```
<?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" >

    <CheckBox android:text="Checked?"
        android:id="@+id/my_check_box"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <Button
        android:id="@+id/my_button"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:text="Send Info Back" />

</LinearLayout>
```

values/strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Android</string>
    <string name="app_name">Hello Android</string>
</resources>
```

