

# Android Notifications

Victor Matos  
Cleveland State University

Notes are based on:

Android Developers

<http://developer.android.com/index.html>





# Notifications

## What is a Notification?

A *notification* is a short message briefly displayed on the *status line*.

It typically announces the happening of an special event for which a trigger has been set.

After opening the *Notification Panel* the user may choose to click on a selection and execute an associated activity.

# Notifications

## What is a Notification?

Notification shown on the status line

Drag down

Click on *Notification Panel* to execute associated application

555-4

5554: Remember to buy bread and milk  
Sent: 8:27PM

5554: Golfing at Manakiki sat. at 1:00pm  
Sent: 8:29PM

Type to compose Send



# Notifications

## Notification Manager

This class notifies the user of events that happen in the background.

Notifications can take different forms:

1. A **persistent icon** that goes in the status bar and is accessible through the launcher, (when the user selects it, a designated Intent can be launched),
2. Turning on or **flashing LEDs** on the device, or
3. Alerting the user by flashing the **backlight**, playing a **sound**, or **vibrating**.



# Notifications

## Notification Manager

You do not instantiate this class directly; instead, retrieve it through **`getSystemService (String )`**.

### Example:

```
String servName = Context.NOTIFICATION_SERVICE;  
  
notificationManager = (NotificationManager) getSystemService (servName);
```



# Notifications

## Notification

This class represents how a persistent notification is to be presented to the user using the NotificationManager.

```
public Notification (int icon, CharSequence tickerText, long when)
```

### Parameters

- |                   |                                                                                  |
|-------------------|----------------------------------------------------------------------------------|
| <i>icon</i>       | The resource id of the icon to put in the status bar.                            |
| <i>tickerText</i> | The text that flows by in the status bar when the notification first activates.  |
| <i>when</i>       | The time to show in the time field.<br>In the System.currentTimeMillis timebase. |



# Notifications

## Notification - Methods

```
public void notify (int id, Notification notification)
```

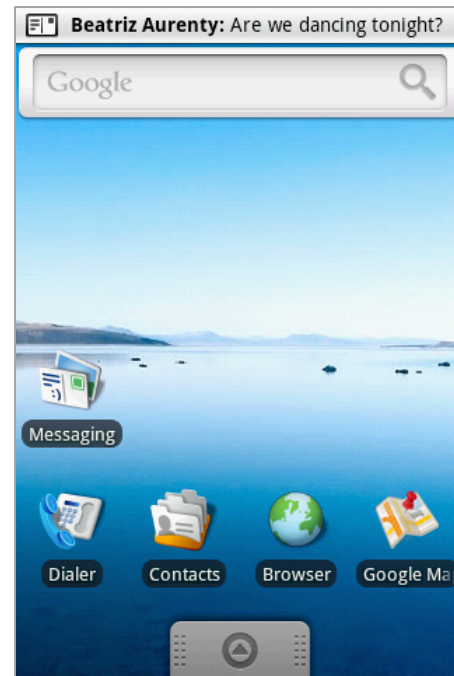
Places a persistent notification on the status bar.



### Parameters

*id* An identifier for this notification unique within your application.

*notification* A Notification object describing how to notify the user, other than the view you're providing. Must not be null.





# Notifications

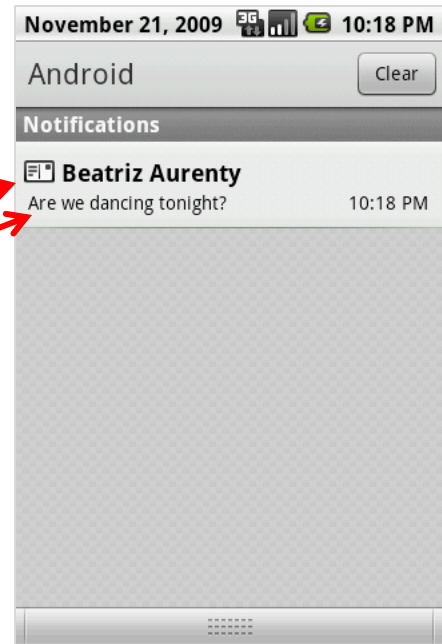
## Notification – Methods

```
public void setLatestEventInfo (
    Context context, CharSequence contentTitle,
    CharSequence contentText, PendingIntent contentIntent)
```

Sets the contentView field to be a view with the standard "Latest Event" layout.

### Parameters

context	The context for your application / activity.
contentTitle	The title that goes in the expanded entry.
contentText	The text that goes in the expanded entry.
contentIntent	The intent to launch when the user clicks the expanded notification.







# Notifications

## Notification – Methods

```
public void cancel ( int id )  
public void cancelAll ( )
```

Cancel a previously shown notification. If it's transient, the view will be hidden. If it's persistent, it will be removed from the status bar.

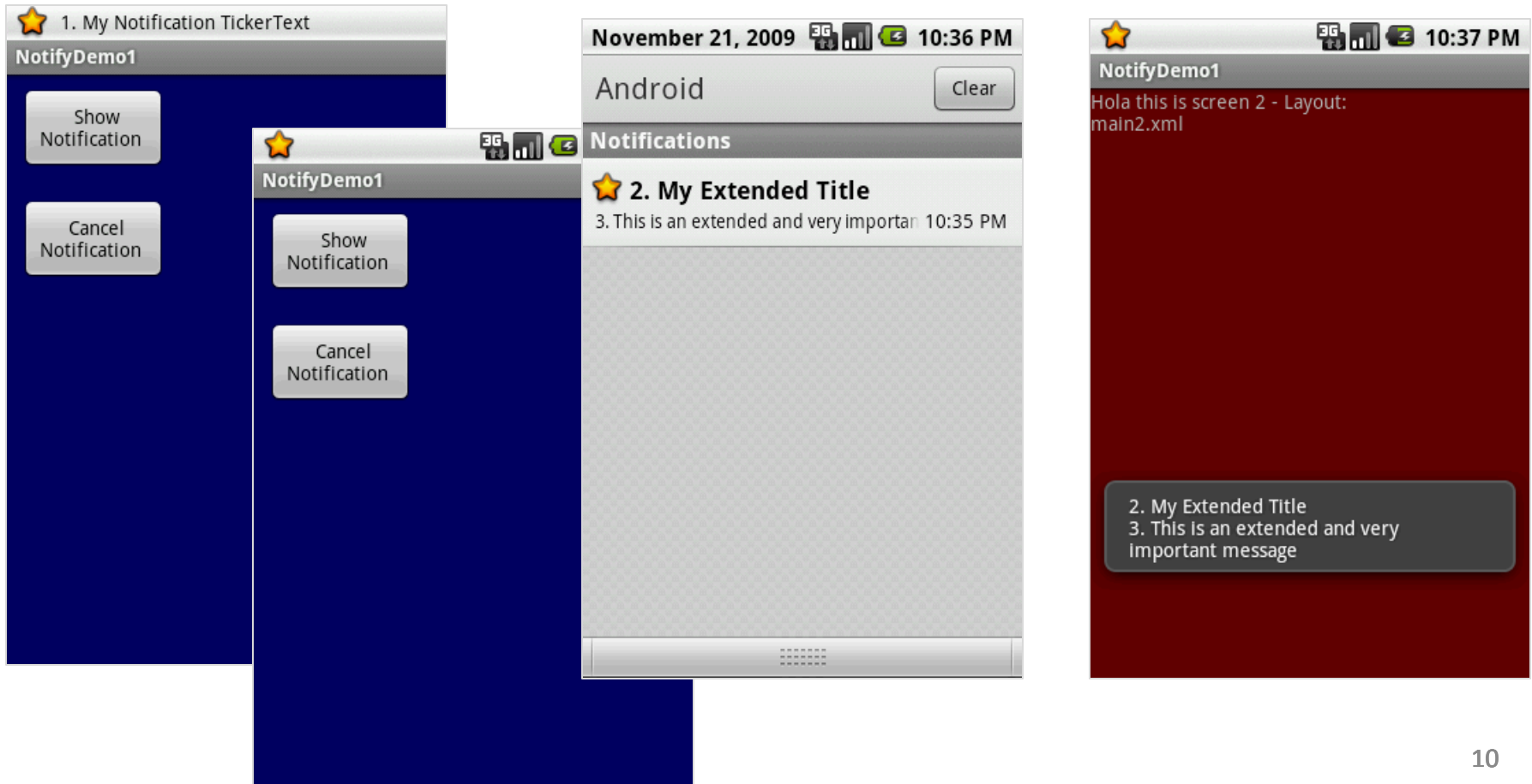
### Parameters

*Id*     An identifier for this notification unique within your application.

# Notifications

## Example.

Produce a notification. Allow the user to click on the Notification Panel and execute appropriate activity to attend the message.





# Notifications

## Example - Layouts

### main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:id="@+id/myLinearLayout1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#ff000066"
    android:orientation="vertical"
    xmlns:android="http://schemas.android.com/apk/res/android" >

    <Button
        android:id="@+id/btnGo"
        android:layout_width="106px"
        android:layout_height="61px"
        android:layout_margin="10px"
        android:text=" Show Notification " >
    </Button>
    <Button
        android:id="@+id/btnStop"
        android:layout_width="106px"
        android:layout_height="61px"
        android:layout_margin="10px"
        android:text=" Cancel Notification " >
    </Button>
</LinearLayout>
```

### main2.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:id="@+id/main2LinLayout"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#ff660000"
    android:orientation="vertical"
    xmlns:android="http://schemas.android.com/apk/res/
    android"
    >
    <TextView
        android:id="@+id/widget29"
        android:layout_width="251px"
        android:layout_height="69px"
        android:text="Hola this is screen 2 - Layout:main2.xml"
    >
    </TextView>
</LinearLayout>
```



# Notifications

## Example – Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="cis493.demos"
    android:versionCode="1"
    android:versionName="1.0">

    <application android:icon="@drawable/icon"
        android:label="@string/app_name">

        <activity android:name=".NotifyDemo1"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity android:name=".NotifyHelper" >
        </activity>

    </application>

    <uses-sdk android:minSdkVersion="4" />

</manifest>
```

## /drawable



btn\_star\_big\_on\_selected.png

### Note:

Obtain the icon from the folder

C:\Android\platforms\android-1.x\data\res\drawable



# Notifications

## Example – Create & Cancel a Notification

```
package cis493.demos;

import android.app.Activity;
import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.*;

////////////////////////////////////

public class NotifyDemo1 extends Activity {
    Button btnGo;
    Button btnStop;
    int notificationId = 1;
    NotificationManager notificationManager;

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



# Notifications

## Example – Create & Cancel a Notification

```

btnGo = (Button)findViewById(R.id.btnGo);
btnGo.setOnClickListener(new OnClickListener() {

    public void onClick(View v) {
        //define a notification manager
        String serName = Context.NOTIFICATION_SERVICE;
        notificationManager = (NotificationManager)getSystemService(serName);

        //define notification using: icon, text, and timing.
        int icon = R.drawable.btn_star_big_on_selected;
        String tickerText = "1. My Notification TickerText";
        long when = System.currentTimeMillis();
        Notification notification = new Notification(icon, tickerText, when);

        //configure appearance of the notification
        String extendedTitle = "2. My Extended Title";
        String extendedText = "3. This is an extended and very important message";

        // set a Pending Activity to take care of the potential request the user
        // may have by clicking on the notification asking for more explanations
        Intent intent = new Intent(getApplicationContext(), NotifyHelper.class);
        intent.putExtra("extendedText", extendedText);
        intent.putExtra("extendedTitle", extendedTitle);
        PendingIntent launchIntent =
            PendingIntent.getActivity(getApplicationContext(), 0, intent, 0);
    }
}

```



# Notifications

## Example – Create & Cancel a Notification

```

notification.setLatestEventInfo(getApplicationContext(),
                                extendedTitle, extendedText, launchIntent);

//trigger notification
notificationId = 1;
notificationManager.notify(notificationId, notification);

} //click
});

////////////////////////////////////
btnStop = (Button) findViewById(R.id.btnStop);
btnStop.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        //canceling a notification
        notificationId = 1;
        notificationManager.cancel(notificationId);
    }
});

} //onCreate

} //NotifyDemo1

```



# Notifications

## Example - SubActivity – Attending the Notification

```
package cis493.demos;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.Toast;

public class NotifyHelper extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main2);

        Intent myData = getIntent();
        // extract the extra-data in the Notification
        String msg = myData.getStringExtra("extendedTitle") + "\n"
            + myData.getStringExtra("extendedText");

        Toast.makeText(getApplicationContext(), msg, 1).show();
    }
}
```





# Notifications

**Questions**

