**[Android: Why, When and How to use AsyncTask with example](http://www.compiletimeerror.com/2013/01/why-and-how-to-use-asynctask.html)**

**What is AsyncTask?**

               AsyncTask is an abstract class provided by Android which helps us to use the UI thread properly. This class allows us to perform long/background operations and show its result on the UI thread without having to manipulate threads.

**When to use AsyncTask?**  
            Android implements single thread model and whenever an android application is launched, a thread is created. Assuming we are doing network operation on a button click in our application. On button click a request would be made to the server and response will be awaited. Due to single thread model of android, till the time response is awaited our screen is non-responsive. So we should avoid performing long running operations on the UI thread. This includes file and network access.

            To overcome this we can create new thread and implement run method to perform this network call, so UI remains responsive.

            But since Android follows single thread model and Android UI toolkit is not thread safe, so if there is a need to make some change to the UI based on the result of the operation performed, then this approach may lead some issues.

So the Android framework has given a very good pattern which is enveloped into AsyncTask.

***Note:****AsyncTask should ideally be used for operations that take few seconds. Some tasks keep the thread running for long time so in that case it is recommended to use java.util.concurrent package such as Executor, ThreadPoolExecutor and FutureTask.*

**AsyncTask has four steps:**

1. **doInBackground:**Code performing long running operation goes in this method.  When onClick method is executed on click of button, it calls execute method which accepts parameters and automatically calls doInBackground method with the parameters passed.
2. **onPostExecute:**This method is called after doInBackground method completes processing. Result from doInBackground is passed to this method.
3. **onPreExecute:**This method is called before doInBackground method is called.
4. **onProgressUpdate:** This method is invoked by calling publishProgress anytime from doInBackground call this method.

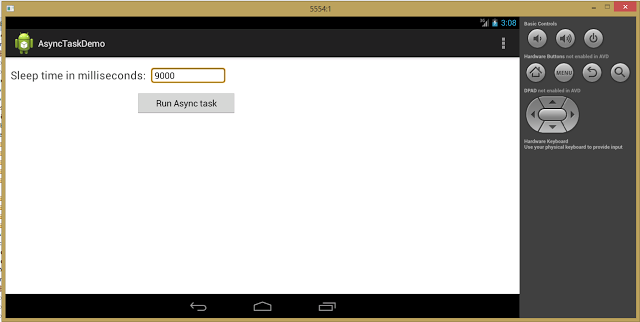
The task can be cancelled by invoking cancel(boolean) method. This will cause subsequent calls to **isCancelled()** to return true. After invoking this method, **onCancelled(Object)** method is called instead of onPostExecute() after doInBackground() returns.

**How to use AyncTask? Example application**

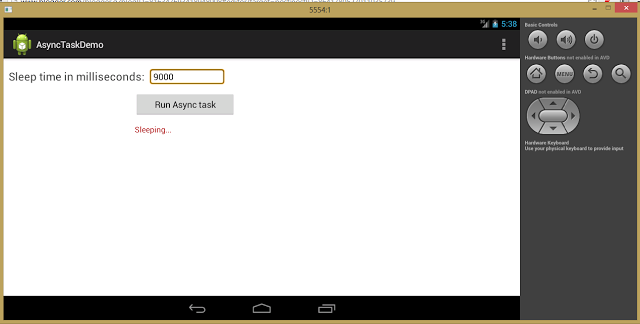
         In this sample application I just made the process to sleep for some period of time instead of doing network operation.( Just to explain the concept of AsyncTask. This is not a realtime application).

In the UI thread user enters a value (time in milli seconds) which will be passed to AsyncTaskRunner.

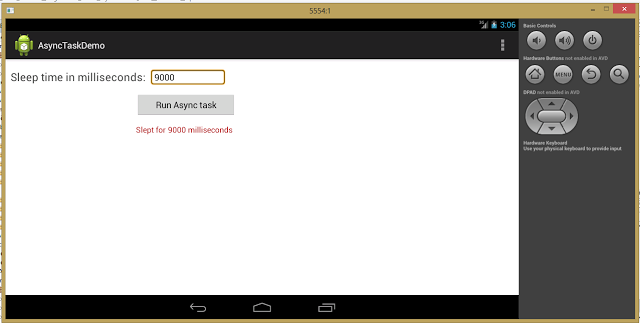
*AsyncTaskRunner is a private class which extends AsyncTask.*



When the user clicks on the **Run Async Task**button, we make the process to sleep for give period of time. At the same time we keep the UI thread responsive by showing the status to the user.



After completing the operation, show the final result to the user.



**Steps to create the application:**

1. Create new android application project in eclipse with default settings.

2. Create the layout for the application.  
3. Create the MainActivity and AsyncTaskRunner at com.example.asynctaskdemo.MainActivity.java  
4. Run the application as given in the above screenshots.

**activity\_main.xml**

<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"

tools:context=".MainActivity" >

<TextView

android:id="@+id/tv\_time"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="10pt"

android:textColor="#444444"

android:layout\_alignParentLeft="true"

android:layout\_marginRight="9dip"

android:layout\_marginTop="20dip"

android:layout\_marginLeft="10dip"

android:text="Sleep time in milliseconds:"/>

<EditText

android:id="@+id/et\_time"

android:layout\_width="150dip"

android:layout\_height="wrap\_content"

android:background="@android:drawable/editbox\_background"

android:layout\_toRightOf="@id/tv\_time"

android:layout\_alignTop="@id/tv\_time"

android:inputType="text"

/>

<Button

android:id="@+id/btn\_do\_it"

android:layout\_width="200dip"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:layout\_marginTop="15dip"

android:layout\_marginLeft="260dip"

android:layout\_below="@id/et\_time"

android:text="Run Async task" />

<TextView

android:id="@+id/tv\_result"

android:layout\_width="400dip"

android:layout\_height="100dip"

android:textSize="7pt"

android:layout\_alignParentLeft="true"

android:layout\_below="@id/btn\_do\_it"

android:layout\_marginRight="9dip"

android:layout\_marginTop="15dip"

android:layout\_marginLeft="260dip"

android:textColor="#AA0000"

android:text=""/>

</RelativeLayout>

**MainActivity.java**

**package** com.example.asynctaskdemo;

**import** **android.app.Activity**;

**import** **android.os.AsyncTask**;

**import** **android.os.Bundle**;

**import** **android.view.Menu**;

**import** **android.view.View**;

**import** **android.widget.Button**;

**import** **android.widget.EditText**;

**import** **android.widget.TextView**;

/\*\*

\* @author Prabu

\* AsyncTask exmple

\*

\*/

**public** **class** **MainActivity** **extends** Activity {

**private** Button button;

**private** EditText time;

**private** TextView finalResult;

**@Override**

**protected** **void** **onCreate**(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

time = (EditText) findViewById(R.id.et\_time);

button = (Button) findViewById(R.id.btn\_do\_it);

finalResult = (TextView) findViewById(R.id.tv\_result);

button.setOnClickListener(**new** View.OnClickListener() {

**@Override**

**public** **void** **onClick**(View v) {

AsyncTaskRunner runner = **new** AsyncTaskRunner();

String sleepTime = time.getText().toString();

runner.execute(sleepTime);

}

});

}

**@Override**

**public** **boolean** **onCreateOptionsMenu**(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.activity\_main, menu);

**return** **true**;

}

/\*\*

\* @author Prabu

\* Private class which runs the long operation. ( Sleeping for some time )

\*/

**private** **class** **AsyncTaskRunner** **extends** AsyncTask<String, String, String> {

**private** String resp;

**@Override**

**protected** String **doInBackground**(String... params) {

publishProgress("Sleeping..."); // Calls onProgressUpdate()

**try** {

// Do your long operations here and return the result

**int** time = Integer.parseInt(params[**0**]);

// Sleeping for given time period

Thread.sleep(time);

resp = "Slept for " + time + " milliseconds";

} **catch** (InterruptedException e) {

e.printStackTrace();

resp = e.getMessage();

} **catch** (Exception e) {

e.printStackTrace();

resp = e.getMessage();

}

**return** resp;

}

/\*

\* (non-Javadoc)

\*

\* @see android.os.AsyncTask#onPostExecute(java.lang.Object)

\*/

**@Override**

**protected** **void** **onPostExecute**(String result) {

// execution of result of Long time consuming operation

finalResult.setText(result);

}

/\*

\* (non-Javadoc)

\*

\* @see android.os.AsyncTask#onPreExecute()

\*/

**@Override**

**protected** **void** **onPreExecute**() {

// Things to be done before execution of long running operation. For

// example showing ProgessDialog

}

/\*

\* (non-Javadoc)

\*

\* @see android.os.AsyncTask#onProgressUpdate(Progress[])

\*/

**@Override**

**protected** **void** **onProgressUpdate**(String... text) {

finalResult.setText(text[**0**]);

// Things to be done while execution of long running operation is in

// progress. For example updating ProgessDialog

}

}

}

*Do you like to add your own tutorials here? Here is an opportunity... Check this*[*link*](http://www.compiletimeerror.com/p/blog-page_8.html)*! !*

[http://4.bp.blogspot.com/-2NZ-kw6U0MY/Uoz0ZPoPehI/AAAAAAAAAjE/IxQx9jZ-vqo/s1600/Previous.png](http://www.compiletimeerror.com/2013/03/android-broadcast-receiver-in-detail.html)

[http://1.bp.blogspot.com/--AKxKx-_hQA/Uoz0Yy9DGhI/AAAAAAAAAjA/p8_22u7vRxk/s1600/Next.png](http://www.compiletimeerror.com/2013/06/check-internet-connection-in-android.html)