# Xamarin Android: Create Xamarin Android Animating Activities

**Introduction**  
Android animations show the visual effects of transactions of the activities and fragments. Here, we will create four types of transactions.

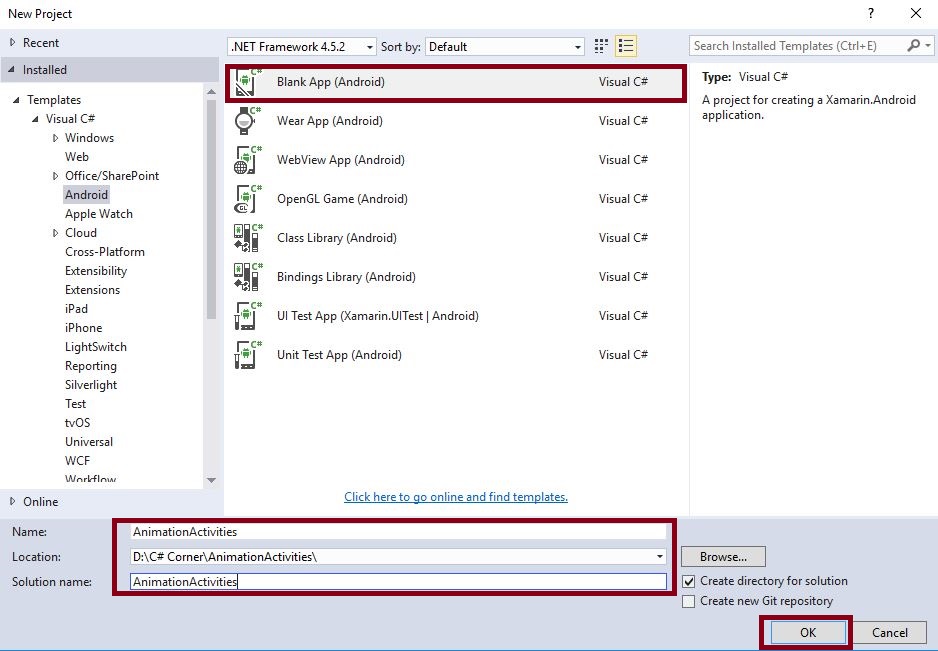
## **LEFT TO RIGHT ANIMATION**

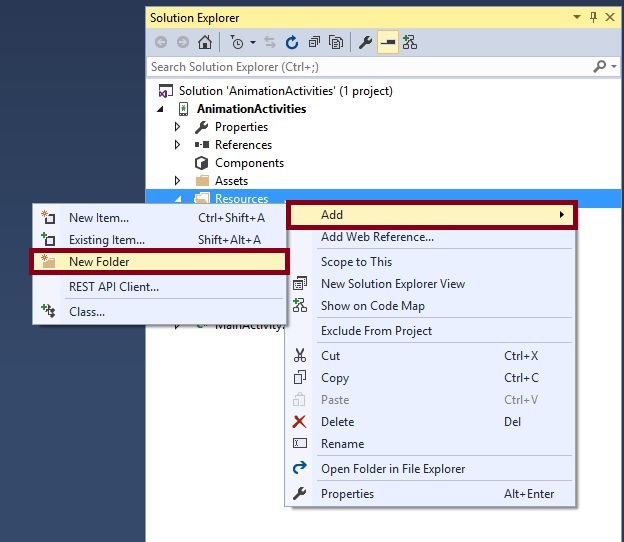
## **RIGHT TO LEFT ANIMATION**

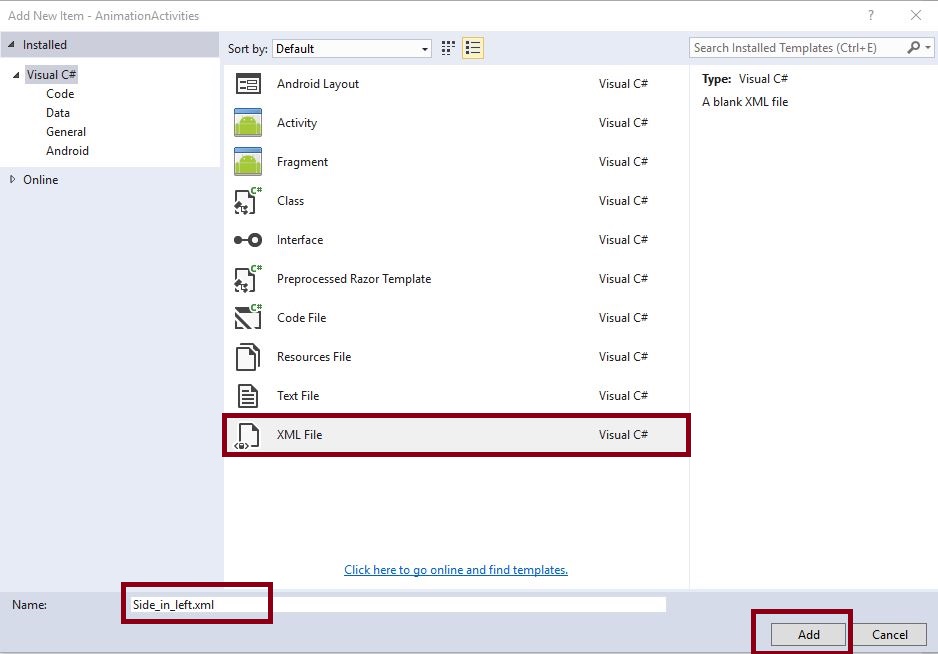
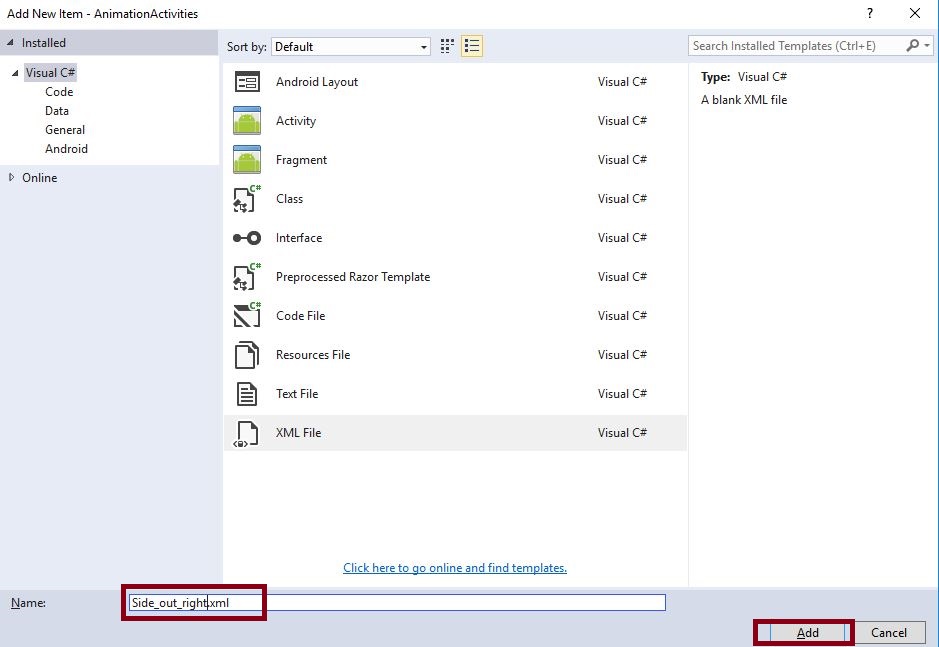
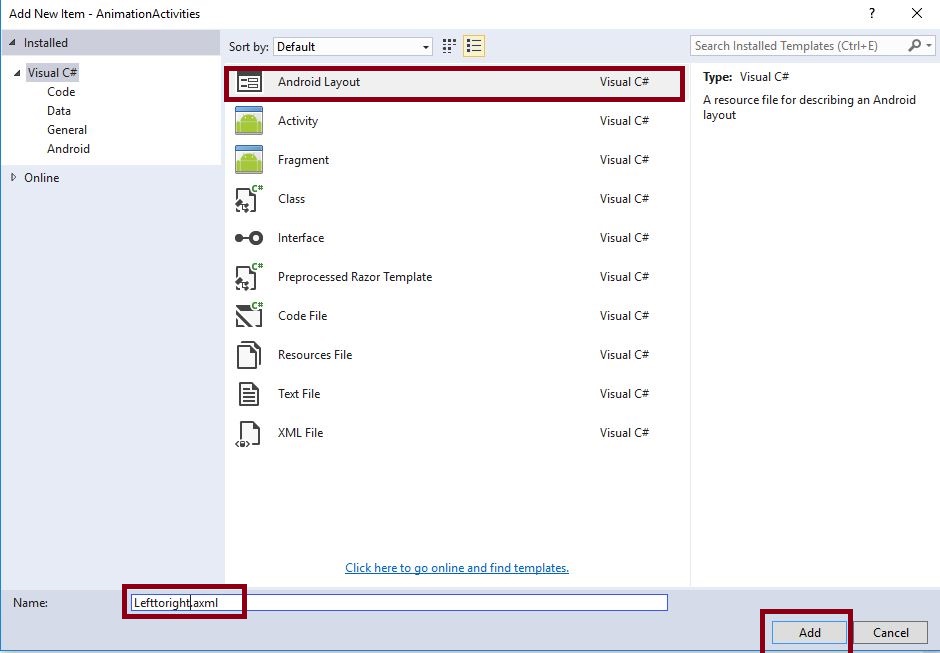
## **BOTTOM TO TOP ANIMATION**

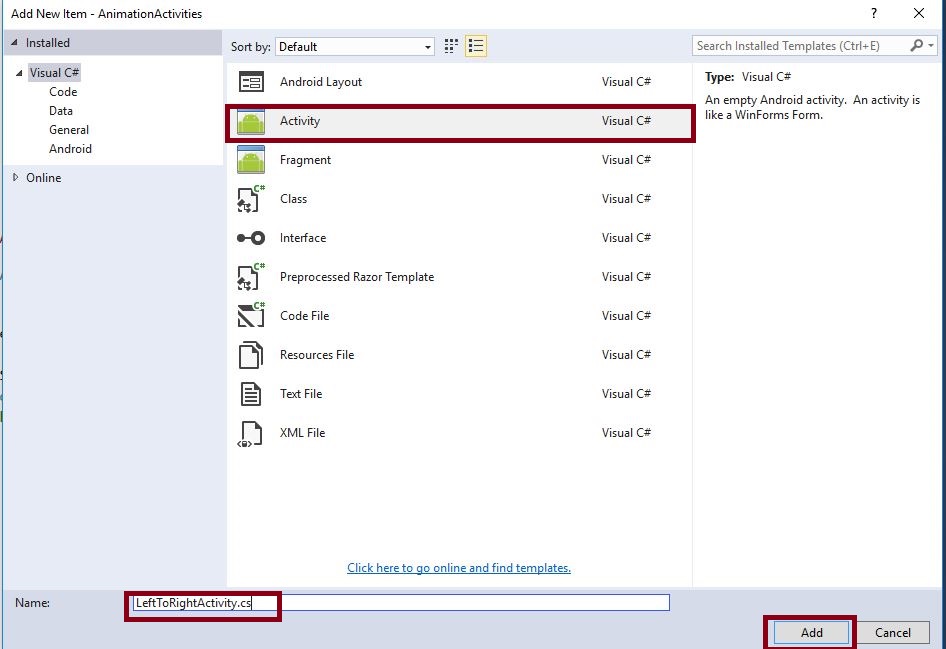
## **TOP TO BOTTOM ANIMATION**

 Let’s start,

LEFT TO RIGHT ANIMATION  
  
**Step 1**Open Visual Studio->New Project->Templates->Visual C#->Android->Blank App  
  
Select Blank App. Give the Project Name and Project Location.  
  


**Step 2:** Go to Solution Explorer-> Project Name-> Resources, followed by Right click to Add-> New Folder and give the name for Anim.  
  


**Step 3**First, we create Left to Right Animation. Thus, we create two XML resources. Go to Solution Explorer-> Project Name-> Resources-> Anim and right click to Add-> New Item, which will open the new dialog box. This dialog box is required to select XML and give the name for Side\_in\_left.xml and one more resource is Side\_out\_right.xml.  
  
  
  
  
  
**Step 4**Add Lefttoright Design view. Go to Solution Explorer-> Project Name-> Resources-> Layout. Right click Add->New Item and open new dialog box. This dialog box is required to select XML and give the name for Lefttoright.axml.  
  


**Step 5**Add Lefttoright Activity page. Go to Solution Explorer-> Project Name, followed by right clicking Add->New Item, open new dialog box. This dialog box will select activity and give the name for LeftToRightActivity.cs.  
  
  
  
**Step 6:** Open Solution Explorer-> Project Name-> Resources->Anim -> Side\_in\_left.xml. Click to open and add the code, given below:

1. <?xml version="1.0" encoding="utf-8" ?>
2. <set
3. xmlns:android="http://schemas.android.com/apk/res/android"
4. android:interpolator="@android:anim/linear\_interpolator">
5. <translate android:duration="600"
6. android:fromXDelta="-100%"
7. android:toXDelta="0%"/>
8. </set>

**Step 7:** Open Solution Explorer-> Project Name-> Resources-> Anim -> Side\_out\_right.xml and click to open. Add the code, given below:

1. <?xml version="1.0" encoding="utf-8" ?>
2. <set
3. xmlns:android="http://schemas.android.com/apk/res/android"
4. android:interpolator="@android:anim/linear\_interpolator">
5. <translate android:duration="600"
6. android:fromXDelta="0%"
7. android:toXDelta="100%"/>
8. </set>

**Step 8: O**pen Solution Explorer-> Project Name-> Resources-> Layout -> Lefttoright.axml. Click to open and add the code, given below:

1. <TextView
2. android:text="LEFT TO RIGHT"
3. android:textAppearance="?android:attr/textAppearanceLarge"
4. android:layout\_width="match\_parent"
5. android:layout\_height="wrap\_content"
6. android:id="@+id/textView1" />

**Step 9:** Open Solution Explorer-> Project Name-> LeftToRightActivity.cs. Click to open and add the code, given below:  
  
SetContentView(Resource.Layout.Lefttoright);

**RIGHT TO LEFT ANIMATION**  
**Step 10:** Go to Solution Explorer-> Project Name-> Resources-> Anim, followed by right click to add-> New Item. Open the new dialog box. This dialog box is required to select XML and give the name for Side\_in\_right.xml and one more resource is Side\_out\_left.xml  
  
**Step 11:** Now, add RighttoLeft Design view. Go to Solution Explorer-> Project Name-> Resources-> Layout and right click to Add-> New Item, followed by opening the new dialog box. This dialog box is required to select XML and give the name for RighttoLeft.axml.  
  
**Step 12:** Now, add RighttoLeft Activity page. Go to Solution Explorer-> Project Name, followed by right clicking to Add-> New Item, followed by opening the new dialog box. This dialog box is required to select activity and give the name for RighttoLeftActivity.cs.  
  
**Step 13:** Now, open Solution Explorer-> Project Name->Resources->Anim -> Side\_in\_right.xml and click to open. Add the code, given below:

1. <?xml version="1.0" encoding="utf-8" ?>
2. <set
3. xmlns:android="http://schemas.android.com/apk/res/android"
4. android:interpolator="@android:anim/linear\_interpolator">
5. <translate android:duration="600"
6. android:fromXDelta="100%"
7. android:toXDelta="0%" />
8. </set>

**Step 14:** Open Solution Explorer-> Project Name->Resources->Anim -> Side\_out\_left.xml and click to open. Add the code, given below:

1. <?xml version="1.0" encoding="utf-8" ?>
2. <set
3. xmlns:android="http://schemas.android.com/apk/res/android"
4. android:interpolator="@android:anim/linear\_interpolator">
5. <translate android:duration="600"
6. android:fromXDelta="0%"
7. android:toXDelta="-100%"/>
8. </set>

**Step 15:** Open Solution Explorer-> Project Name->Resources->Layout -> RighttoLeft.axml and click to open. Add the code, given below:

1. <TextView
2. android:text="RIGHT TO LEFT"
3. android:textAppearance="?android:attr/textAppearanceLarge"
4. android:layout\_width="match\_parent"
5. android:layout\_height="wrap\_content"
6. android:id="@+id/textView1" />

**Step 16:** Open Solution Explorer-> Project Name-> RighttoLeftActivity.cs and click to open. Add the code, given below:  
  
SetContentView(Resource.Layout.Righttoleft);

**BOTTOM TO TOP ANIMATION**  
**Step 17:** Go to Solution Explorer-> Project Name->Resources-> Anim, followed by right click to Add-> New Item, open new dialog box. This dialog box is required to select XML and give the name for Pushup\_in.xml. One more resource is Pushup\_out.xml.  
  
**Step 18:** Add bottomtotop design view. Go to Solution Explorer-> Project Name->Resources-> Layout, followed by right clicking to Add->New Item. Open the new dialog box. This dialog box is required to select XML and give the name for bottomtotop.axml.  
  
**Step 19:** Add bottomtotopActivity page. Go to Solution Explorer-> Project Name, right click to Add-> New Item. New dialog box will open. This dialog box is required to select activity and give the name for bottomtotop.cs.  
  
**Step 20:** Open Solution Explorer-> Project Name-> Resources-> Anim-> Pushup\_in.xml. Click to open and add the code, given below:.

1. <?xml version="1.0" encoding="utf-8" ?>
2. <set
3. xmlns:android="http://schemas.android.com/apk/res/android"
4. android:interpolator="@android:anim/linear\_interpolator">
5. <translate android:fromYDelta="100%p"
6. android:toYDelta="0"
7. android:duration="600"/>
8. </set>

**Step 21:** Open Solution Explorer-> Project Name->Resources->Anim -> Pushup\_out.xml. Click to open and add the code, given below:

1. <?xml version="1.0" encoding="utf-8" ?>
2. <set
3. xmlns:android="http://schemas.android.com/apk/res/android"
4. android:interpolator="@android:anim/linear\_interpolator">
5. <translate android:fromYDelta="0"
6. android:toYDelta="-100%p"
7. android:duration="600"/>
8. </set>

**Step 22:** Open Solution Explorer-> Project Name-> Resources-> Layout -> RighttoLeft.axml. Click to open and add the code, given below:

1. <TextView
2. android:text="BOTTOM TO TOP"
3. android:textAppearance="?android:attr/textAppearanceLarge"
4. android:layout\_width="match\_parent"
5. android:layout\_height="wrap\_content"
6. android:id="@+id/textView1" />

**Step 23:** Open Solution Explorer-> Project Name-> RighttoLeftActivity.cs. Click to open and add the code, given below:  
  
SetContentView(Resource.Layout.bottomtotop);

**TOP TO BOTTOM ANIMATION**  
**Step 24:** Go to Solution Explorer-> Project Name-> Resources-> Anim, followed by right click to Add-> New Item. Open new dialog box. This dialog box is required to select XML and give the name for Pushdown\_in.xml. One more resource is Pushdown\_out.xml.  
  
**Step 25:** Add Toptobottom Design view.go to Solution Explorer-> Project Name->Resources->Layout then Right Click to Add->New Item then open new Dialog box. That Dialog box to Select XML and give Name for Toptobottom.axml.  
  
**Step 26:** Add Toptobottom Activity page. Go to Solution Explorer-> Project Name, followed by right clicking to Add->New Item, open new dialog box. This dialog box is required to select activity and give name for Toptobottom Activity.cs.  
  
**Step 27:** Open Solution Explorer-> Project Name->Resources-> Anim -> Pushdown\_in.xml. Click to open and add the code, given below:

1. <?xml version="1.0" encoding="utf-8" ?>
2. <set
3. xmlns:android="http://schemas.android.com/apk/res/android"
4. android:interpolator="@android:anim/linear\_interpolator">
5. <translate android:fromYDelta="-100%p"
6. android:toYDelta="0"
7. android:duration="600"/>
8. </set>

**Step 28:** Open Solution Explorer-> Project Name->Resources-> Anim -> Pushdown\_out.xml. Click to open and add the code, given below:

1. <?xml version="1.0" encoding="utf-8" ?>
2. <set
3. xmlns:android="http://schemas.android.com/apk/res/android"
4. android:interpolator="@android:anim/linear\_interpolator">
5. <translate android:fromYDelta="0"
6. android:toYDelta="100%p"
7. android:duration="600" />
8. </set>

**Step 29:** Open Solution Explorer-> Project Name-> Resources-> Layout -> RighttoLeft.axml. Click to open and add the code, given below:

1. <TextView
2. android:text="TOP TO BOTTOM"
3. android:textAppearance="?android:attr/textAppearanceLarge"
4. android:layout\_width="match\_parent"
5. android:layout\_height="wrap\_content"
6. android:id="@+id/textView1" />

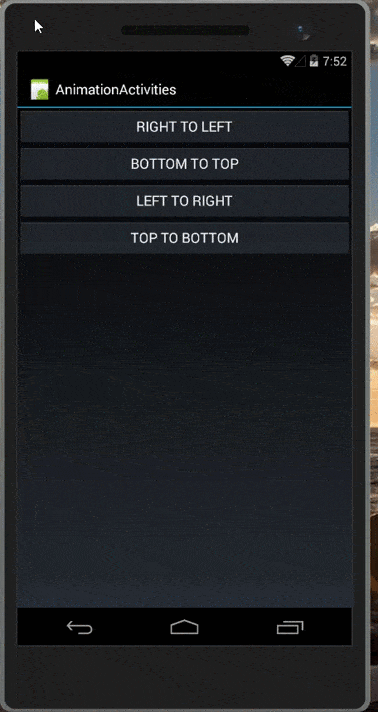
**Step 30:** Open Solution Explorer-> Project Name-> RighttoLeftActivity.cs. Click to open and add the code, given below:  
  
SetContentView(Resource.Layout. Toptobottom);

**Step 31:** Finally, create button click event each animation. Open Main.axml.Solution Explorer-> Project Name->Resources->Layout -> Main.axml. Click to open and add the code,, given below:

1. <Button
2. android:text="RIGHT TO LEFT"
3. android:layout\_width="match\_parent"
4. android:layout\_height="wrap\_content"
5. android:id="@+id/righttoleft" />
6. <Button
7. android:layout\_width="match\_parent"
8. android:layout\_height="wrap\_content"
9. android:id="@+id/bottomtotop"
10. android:text="BOTTOM TO TOP" />
11. <Button
12. android:text="LEFT TO RIGHT"
13. android:layout\_width="match\_parent"
14. android:layout\_height="wrap\_content"
15. android:id="@+id/lefttoright" />
16. <Button
17. android:layout\_width="match\_parent"
18. android:layout\_height="wrap\_content"
19. android:id="@+id/toptobottom"
20. android:text="TOP TO BOTTOM" />

**Step 32:** Open Solution Explorer-> Project Name-> MainActivity.cs. Click to open and add the code, given below:

1. **public** **class** MainActivity: Activity {
2. **int** count = 1;
3. Button Righttoleft;
4. Button bottomtotop;
5. Button lefttoright;
6. Button toptobottom;
7. **protected** **override** **void** OnCreate(Bundle bundle) {
8. **base**.OnCreate(bundle);
9. // Set our view from the "main" layout resource
10. SetContentView(Resource.Layout.Main);
11. Righttoleft = FindViewById < Button > (Resource.Id.righttoleft);
12. bottomtotop = FindViewById < Button > (Resource.Id.bottomtotop);
13. lefttoright = FindViewById < Button > (Resource.Id.lefttoright);
14. toptobottom = FindViewById < Button > (Resource.Id.toptobottom);
15. Righttoleft.Click += Righttoleft\_Click;
16. bottomtotop.Click += bottomtotop\_Click;
17. lefttoright.Click += Lefttoright\_Click;
18. toptobottom.Click += Toptobottom\_Click;
19. }
20. **private** **void** Toptobottom\_Click(**object** sender, EventArgs e) {
21. Intent intent;
22. intent = **new** Intent(**this**, **typeof**(ToptoBottomActivity));
23. StartActivity(intent);
24. OverridePendingTransition(Resource.Animation.@Pushdown\_in, Resource.Animation.@Pushdown\_out);
25. }
26. **private** **void** Lefttoright\_Click(**object** sender, EventArgs e) {
27. Intent intent;
28. intent = **new** Intent(**this**, **typeof**(LeftToRightActivity));
29. StartActivity(intent);
30. OverridePendingTransition(Resource.Animation.@Side\_in\_left, Resource.Animation.@Side\_out\_right);
31. }
32. **private** **void** bottomtotop\_Click(**object** sender, EventArgs e) {
33. Intent intent;
34. intent = **new** Intent(**this**, **typeof**(BottomToTopActivity));
35. StartActivity(intent);
36. OverridePendingTransition(Resource.Animation.@Pushup\_in, Resource.Animation.@Pushup\_out);
37. }
38. **private** **void** Righttoleft\_Click(**object** sender, EventArgs e) {
39. Intent intent;
40. intent = **new** Intent(**this**, **typeof**(RighttoLeftActivity));
41. StartActivity(intent);
42. OverridePendingTransition(Resource.Animation.@Side\_in\_right, Resource.Animation.@Side\_out\_left);
43. }
44. }

**Step 33:** Press F5 or build and run the Application.  
  
  
  
Finally, we successfully created Xamarin Android Animating activities.