在drawable文件夹中建立如下旋转动画文件

<?xml version="1.0" encoding="utf-8"?>  
<animated-rotate xmlns:android="http://schemas.android.com/apk/res/android"      
    android:drawable="@drawable/loading1"      
    android:pivotX="50%"      
    android:pivotY="50%" />

其中loading1即为您想用的图片

二 在布局文件中设置progressbar

 <ProgressBar  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
         android:indeterminateDrawable="@drawable/progressbg"  
         android:indeterminateOnly="true"  
         android:indeterminateBehavior="repeat"  
      
       />

[android 自定义水平的ProgressBar](http://blog.csdn.net/jiaruihua_blog/article/details/12783943)

先看效果图：

http://img.blog.csdn.net/20131016174039234?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvamlhcnVpaHVhX2Jsb2c=/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/SouthEast

代码：

新建drawable

新建一个drawable/progressbarclor

<?xml version="1.0" encoding="utf-8"?>  
<layer-list xmlns:android="http://schemas.android.com/apk/res/android" >  
    <item  
        android:id="@android:id/progress"  
        android:drawable="@drawable/common\_button\_red"/>  
</layer-list>

新建drawable/progresssape

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android"  
    android:shape="rectangle" >  
  
  
    <corners android:radius="10dp" />  
  
  
    <solid android:color="#ffffff" />  
  
  
</shape>

布局：

<ProgressBar  
                android:background="@drawable/progresssape"  
                android:id="@+id/pb"  
                android:layout\_width="wrap\_content"  
                android:layout\_height="10dp"  
                android:layout\_marginLeft="8dp"  
                android:layout\_marginRight="8dp"  
                android:layout\_weight="1"  
                android:indeterminateOnly="false"  
                android:max="100"  
                android:progressDrawable="@drawable/progressbarclor" />

http://winwyf.blog.51cto.com/4561999/857867

Android原生控件只有横向进度条一种，而且没法变换样式，比如原生rom的样子

[http://img1.51cto.com/attachment/201205/215458859.jpg](http://img1.51cto.com/attachment/201205/215458859.jpg)

很丑是吧，当伟大的产品设计要求更换前背景，甚至纵向，甚至圆弧状的，咋办，比如

[](http://img1.51cto.com/attachment/201205/215749775.png)

ok，我们开始吧：

一）变换前背景

先来看看progressbar的属性：

1. **<ProgressBar**
2. android:id="@+id/progressBar"
3. style="?android:attr/progressBarStyleHorizontal"
4. android:layout\_width="match\_parent"
5. android:layout\_height="wrap\_content"
6. android:layout\_margin="5dip"
7. android:layout\_toRightOf="@+id/progressBarV"
8. android:indeterminate="false"
9. android:padding="2dip"
10. android:progress="50" **/>**

根据style="?android:attr/progressBarStyleHorizontal"，我们找到源码中的style.xml

1. **<style** name="Widget.ProgressBar.Horizontal"**>**
2. **<item** name="android:indeterminateOnly"**>**false**</item>**
3. **<item** name="android:progressDrawable"**>**@android:drawable/progress\_horizontal**</item>**
4. **<item** name="android:indeterminateDrawable"**>**@android:drawable/progress\_indeterminate\_horizontal**</item>**
5. **<item** name="android:minHeight"**>**20dip**</item>**
6. **<item** name="android:maxHeight"**>**20dip**</item>**
7. **</style>**

看到

**<item** name="android:progressDrawable"**>**@android:drawable/progress\_horizontal**</item>**

木有，继续发掘源码，找到drawable下面的progress\_horizontal.xml，这就是我们今天的主角了：

1. **<layer-list** xmlns:android="http://schemas.android.com/apk/res/android"**>**
3. **<item** android:id="@android:id/background"**>**
4. **<shape>**
5. **<corners** android:radius="5dip" **/>**
6. **<gradient**
7. android:startColor="#ff9d9e9d"
8. android:centerColor="#ff5a5d5a"
9. android:centerY="0.75"
10. android:endColor="#ff747674"
11. android:angle="270"
12. **/>**
13. **</shape>**
14. **</item>**
16. **<item** android:id="@android:id/secondaryProgress"**>**
17. **<clip>**
18. **<shape>**
19. **<corners** android:radius="5dip" **/>**
20. **<gradient**
21. android:startColor="#80ffd300"
22. android:centerColor="#80ffb600"
23. android:centerY="0.75"
24. android:endColor="#a0ffcb00"
25. android:angle="270"
26. **/>**
27. **</shape>**
28. **</clip>**
29. **</item>**
31. **<item** android:id="@android:id/progress"**>**
32. **<clip>**
33. **<shape>**
34. **<corners** android:radius="5dip" **/>**
35. **<gradient**
36. android:startColor="#ffffd300"
37. android:centerColor="#ffffb600"
38. android:centerY="0.75"
39. android:endColor="#ffffcb00"
40. android:angle="270"
41. **/>**
42. **</shape>**
43. **</clip>**
44. **</item>**
46. **</layer-list>**

 看到android:id="@android:id/progress"木有，看到android:id="@android:id/secondaryProgress"木有

把这个文件复制到自己工程下的drawable，就可以随心所欲的修改shape的属性，渐变，圆角等等

那么怎么放一个图片进去呢，ok，新建progress\_horizontal1.xml：

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<layer-list** xmlns:android="http://schemas.android.com/apk/res/android"**>**
4. **<item** android:id="@android:id/progress" android:drawable="@drawable/progressbar" **/>**
6. **</layer-list>**

在android:drawable中指定你处理好的图片

然后回到布局中

1. **<ProgressBar**
2. android:id="@+id/progressBar1"
3. android:layout\_width="match\_parent"
4. android:layout\_height="wrap\_content"
5. android:layout\_below="@+id/progressBar"
6. android:layout\_margin="5dip"
7. android:layout\_toRightOf="@+id/progressBarV"
8. android:background="@drawable/progress\_bg"
9. android:indeterminate="false"
10. android:indeterminateOnly="false"
11. android:maxHeight="20dip"
12. android:minHeight="20dip"
13. android:padding="2dip"
14. android:progress="50"
15. android:progressDrawable="@drawable/progress\_horizontal1" **/>**

android:background="@drawable/progress\_bg"指定背景

android:progressDrawable="@drawable/progress\_horizontal1"前景使用上面的progress\_horizontal1

ok，搞定

[http://img1.51cto.com/attachment/201205/221809734.jpg](http://img1.51cto.com/attachment/201205/221809734.jpg)

 注意看，四角还是有圆倒角的，貌似是系统自己加上去的，总之我的图片里面是没有做这个倒角处理的

二）纵向进度条

还是得从源码入手，看回progress\_horizontal.xml

1. **<item** android:id="@android:id/progress"**>**
2. **<clip>**
3. **<shape>**
4. **<corners** android:radius="5dip" **/>**
5. **<gradient**
6. android:startColor="#ffffd300"
7. android:centerColor="#ffffb600"
8. android:centerY="0.75"
9. android:endColor="#ffffcb00"
10. android:angle="270"
11. **/>**
12. **</shape>**
13. **</clip>**
14. **</item>**

为什么shape外面要包一层clip呢，官方文档解释是clipdrawable是可以自我复制的，来看看定义

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<clip**
3. xmlns:android="http://schemas.android.com/apk/res/android"
4. android:drawable="@drawable/drawable\_resource"
5. android:clipOrientation=["horizontal" | "vertical"]
6. android:gravity=["top" | "bottom" | "left" | "right" | "center\_vertical" |
7. "fill\_vertical" | "center\_horizontal" | "fill\_horizontal" |
8. "center" | "fill" | "clip\_vertical" | "clip\_horizontal"] **/>**

android:clipOrientation有两个属性，默认为horizontal

android:gravity有两个属性，默认为left

那我们试试改成vertical和bottom会有什么效果，新建一个progress\_vertical.xml，把源码progress\_horizontal.xml的内容复制过来，这里去掉了secondaryProgress，修改了clip，shape的渐变中心centerY改为centerX

1. **<item** android:id="@android:id/progress"**>**
2. **<clip**
3. android:clipOrientation="vertical"
4. android:gravity = "bottom"**>**
5. **<shape>**
6. **<corners** android:radius="5dip" **/>**
7. **<gradient**
8. android:startColor="#ffffd300"
9. android:centerColor="#ffffb600"
10. android:centerX="0.75"
11. android:endColor="#ffffcb00"
12. android:angle="90"
13. **/>**
14. **</shape>**
15. **</clip>**
16. **</item>**

布局中android:progressDrawable="@drawable/progress\_vertical"

ok，搞定，就是这么简单：

[http://img1.51cto.com/attachment/201205/223030501.jpg](http://img1.51cto.com/attachment/201205/223030501.jpg)

三）弧形bar

这个也许算不上是进度条，用的也不多，最多也就仪表盘用用，不然谁会把进度条整成圆弧的呢。好吧这个可不是改改源码就能搞定的，看代码

1. **public** **class** Arcs **extends** View {
2. **private** Paint mArcPaint;
3. **private** Paint mArcBGPaint;
5. **private** RectF mOval;
6. **private** **float** mSweep = 0;
7. **private** **int** mSpeedMax = 200;
8. **private** **int** mThreshold = 100;
9. **private** **int** mIncSpeedValue = 0;
10. **private** **int** mCurrentSpeedValue = 0;
11. **private** **float** mCenterX;
12. **private** **float** mCenterY;
13. **private** **float** mSpeedArcWidth;
15. **private** **final** **float** SPEED\_VALUE\_INC = 2;
17. ..........
19. }

首先是一堆成员变量，两个Paint用来画圆弧一个前景一个背景，一个RectF圆弧就画在上面，然后是一些控制参数比如sweep圆弧扫过的角度，xy坐标等等

1. mArcPaint = **new** Paint(Paint.ANTI\_ALIAS\_FLAG);
2. mArcPaint.setStyle(Paint.Style.STROKE);
3. mArcPaint.setStrokeWidth(mSpeedArcWidth);
4. //        mPaint.setStrokeCap(Paint.Cap.ROUND);
5. mArcPaint.setColor(0xff81ccd6);
6. BlurMaskFilter mBlur = **new** BlurMaskFilter(8, BlurMaskFilter.Blur.INNER);
7. mArcPaint.setMaskFilter(mBlur);
9. mArcBGPaint = **new** Paint(Paint.ANTI\_ALIAS\_FLAG);
10. mArcBGPaint.setStyle(Paint.Style.STROKE);
11. mArcBGPaint.setStrokeWidth(mSpeedArcWidth+8);
12. mArcBGPaint.setColor(0xff171717);
14. BlurMaskFilter mBGBlur = **new** BlurMaskFilter(8, BlurMaskFilter.Blur.INNER);
15. mArcBGPaint.setMaskFilter(mBGBlur);

设置两个画笔，颜色，宽度，样式等等，BlurMaskFilter笔是边缘模糊效果，有几种，可以自己尝试

1. @Override
2. **protected** **void** onSizeChanged(**int** w, **int** h, **int** ow, **int** oh) {
3. **super**.onSizeChanged(w, h, ow, oh);
4. Log.i("onSizeChanged w", w+"");
5. Log.i("onSizeChanged h", h+"");
6. mCenterX = w \* 0.5f;  // remember the center of the screen
7. mCenterY = h - mSpeedArcWidth;
8. mOval = **new** RectF(mCenterX - mCenterY, mSpeedArcWidth, mCenterX + mCenterY, mCenterY \* 2);
9. }

重写父类View的onSizeChanged，为的是自己根据布局中的大小做居中处理

1. @Override
2. **protected** **void** onDraw(Canvas canvas) {
3. drawSpeed(canvas);
4. calcSpeed();
5. }
7. **private** **void** drawSpeed(Canvas canvas) {
8. canvas.drawArc(mOval, 179, 181, **false**, mArcBGPaint);
10. mSweep = (**float**) mIncSpeedValue / mSpeedMax \* 180;
11. **if** (mIncSpeedValue > mThreshold) {
12. mArcPaint.setColor(0xFFFF0000);
13. }
14. **else** {
15. mArcPaint.setColor(0xFF00B0F0);
16. }
18. canvas.drawArc(mOval, 180, mSweep, **false**, mArcPaint);
19. }
21. **private** **void** calcSpeed() {
22. **if** (mIncSpeedValue < mCurrentSpeedValue) {
23. mIncSpeedValue += SPEED\_VALUE\_INC;
24. **if** (mIncSpeedValue > mCurrentSpeedValue) {
25. mIncSpeedValue = mCurrentSpeedValue;
26. }
27. invalidate();
28. }
29. **else** **if** (mIncSpeedValue > mCurrentSpeedValue) {
30. mIncSpeedValue -= SPEED\_VALUE\_INC;
31. **if** (mIncSpeedValue < mCurrentSpeedValue) {
32. mIncSpeedValue = mCurrentSpeedValue;
33. }
34. invalidate();
35. }
36. }

重写onDraw以便重绘canvas

drawSpeed里面

通过计算mSweep = (float) mIncSpeedValue / mSpeedMax \* 180;

然后canvas.drawArc(mOval, 180, mSweep, false, mArcPaint);

会根据mSweep的变化，画出相应长度的弧度来

根据与阈值的对比，还可以设定不同的 颜色：

if (mIncSpeedValue > mThreshold) {

mArcPaint.setColor(0xFFFF0000);

}

else {

mArcPaint.setColor(0xFF00B0F0);

}

calcSpeed通过一个步进来控制增量或减量，以使弧度自然过渡，减少跳跃

ok，大功告成

[](http://img1.51cto.com/attachment/201205/230319998.jpg)