About This Article

Android provides the following four mechanisms for storing and retrieving the data:

1. Preferences
2. Databases
3. Files
4. Network

This article explains how to store persistent application data with shared preferences. Preferences are an Android lightweight mechanism stored and retrieved as groups of key/value pairs. Typically used to keep state information and shared data among several activities of an application. On each entry key/value the key is a string and the value must be primitive data type.

Scope:

This article is intended for novice users who want brief understanding on Android programming. This article guides on using preferences in developing Android applications. It assumes that the user has already installed Android and necessary tools to develop application. It also assumes that the user to be familiar with basic knowledge of Android and Java.

Introduction

While developing applications a common need that arises is the one of storing user's preferences and using them when the application starts. Android provides a mechanism to store user’s data through using preferences. Shared preferences are simply sets of data values that are share persistently.

Android preferences is a key/value entries that store data that can be specific to a certain activity or shared among all activities within the application. There are also built-in preferences related widgets that are defined in xml file.

Using Preferences

In Android there are three different types of preferences used

1. Activity-level Preferences
2. Application-level Preferences
3. Sharing preferences across Applications

Activity-level Preferences

Here Preferences can be retrieved only by a single activity To save preferences those are accessed only from a single activity

```
SharedPreferences sharedPref=getPreferences(Context.MODE_PRIVATE);
SharedPreferences.Editor sharedEditor= sharedPref.edit();
sharedEditor.putString("UserName", "Robot");
sharedEditor.commit();
```

SharedPreferences object can be retrieved by calling getPreferences(int mode) method which takes an integer value as parameter,Given below are the different modes available in Preferences API

- **Context.MODE_PRIVATE**

Using this mode allows the created file only accessible by applications with the same userID. Access is provided within the same Activity only.

- **Context.MODE_WORLD_READABLE**
  Using this mode allows the created file readable from other applications.
- **Context.MODE_WORLD_WRITABLE**
  Using this mode allows other applications to write to the file.

getPreferences(int mode) returns an instance of SharedPreferences.Editor and write the preference value with sharedEditor.putString(String key, String value) method.
Then we call sharedEditor.commit() to save the preferences to the file. Commit returns a boolean indicating the result of saving, true if successful and false if failed. Now to read the saved Preference value use sharedPref.getString(String key,String defaultValue) to return the value stored with specific key or default value if not found.

```
SharedPreferences sharedPref=getPreferences(Context.MODE_PRIVATE);
String userName= sharedPref.getString(“UserName”, "Enter name");
```

Application-level Preferences

Here Preferences can be shared and retrieved among all activities within the application To save preferences that can be accessed from all activities in the application

```
SharedPreferences
sharedPref=getSharedPreferences(“myappContext”,Context.MODE_WORLD_READABLE);
SharedPreferences.Editor sharedEditor = sharedPref.edit();
sharedEditor.putString("UserName", "Sam");
sharedEditor.commit();
```

Sharing preferences across Applications

Here Preferences can be shared and retrieved through all applications on the device
Preferences can be stored in one application and read them in another application and assume that both applications are in different package.

For reading the value from other applications in different packages is as below

```
  try {
     Context context = createPackageContext("com.pref.shared", 0);
    SharedPreferences sharedPref=context.getSharedPreferences("myappContext",
                                                          Context.MODE_PRIVATE);
    String userName= sharedPref.getString("UserName", "Name");
    txt.setText(userName);
  } catch (NameNotFoundException e) {
     Log.e(“error”, e.toString());
  }
```

Creating Preferences Activities

In Android saving preferences can be done by creating activities that extend PreferenceActivity.

PreferenceActivity is an activity that displays a set of built-in preferences related widgets that are defined in xml file. In this case it is not required to save and load the data in code every time, the OS does that for developer.

The preferences widgets that android provide are

- **CheckBoxPreference** Using this mode allows the created file only accessible by applications with the same userID. Access is provided within the same Activity only.
- **EditTextPreference** Using this mode allows the created file readable from other applications.
- **ListPreference** Using this mode allows other applications to write to the file.
- **RingtonePreference** Displays a list of existing ringtones in the system

Sample Example on Preferences

Given below is the Sample Example on Using Preferences and Creating Preferences Activities

Class: AndroidPreferences.java

```
package com.preferences;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class AndroidPreferences extends Activity {

   EditText editText;
   Button bt;

    /** Called when the activity is first created. */
     @Override
     public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        editText = (EditText) findViewById(R.id.editText1);
        bt = (Button) findViewById(R.id.button1);
        SharedPreferences prefs = getPreferences(Context.MODE_PRIVATE);
        String val = prefs.getString("pref", "some text");
        editText.setText(val);

         bt.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
             SharedPreferences prefs = getPreferences(Context.MODE_PRIVATE);
         SharedPreferences.Editor editor = prefs.edit();
```

```
          editor.putString("pref", editText.getText().toString());
         editor.commit();
          }
       });

        Button prefBtn = (Button) findViewById(R.id.prefButton);
        prefBtn.setOnClickListener(new OnClickListener() {
               public void onClick(View v) {
              Intent settingsActivity = new Intent(getApplicationContext(),
                        Preferences.class);
             startActivity(settingsActivity);
                        }
           });

         }
}
```

Class: Preferences.java

```
package com.preferences;

import android.app.Activity;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.preference.Preference;
import android.preference.PreferenceActivity;
import android.preference.Preference.OnPreferenceClickListener;
import android.widget.Toast;

public class Preferences extends PreferenceActivity {
   @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        addPreferencesFromResource(R.xml.preferences);
        //to show the custom preference
        Preference customPref = (Preference) findPreference("customPref");
        customPref.setOnPreferenceClickListener(new OnPreferenceClickListener() {
          public boolean onPreferenceClick(Preference preference) {
            Toast.makeText(getBaseContext(),"Custom Preference has been clicked",

Toast.LENGTH_LONG).show();
         SharedPreferences customSharedPreference = getSharedPreferences(
                                   "myCustomSharedPrefs", Activity.MODE_PRIVATE);
                                   SharedPreferences.Editor editor =
customSharedPreference.edit();
         editor.putString("myCustomPref","Custom Preference has been clicked");
         editor.commit();
         return true;
           }
       });
    }
}
```
The construction of the preferences.xml (saved in res/xml directory) layout is as below

preferences.xml

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android">
<PreferenceCategory android:title="First Category">
  <CheckBoxPreference
     android:title="Checkbox Preference"
     android:defaultValue="false"
     android:summary="enables a preference"
     android:key="prefCheckBox" />
  <ListPreference
     android:title="List Preference"
     android:summary="Select an Article from the list"
     android:key="prefList"
     android:defaultValue="digiGreen"
     android:entries="@array/pref_items"
     android:entryValues="@array/pref_items_values" />
</PreferenceCategory>
<PreferenceCategory android:title="Second Category">
    <EditTextPreference
       android:name="EditText Preference"
       android:summary="Enter a string to save in preferences"
       android:defaultValue="Enter Text"
       android:title="Edit This Text"
       android:key="prefEditText" />
    <RingtonePreference
       android:name="Ringtone Preference"
       android:summary="Select a ringtone"
       android:title="Ringtones"
       android:key="prefRingtone" />
    <PreferenceScreen
       android:key="SecondScreenPref"
       android:title="Second PreferenceScreen"
       android:summary="This is a second PreferenceScreen">
    <EditTextPreference
       android:name="Another EditText Preference"
       android:summary="This is a preference in the second PreferenceScreen"
       android:title="Edit text"
       android:key="SecondEditTextPref" />
    </PreferenceScreen>
    <Preference
       android:title="Custom Preference"
       android:summary="This is a custom preference"
       android:key="customPref" />
    </PreferenceCategory>
    </PreferenceScreen>
```
The ListPreference can be associated with String array resources as its key/value entries

Strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">AndroidPreferences!</string>
    <string name="app_name">Preferences in Android</string>
    <string name="settings_name">Settings in Android</string>
<string-array name="pref_items">
   <item>Article 1</item>
   <item>Article 2</item>
   <item>Article 3</item>
   <item>Article 4</item>
   <item>Article 5</item>
   <item>Article 6</item>
</string-array>
<string-array name="pref_items_values">
   <item>1</item>
   <item>2</item>
   <item>3</item>
   <item>4</item>
   <item>5</item>
   <item>6</item>
</string-array>
</resources>
```
The main.xml in layout looks as below

main.xml

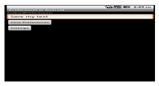
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
         android:orientation="vertical"
```

```
        android:layout_width="fill_parent"
        android:layout_height="fill_parent">
  <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />
        <EditText
            android:id="@+id/editText1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="EditText">
        </EditText>
        <Button
            android:text="Save Preferences"
            android:id="@+id/button1"
            android:layout_width="wrap_content"
             android:layout_height="wrap_content">
        </Button>
        <Button
            android:text="Settings"
            android:id="@+id/prefButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content">
        </Button>
</LinearLayout>
```

Add the following tag in AndroidManifest.xml and add new string item with the name settings_name with title “Settings in Android” in string.xml

```
<activity android:name=".Preferences"
                    android:label="@string/settings_name">
</activity>
```

On running the above sample example output will be as shown in below figures



In figure 1 click on Save preferences to save entered text in preferences and click on Settings to go Preference Activities.

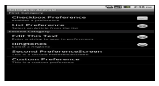


Figure 2 shows the list of different preferences

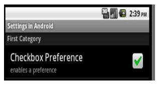


Figure 3 shows the Checkbox preference

Figure 4 shows the List preference

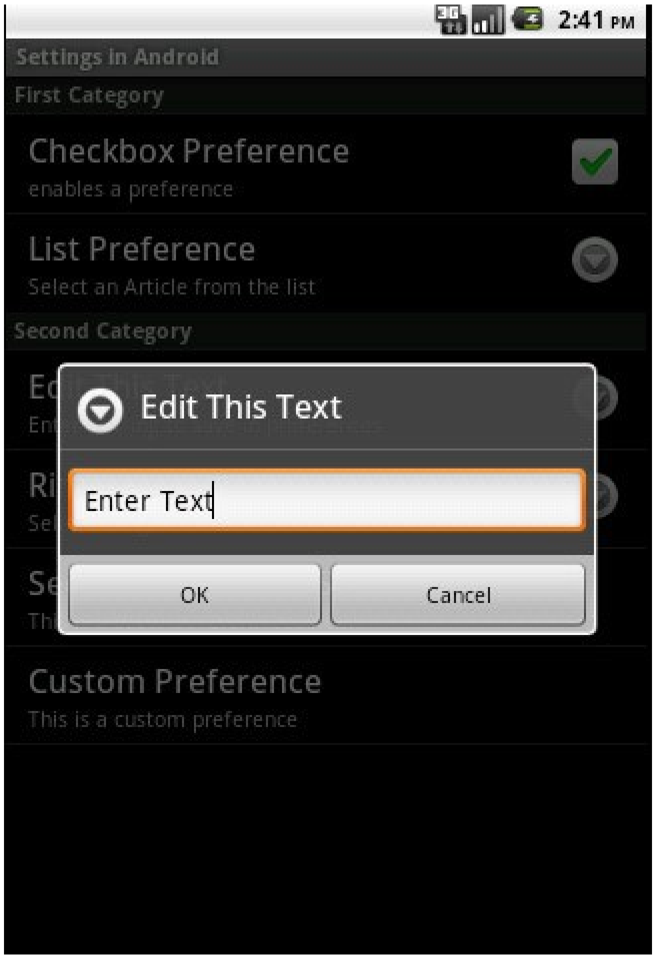


Figure 5 shows the edit preference

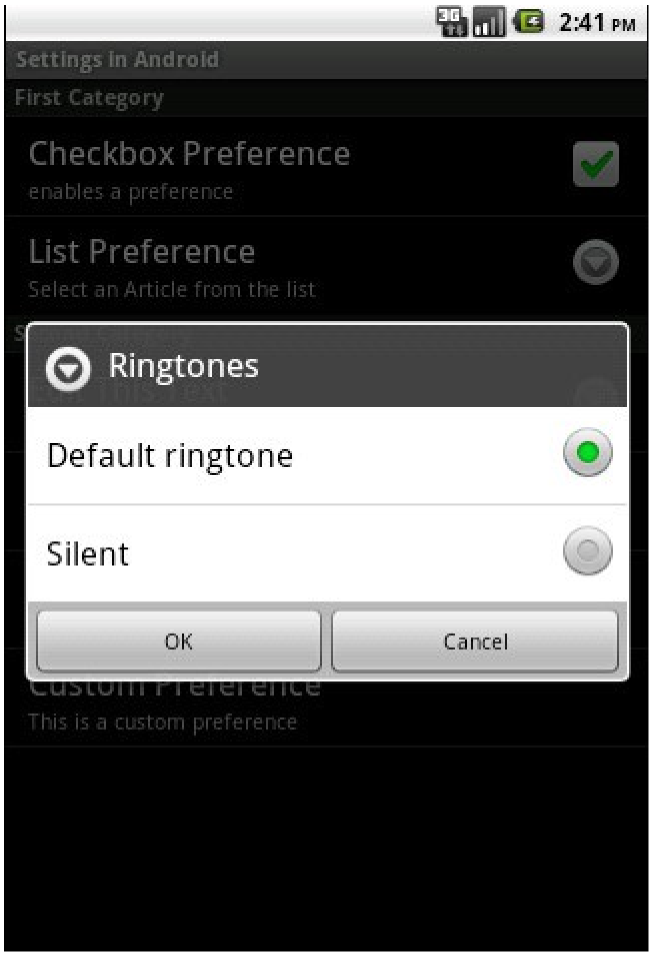


Figure 6 shows the Ringtone preference

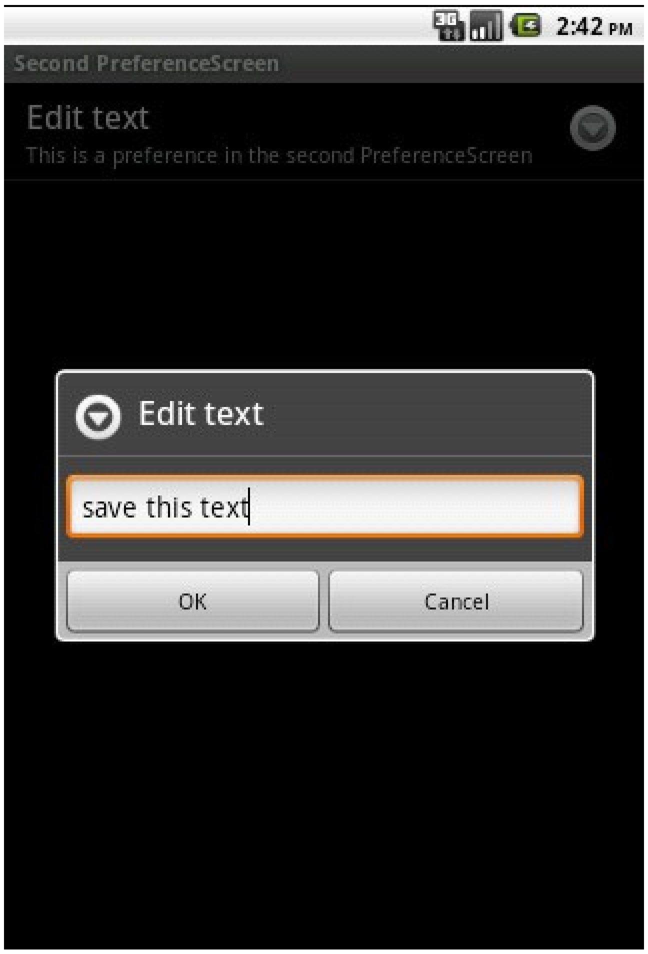


Figure 7 shows the second edit preference

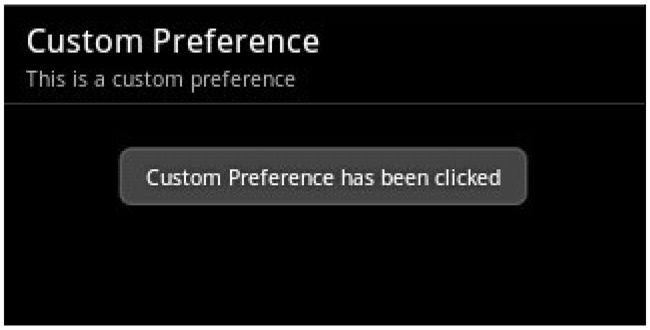

Custom Preference
This is a custom preference
Custom Preference has been clicked

Figure 8 shows the custom preference

ref:http://developer.samsung.com/android/technical-docs/Using-Preferences-in-Android