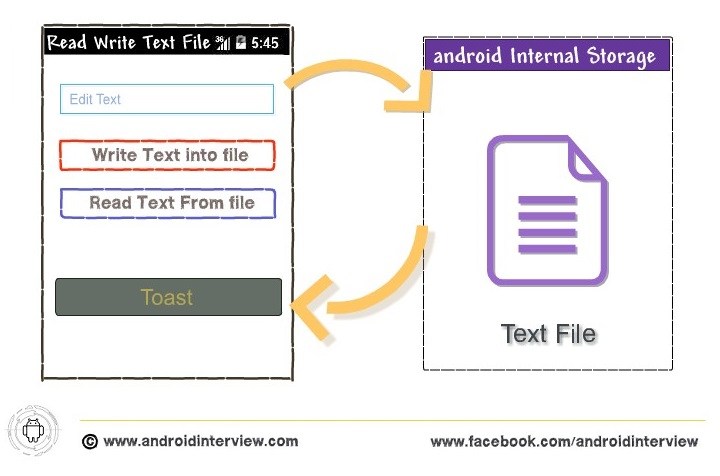
**Android Internal Storage: Read and Write text file**

## **Android Internal Storage :**

In Android you can also save files (text file etc) directly on the device’s internal storage. By default,Saving and Loading Files to the internal storage are private to your application and different applications will not access them (nor can the user). once the user uninstalls this application, these files will automatically remove.

[](https://i0.wp.com/www.androidinterview.com/wp-content/uploads/2014/09/android-read-and-write-text-file-example1.jpg)

In Android data storage, It’s good practice to use [Shared Preferences](http://www.androidinterview.com/android-data-storage-options-sharedpreferences-tutorial/) or a database to store your application data, but there are still times once you’ll need to use files directly rather than rely on Android’s managed mechanisms.

Sometime want to save data which have not be best represented as key/value pairs, then you may need to use the primitive method of saving it on to the filesystem as files. Saving data in text file is helpful for saving long strings of text, or binary data etc in your application.

### **Solution Steps Write and Read Text File :**

As well as the standard Java I/O classes and methods, android also offers **openFileInput**and**openFileOuput** to modify reading and writing streams from and to local files. In  android to create a file and saving, you can use the **openFileOutput()**method and **openFileInput()** method to open a file for reading:

Creating file

Java



|  |  |
| --- | --- |
| 1  2  3  4  5 | String MY\_FILE\_NAME = “mytextfile.txt”;  // Create a new output file stream  FileOutputStream fileos = openFileOutput(MY\_FILE\_NAME, Context.MODE\_PRIVATE);  // Create a new file input stream.  FileInputStream fileis = openFileInput(My\_FILE\_NAME); |

1. Add Following xml code into **activity\_main.xml**

File : res/layout/activity\_main.xml

activity\_main.xml

XHTML



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51 | <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="{relativePackage}.${activityClass}" >        <TextView          android:id="@+id/textView1"          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content"          android:layout\_alignParentLeft="true"          android:layout\_alignParentRight="true"          android:background="#008080"          android:padding="5dp"          android:text="Android Read and Write Text form File"          android:textColor="#fff" />        <EditText          android:id="@+id/editText1"          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content"          android:layout\_alignParentLeft="true"          android:layout\_alignParentRight="true"          android:layout\_below="@+id/textView1"          android:layout\_marginTop="22dp"          android:ems="10"          android:layout\_margin="5dp">          <requestFocus />      </EditText>        <Button          android:id="@+id/button1"          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content"          android:layout\_alignParentLeft="true"          android:layout\_alignParentRight="true"          android:layout\_below="@+id/editText1"          android:text="Write Text into File"          android:onClick="WriteBtn"          android:layout\_margin="5dp"/>      <Button          android:id="@+id/button2"          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content"          android:layout\_alignParentLeft="true"          android:layout\_alignParentRight="true"          android:layout\_below="@+id/button1"          android:text="Read Text From file"          android:onClick="ReadBtn"          android:layout\_margin="5dp" />  </RelativeLayout> |

2. Open “**MainActivity.java**” file and add following JAVA code.

File : src/package-name/MainActivity.java

For writing text to a file, use the **OutputStreamWriter** class and use its **write()** method to write a string to the file. To save the changes to the file, use its **close()** method:

MainActivity.java

Java



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67 | package androidinterview.com.androidreadwritetextfile;    import java.io.FileInputStream;  import java.io.FileOutputStream;  import java.io.InputStreamReader;  import java.io.OutputStreamWriter;    import android.app.Activity;  import android.os.Bundle;  import android.view.View;  import android.widget.EditText;  import android.widget.Toast;    public class MainActivity extends Activity {    EditText textmsg;  static final int READ\_BLOCK\_SIZE = 100;  *@Override*  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);    textmsg=(EditText)findViewById(R.id.editText1);  }    // write text to file  public void WriteBtn(View v) {  // add-write text into file  try {  FileOutputStream fileout=openFileOutput("mytextfile.txt", MODE\_PRIVATE);  OutputStreamWriter outputWriter=new OutputStreamWriter(fileout);  outputWriter.write(textmsg.getText().toString());  outputWriter.close();    //display file saved message  Toast.makeText(getBaseContext(), "File saved successfully!",  Toast.LENGTH\_SHORT).show();    } catch (Exception e) {  e.printStackTrace();  }  }    // Read text from file  public void ReadBtn(View v) {  //reading text from file  try {  FileInputStream fileIn=openFileInput("mytextfile.txt");  InputStreamReader InputRead= new InputStreamReader(fileIn);    char[] inputBuffer= new char[READ\_BLOCK\_SIZE];  String s="";  int charRead;    while ((charRead=InputRead.read(inputBuffer))>0) {  // char to string conversion  String readstring=String.copyValueOf(inputBuffer,0,charRead);  s +=readstring;  }  InputRead.close();  Toast.makeText(getBaseContext(), s,Toast.LENGTH\_SHORT).show();    } catch (Exception e) {  e.printStackTrace();  }  }  } |

### **Output of  Android Read Write Text File in emulator :**

[](https://i0.wp.com/www.androidinterview.com/wp-content/uploads/2014/09/android-read-and-write-text-file-example2.jpg)