CS 696 Emerging Web and Mobile Technologies
Spring Semester, 2011
Doc 26 Android App Widgets
Apr 28, 2011

References

App Widgets, http://developer.android.com/guide/topics/appwidgets/index.html

Introducing home screen widgets and the AppWidget framework, http://android-developers.blogspot.com/2009/04/introducing-home-screen-widgets-and.html

Widget Design Guidelines http://developer.android.com/guide/practices/ui_guidelines/widget_design.html

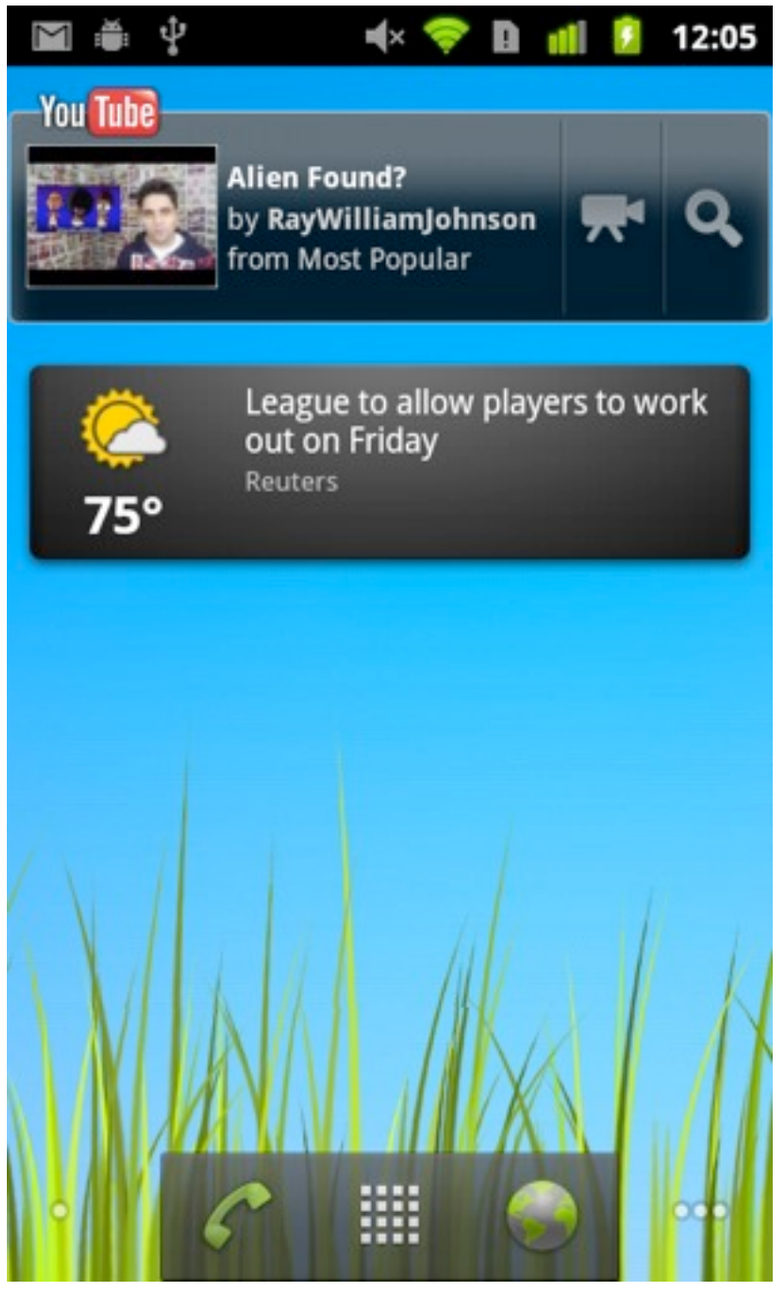
App Widget

Small application that can be embedded in Home screen

Receive periodic updates

Sample App Widgets

Current Weather
Quote of the Day
Stock Prices



Basic Parts

AppWidgetProviderInfo

- Metadata for an App Widget in XML

AppWidgetProvider class implementation

- Java code for the App Widget

View layout

- Initial layout for the App Widget in XML

Configuration Activity

- Optional
- Configures App Widget when created

Manifest Info

```
<receiver android:name="ExampleAppWidgetProvider" >
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
    </intent-filter>
    <meta-data android:name="android.appwidget.provider"
               android:resource="@xml/example_appwidget_info" />
</receiver>
```

AppWidgetProviderInfo Metadata

```
<appwidget-provider xmlns:android="http://schemas.android.com/apk/res/android"
    android:minWidth="294dp" <!-- density-independent pixels -->
    android:minHeight="72dp"
    android:updatePeriodMillis="86400000" <!-- once per day -->
    android:initialLayout="@layout/example_appwidget"
    android:configure="com.example.android.ExampleAppWidgetConfigure" >
</appwidget-provider>
```

Metadata - minWidth & minHeight

Home Screen - grid of cells

Cell is 74 pixels by 74 pixels

Pixel rounding error - 2 pixels

App Widget size

(number of cells * 74) - 2

updatePeriodMillis

how often the App Widget framework calls update on your App Widget

Updates of every hour or longer should not drain the battery

App Widget Layout

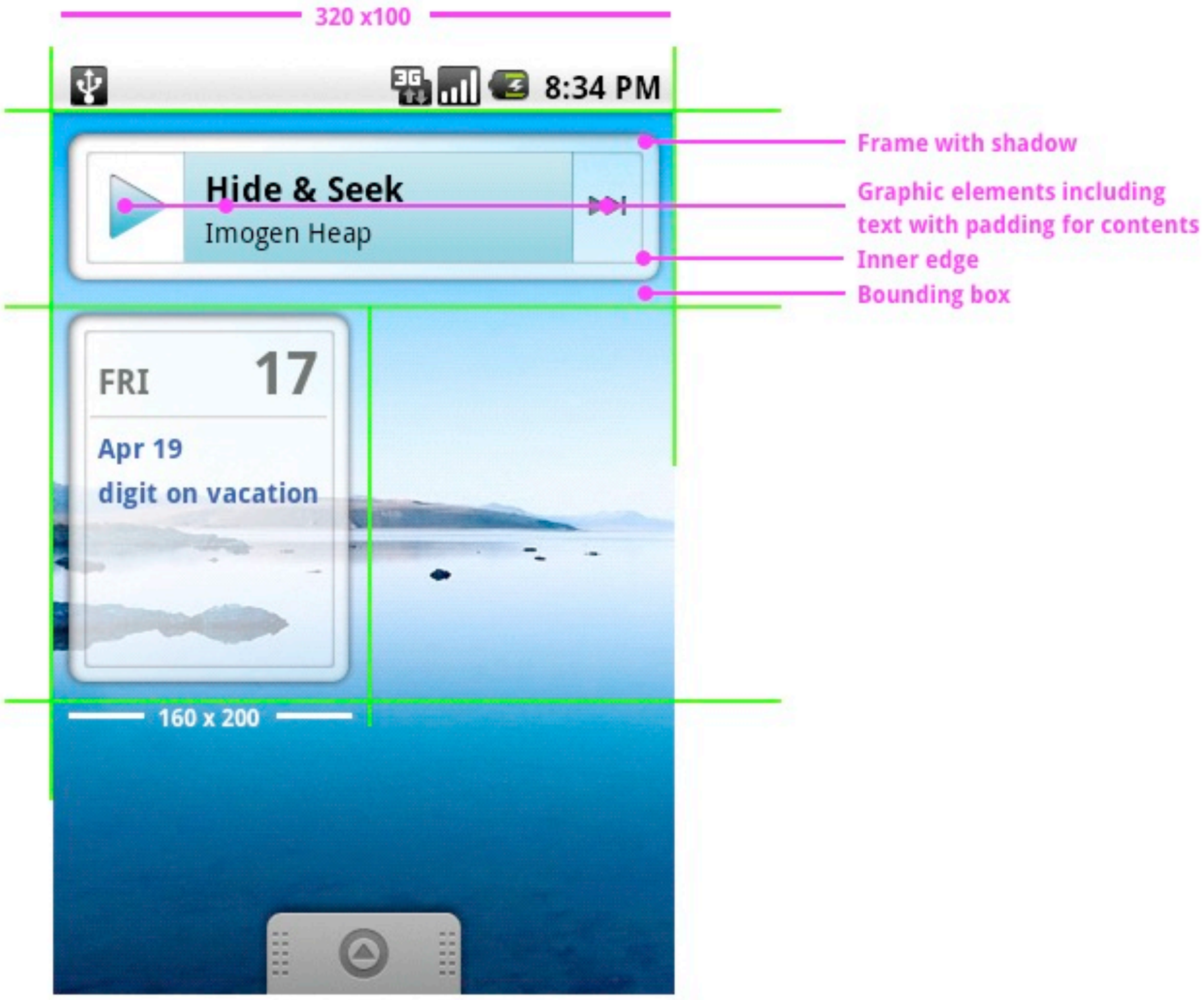
Layouts

FrameLayout
LinearLayout
RelativeLayout

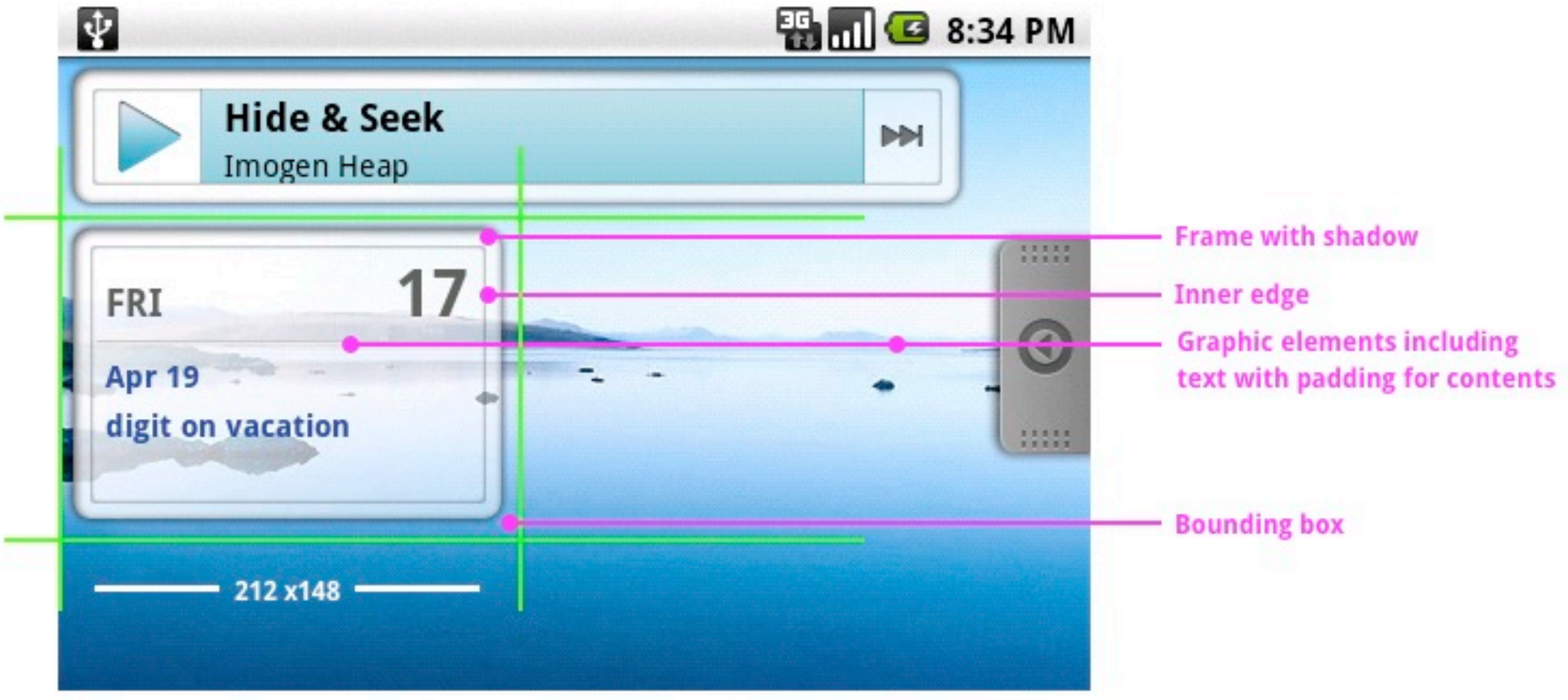
Views

AnalogClock
Button
Chronometer
ImageButton
ImageView
ProgressBar
TextView

Anatomy of Widget



Landscape


8:34 PM
Hide & Seek
Imogen Heap
FRI
17
Apr 19
digit on vacation
212 x148
Frame with shadow
Inner edge
Graphic elements including
text with padding for contents
Bounding box

App Widget Graphics

You will need graphics for your App Widget

http://developer.android.com/guide/practices/ui_guidelines/widget_design.html

AppWidgetProvider

onUpdate(Context, AppWidgetManager, int[])

The main method

Called by App Widget manager

onDeleted(Context, int[])

Called every time an App Widget is deleted

onEnabled(Context)

Called when the App Widget is created for the first time.

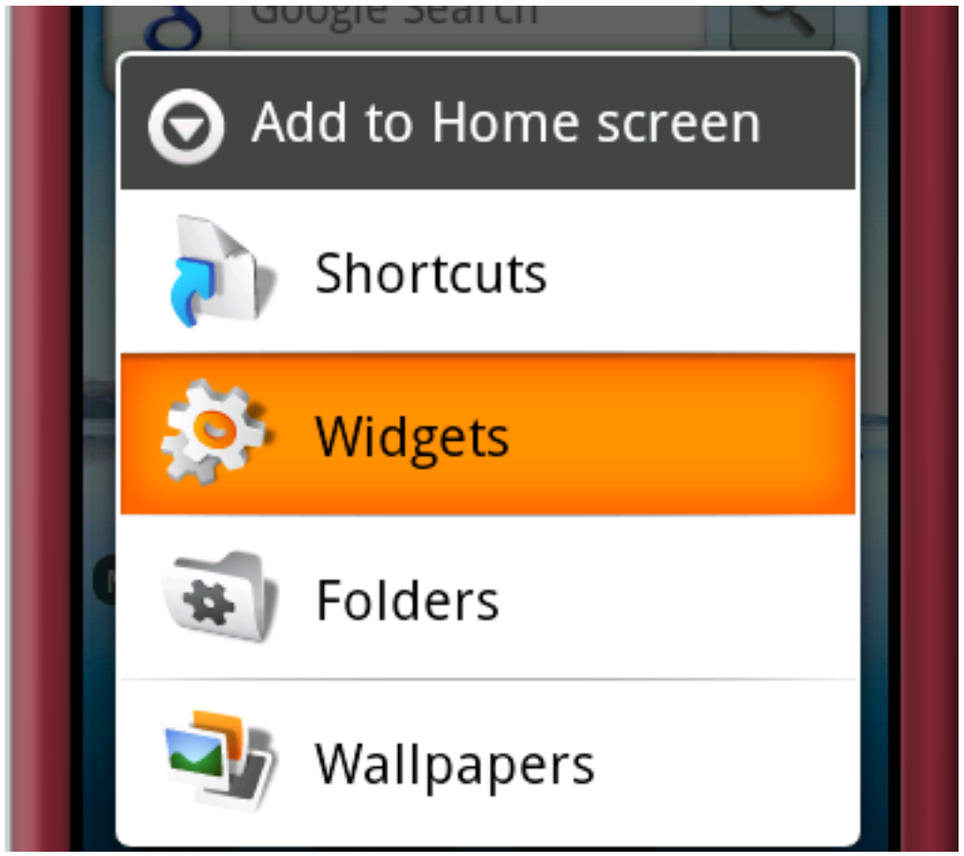
onDisabled(Context)

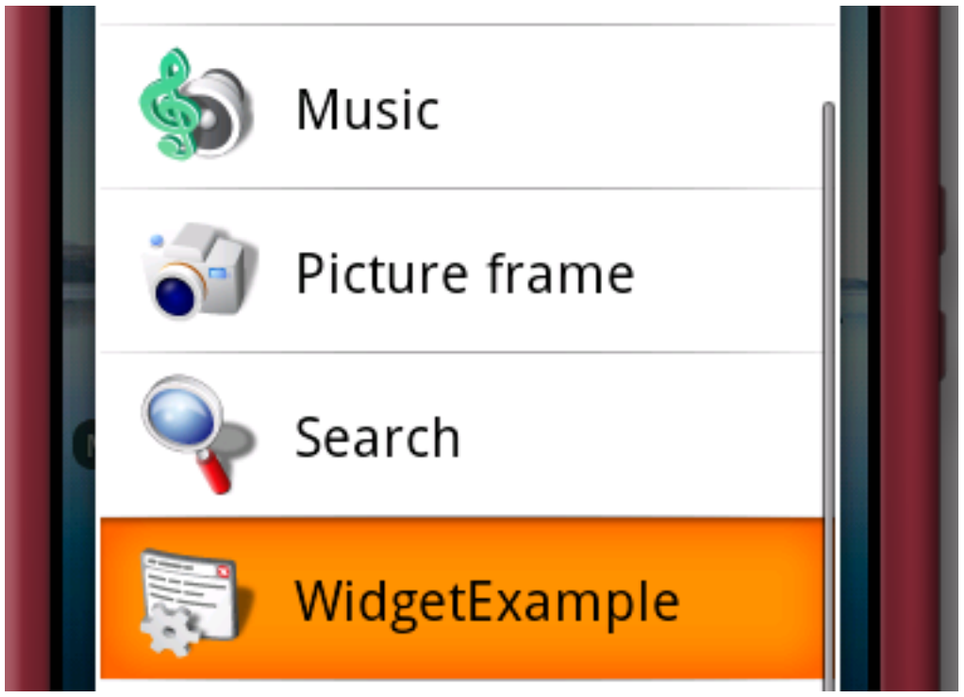
Called when last App Widget is deleted

onReceive(Context, Intent)

All calls go here first

First Example - Just Text





SampleWidget.java

```
public class SampleWidget extends AppWidgetProvider {

    public void onUpdate(Context context, AppWidgetManager appWidgetManager,
            int[] appWidgetIds) {
            RemoteViews views = new RemoteViews(context.getPackageName(),
                R.layout.widget);
            appWidgetManager.updateAppWidget(appWidgetIds[0], views);
    }
}
```

res/layout/widget.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/widget"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <TextView
        android:id="@+id/message"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="12dip"
        android:padding="10dip"
        android:gravity="center"
        android:text="Sample"
    />
</LinearLayout>
```

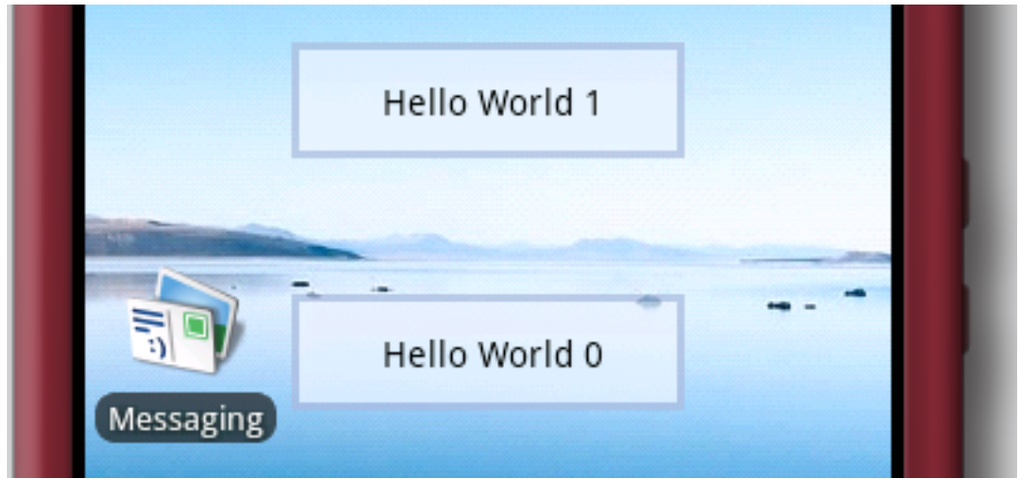
res/xml/appwidget_definition.xml

```
<?xml version="1.0" encoding="utf-8"?>
<appwidget-provider
  xmlns:android="http://schemas.android.com/apk/res/android"
   android:minWidth="146dip"
  android:minHeight="72dip"
  android:updatePeriodMillis="86400000"
  android:initialLayout="@layout/widget"
  >
</appwidget-provider>
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="edu.sdsu.cs.whitney"
      android:versionCode="1"
      android:versionName="1.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
            <receiver android:name=".SampleWidget"
           android:label="@string/app_name" >
           <intent-filter>
               <action
                  android:name="android.appwidget.action.APPWIDGET_UPDATE" />
           </intent-filter>
           <meta-data
              android:name="android.appwidget.provider"
              android:resource="@xml/appwidget_definition" />
        </receiver>
    </application>
    <uses-sdk android:minSdkVersion="3" />
</manifest>
```

With Background and Multiple Widgets



SampleWidget

```
public class SampleWidget extends AppWidgetProvider {

    public void onUpdate(Context context, AppWidgetManager appWidgetManager,
            int[] appWidgetIds) {
        final int N = appWidgetIds.length;
        Log.i("test", "start");
        for (int id = 0; id < N; id++) {
            Log.i("test", "id " + id);
            RemoteViews views = new RemoteViews(context.getPackageName(),
                    R.layout.widget);
            views.setTextViewText(R.id.message, "Hello World " + id);
            appWidgetManager.updateAppWidget(appWidgetIds[id], views);
        }
    }
}
```

Layout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/widget"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
     style="@style/WidgetBackground">
    <TextView
        android:id="@+id/message"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="12dip"
        android:padding="10dip"
        android:gravity="center"
        android:text="Sample"
        style="@style/Text"
    />
</LinearLayout>
```

res/values/styles.xml

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

    <style name="WidgetBackground">
        <item name="android:background">@drawable/widget_bg</item>
    </style>

    <style name="Text">
      <item name="android:textSize">14sp</item>
      <item name="android:textColor">@android:color/black</item>
    </style>
</resources>
```

Starting an Activity

Widget Button Starts an Activity

```
public class SampleWidget extends AppWidgetProvider {

    public void onUpdate(Context context, AppWidgetManager appWidgetManager,
            int[] appWidgetIds) {
        final int N = appWidgetIds.length;
        for (int id = 0; id < N; id++) {
            RemoteViews views = new RemoteViews(context.getPackageName(),
                    R.layout.widget);
            views.setTextViewText(R.id.message, "Hello " + id);
            Intent intent = new Intent(context, SampleActivity.class);
            PendingIntent pendingIntent = PendingIntent.getActivity(context, 0,
                    intent, 0);
            views.setOnClickPendingIntent(R.id.go, pendingIntent);
            appWidgetManager.updateAppWidget(appWidgetIds[id], views);
        }
    }
}
```

Sending Information to the App Widget

Listening to Broadcasts

AppWidgetProvider is a Broadcast Listener

So we can send broadcast to it

onReceive() will receive the broadcast.

Make sure to register the Action

```
 <receiver android:name=".SampleWidget"
android:label="@string/app_name" >
<intent-filter>
    <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
    <action android:name="edu.sdsu.cs.whitney.ACTION_UPDATE_WIDGET" />
</intent-filter>
<meta-data
    android:name="android.appwidget.provider"
    android:resource="@xml/appwidget_definition" />
</receiver>
```

Sending the Broadcast

```
public class SampleActivity extends Activity implements View.OnClickListener {
        public static final String ACTION_UPDATE_WIDGET = "edu.sdsu.cs.whitney.ACTION_UPDATE_WIDGET";
        public static final String MESSAGE = "message";

        public void onClick(View v) {
                sendBroadcast();
        }

        public void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.main);
                Button start = (Button) findViewById(R.id.start);
                start.setOnClickListener(this);
        }

        private void sendBroadcast() {
                Log.i("test", "send broadcast");
                Intent broadcast = new Intent(this, AppWidgetProvider.class);
                broadcast.putExtra("MESSAGE", "Bye");
                broadcast.setAction(ACTION_UPDATE_WIDGET);
                sendBroadcast(broadcast);
        }
}
```

Getting the Broadcasts

```
public class SampleWidget extends AppWidgetProvider {

    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();
        if (action != null
                && action.equals(SampleActivity.ACTION_UPDATE_WIDGET)) {
            String message;
            if (intent.hasExtra(SampleActivity.MESSAGE))
                message = (String) intent
                        .getCharSequenceExtra(SampleActivity.MESSAGE);
            else
                message = "Not Found";
            AppWidgetManager appWidgetManager = AppWidgetManager
                    .getInstance(context);
            int[] appWidgetIds = appWidgetManager
                    .getAppWidgetIds(new ComponentName(context,
                            SampleWidget.class));
            for (int index = 0; index < appWidgetIds.length; index++)
                updateSampleWidget(context, appWidgetManager,
                        appWidgetIds[index], message);
        } else
            super.onReceive(context, intent);
    }
```

Getting the Broadcasts

```
public void onUpdate(Context context, AppWidgetManager appWidgetManager,
        int[] appWidgetIds) {
    Log.i("test", "on update");
    final int N = appWidgetIds.length;
    for (int index = 0; index < N; index++)
        updateSampleWidget(context, appWidgetManager, appWidgetIds[index],
            "Hello " + index);
}

private void updateSampleWidget(Context context,
        AppWidgetManager appWidgetManager, int widget, String text) {
    RemoteViews views = new RemoteViews(context.getPackageName(),
        R.layout.widget);
    views.setTextViewText(R.id.message, text);
    Intent intent = new Intent(context, SampleActivity.class);
    PendingIntent pendingIntent = PendingIntent.getActivity(context, 0,
        intent, 0);
    views.setOnClickPendingIntent(R.id.go, pendingIntent);
    appWidgetManager.updateAppWidget(widget, views);
  }
}
```

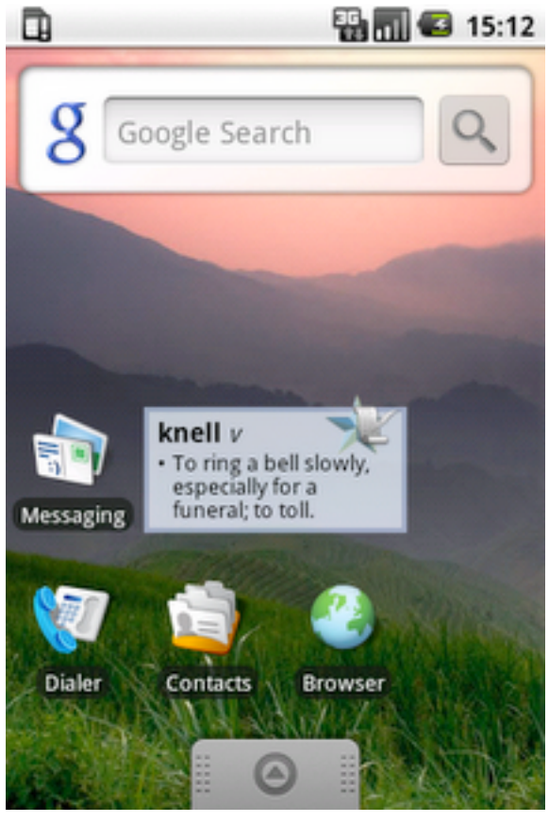
More Normal Use of App Widgets

Normal Use Case

App Widget updates itself with information from the network

Word of the day

http://android-developers.blogspot.com/2009/04/introducing-home-screen-widgets-and.html



Use Service to get Data from Network

```
public class WordWidget extends AppWidgetProvider {
    public void onUpdate(Context context, AppWidgetManager appWidgetManager,
            int[] appWidgetIds) {
        context.startService(new Intent(context, UpdateService.class));
    }

    public static class UpdateService extends Service {
        public void onStart(Intent intent, int startId) {
            RemoteViews updateViews = buildUpdate(this);

            ComponentName thisWidget = new ComponentName(this, WordWidget.class);
            AppWidgetManager manager = AppWidgetManager.getInstance(this);
            manager.updateAppWidget(thisWidget, updateViews);
        }

    public RemoteViews buildUpdate(Context context) {
       Lots of code to get data from the network and put it into RemoteViews
```


Example from http://android-developers.blogspot.com/2009/04/introducing-home-screen-widgets-and.html