

## Module 3 : UI Design and Data storage

### DATA STORAGE IN ANDROID

For Android, there are primarily three basic ways of persisting data:

- A lightweight mechanism known as *shared preferences* to save small chunks of data
- Traditional file systems
- A relational database management system through the support of SQLite databases

### Saving And Loading User Preference

Android provides the Shared Preferences object to help you save simple application data. For example, your application may have an option to allow users to specify the font size of the text displayed in your application. In this case, your application needs to remember the size set by the user so that the next time he or she uses the application again, your application can set the size appropriately. In order to do so, you have several options. You can save the data to a file, but you have to perform some file management routines, such as writing the data to the file, indicating how many characters to read from it, and so on. Also, if you have several pieces of information to save, such as text size, font name, preferred background color, and so on, then the task of writing to a file becomes more onerous. An alternative to writing to a text file is to use a database, but saving simple data to a database is overkill, both from a developer's point of view and in terms of the application's run-time performance. Using the Shared Preferences object, however, you save the data you want through the use of key/ value pairs — specify a key for the data you want to save, and then both it and its value will be saved automatically to an XML file for you.

### Using `getSharedPreferences()`

```
<?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" >
```

```
<SeekBar
```

```
    android:id="@+id/SeekBar01"
```

```
    android:layout_width="fill_parent"
```

```
    android:layout_height="wrap_content" />
```

```
<TextView
```

```
    android:id="@+id/TextView01"
```

```
    android:layout_width="fill_parent"
```

```
android:layout_height="wrap_content"
android:text="@string/hello" />
<EditText
android:id="@+id/EditText01"
android:layout_width="fill_parent"
android:layout_height="wrap_content" />
```

```
<Button
```

```
android:id="@+id/btnSave"
android:text="Save"
android:layout_width="wrap_content"
android:layout_height="wrap_content" />
```

```
</LinearLayout>
```

```
package net.learn2develop.SharedPreferences;
import android.app.Activity;
import android.os.Bundle;
import android.content.SharedPreferences;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.SeekBar;
import android.widget.SeekBar.OnSeekBarChangeListener;
import android.widget.Toast;
public class MainActivity extends Activity {
    private SharedPreferences prefs;
    private String prefName = "MyPref";
    private EditText editText;
    private SeekBar seekBar;
    private Button btn;
    private static final String FONT_SIZE_KEY = "fontsize";
    private static final String TEXT_VALUE_KEY = "textvalue";
    /** 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.EditText01);
        seekBar = (SeekBar) findViewById(R.id.SeekBar01);
```

```

btn = (Button) findViewById(R.id.btnSave);
btn.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        !---get the SharedPreferences object---
        prefs = getSharedPreferences(prefName, MODE_PRIVATE);
        SharedPreferences.Editor editor = prefs.edit();
        !---save the values in the EditText view to preferences---
        editor.putFloat(FONT_SIZE_KEY, editText.getTextSize());
        editor.putString(TEXT_VALUE_KEY, editText.getText().toString());
        !---saves the values---
        editor.commit();
        !---display file saved message---
        Toast.makeText(getApplicationContext(),
            "Font size saved successfully!",
            Toast.LENGTH_SHORT).show();
    }
});
!---load the SharedPreferences object--
SharedPreferences prefs = getSharedPreferences(prefName, MODE_PRIVATE);
!---set the TextView font size to the previously saved values---
float fontSize = prefs.getFloat(FONT_SIZE_KEY, 12);
!---init the SeekBar and EditText---
seekBar.setProgress((int) fontSize);
editText.setText(prefs.getString(TEXT_VALUE_KEY, ""));
editText.setTextSize(seekBar.getProgress());
seekBar.setOnSeekBarChangeListener(new OnSeekBarChangeListener() {
    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {
    }
    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {
    }
    @Override
    public void onProgressChanged(SeekBar seekBar, int progress,
        boolean fromUser) {
        !---change the font size of the EditText---
        editText.setTextSize(progress);
    }
});

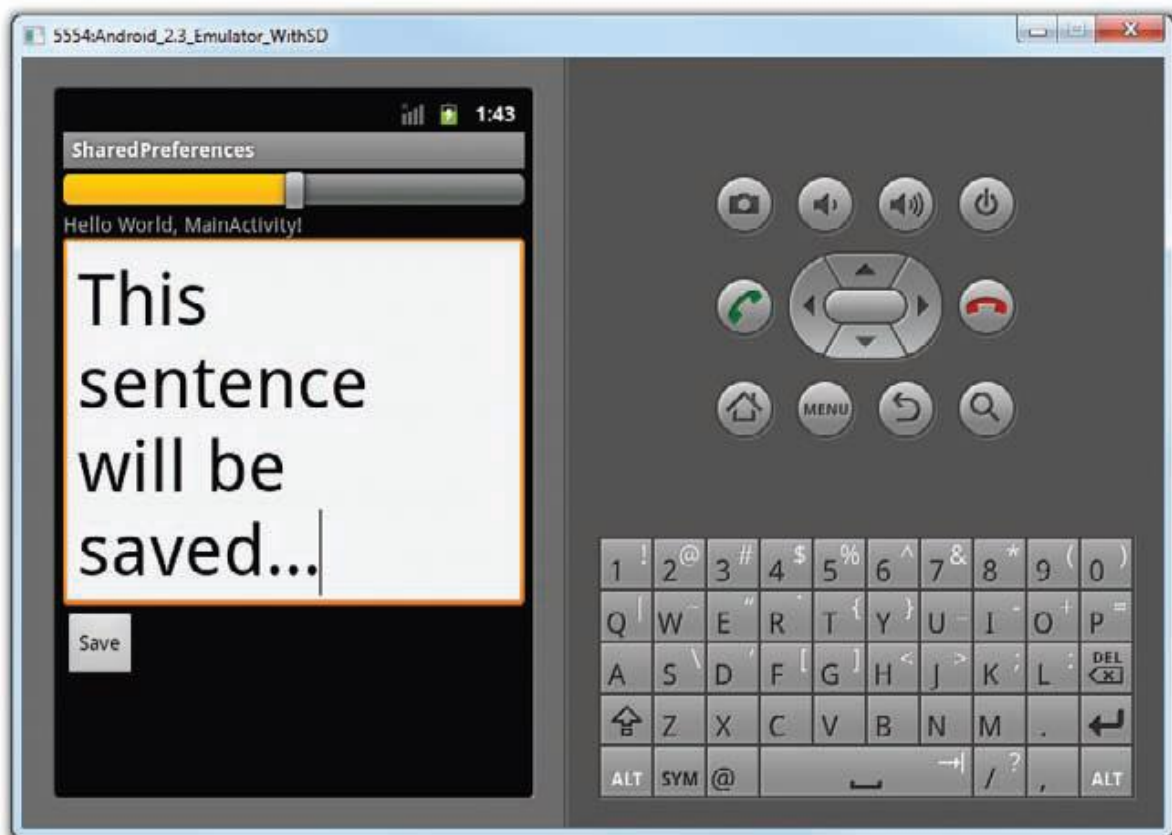
```

```

}
});
}
}

```

Enter some text into the EditText view and then change its font size by adjusting the SeekBar view (see Figure 6-2). Click Save. debug the application on the Android Emulator again. The application now displays the same text that you entered earlier using the same font size set earlier.



## How It Works

To use the SharedPreferences object, you use the `getSharedPreferences()` method, passing it the name of the shared preferences file (in which all the data will be saved), as well as the mode in which it should be opened:

```
private SharedPreferences prefs;
```

...

```
//---get the SharedPreferences object---
```

```
prefs = getSharedPreferences(prefName, MODE_PRIVATE);
```

```
SharedPreferences.Editor editor = prefs.edit();
```

The `MODE_PRIVATE` constant indicates that the shared preference file can only be opened by the application that created it. The `Editor` class allows you to save key/value pairs to the preferences file by exposing methods such as the following:

- >> `putString()`
- >> `putBoolean()`
- >> `putLong()`
- >> `putInt()`
- >> `putFloat()`

When you are done saving the values, call the `commit()` method to save the changes:

```
//---save the values in the EditText view to preferences---
editor.putFloat(FONT_SIZE_KEY, editText.getTextSize());
editor.putString(TEXT_VALUE_KEY, editText.getText().toString());
//---saves the values---
editor.commit();
```

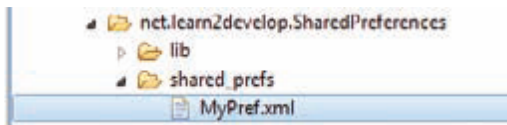
When the activity is loaded, you first obtain the `SharedPreferences` object and then retrieve all the values saved earlier:

```
//---load the SharedPreferences object---
SharedPreferences prefs = getSharedPreferences(prefName, MODE_PRIVATE);
//---set the TextView font size to the previously saved values---
float fontSize = prefs.getFloat(FONT_SIZE_KEY, 12);
//---init the SeekBar and EditText---
seekBar.setProgress((int) fontSize);
editText.setText(prefs.getString(TEXT_VALUE_KEY, ""));
editText.setTextSize(seekBar.getProgress());
```

The shared preferences file is saved as an XML file in the `/data/data/<package_name>/shared_prefs` folder (see Figure 6-3).

Its content is shown here (formatted for clarity):

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
<string name="textvalue">This is so cool!</string>
<float name="fontsize" value="75.0" />
</map>
```



## Using `getPreferences()`

In the previous section, you used the `SharedPreferences` object by supplying it with a name, like this:

```
//---get the SharedPreferences object---
```

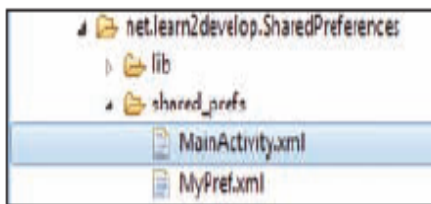
```
prefs = getSharedPreferences(prefName, MODE_PRIVATE);
```

In this case, the information saved inside the `SharedPreferences` object is visible to all the activities within the same application. However, if you don't need to share the data between activities, you can use the `getPreferences()` method, like this:

```
//---get the SharedPreferences object---
```

```
prefs = getPreferences(MODE_PRIVATE);
```

The `getPreferences()` method does not require a name, and the data saved is restricted to the activity that created it. In this case, the filename used for the preferences file will be named after the activity that created it (see Figure 6-4).



**FIGURE 6-4**