 SkBitmap  bmp;

     bmp.setConfig(SkBitmap::kARGB\_8888\_Config, rect.Width(), rect.Height());

     bmp.allocPixels();

     SkCanvas canvas(bmp);

     SkPaint paint;

     SkRect rect;

     SkPaint::Style oldStyle;





     //画矩形

 //  rect.setXYWH(100, 100, 200, 200);

 //  paint.setColor(SK\_ColorRED);

 //  paint.setStrokeWidth(1);

 //  oldStyle = paint.getStyle();

 //  //kStroke\_Style，画边框

 //  //kFill\_Style，填充

 //  paint.setStyle(SkPaint::kStroke\_Style);

 //  canvas.drawRect(rect, paint);

 //  paint.setStyle(oldStyle);





     //画线

 //  paint.setStrokeWidth(1);

 //  paint.setColor(SK\_ColorBLACK);

 //  paint.setAntiAlias(true);//抗锯齿

 //  canvas.drawLine(0, 0, 300, 100, paint);





     //画贝塞尔曲线

 //  SkPath path;

 //  SkPoint pt[4];

 //

 //  pt[0].set(100, 100);

 //  pt[1].set(200, 80);

 //  pt[2].set(300, 80);

 //  pt[3].set(400, 100);

 //  path.moveTo(pt[0]);

 //  //二次贝塞尔曲线

 //  //path.quadTo(pt1, pt2);

 //  //三次贝塞尔曲线

 //  path.cubicTo(pt[1], pt[2], pt[3]);

 //  paint.setAntiAlias(true);//抗锯齿

 //  paint.setStyle(SkPaint::kStroke\_Style);

 //  paint.setStrokeWidth(1);

 //  paint.setColor(SK\_ColorBLACK);

 //  canvas.drawPath(path, paint);

 //  //贝塞尔点

 //  paint.setColor(SK\_ColorRED);

 //  canvas.drawCircle(pt[0].x(), pt[0].y(), 4, paint);

 //  canvas.drawCircle(pt[1].x(), pt[1].y(), 4, paint);

 //  canvas.drawCircle(pt[2].x(), pt[2].y(), 4, paint);

 //  canvas.drawCircle(pt[3].x(), pt[3].y(), 4, paint);





     //画椭圆

 //  rect.setXYWH(100, 100, 300, 200);

 //  paint.setColor(SK\_ColorBLUE);

 //  paint.setStyle(SkPaint::kFill\_Style);//填充

 //  paint.setAntiAlias(true);//抗锯齿

 //  canvas.drawOval(rect, paint);



     //渐变

 //  SkPoint pts[2] = {100, 0, 400, 0};

 //  SkColor colors[3] = {SK\_ColorRED, SK\_ColorGREEN, SK\_ColorBLUE};//渐变颜色

 //  SkScalar pos[3] = {0, 0.2, 1.0};//颜色过渡率

 //  int count = SK\_ARRAY\_COUNT(colors);

 //  SkShader\* shader = SkGradientShader::CreateLinear(pts,

 //      colors,

 //      pos,

 //      count,

 //      SkShader::kMirror\_TileMode);

 //  //shader对象控制权交给paint

 //  paint.setShader(shader);

 //  shader-**>**unref();

 //  rect.setXYWH(100, 100, 300, 200);

 //  canvas.drawRect(rect, paint);

 //  paint.setShader(NULL);//释放shader





     //画点集合

 //  SkPoint pts[] = {{100, 100},

 //  {200, 100},

 //  {200, 200},

 //  {100, 200}};

 //  int count = SK\_ARRAY\_COUNT(pts);

 //  SkCanvas::PointMode mode = SkCanvas::kPolygon\_PointMode;

 //  paint.setColor(SK\_ColorBLUE);

 //  paint.setStrokeWidth(1);

 //  canvas.drawPoints(mode,

 //      count,

 //      pts,

 //      paint);





     //画路径

 //  SkPath path;

 //

 //  path.addRect(0, 0, 100, 100);

 //  path.addCircle(100, 100, 50);

 //  paint.setColor(SK\_ColorRED);

 //  //kStroke\_Style，路径轮廓，

 //  //kFill\_Style，填充模式

 //  paint.setStyle(SkPaint::kStroke\_Style);

 //  canvas.drawPath(path, paint);





     //画文字

 //  wstring text = L"hello Skia!";

 //  paint.setColor(SK\_ColorRED);

 //  paint.setTextEncoding(SkPaint::kUTF16\_TextEncoding);//UNICODE编码

 //  paint.setTextSize(30);

 //  paint.setAntiAlias(true);//抗锯齿

 //  paint.setStyle(SkPaint::kFill\_Style);

 //  canvas.drawText(text.c\_str(), text.length() \* 2, 100, 100, paint);//(100, 100)左下点坐标

 //  //得到绘画文字的高度和宽度

 //  SkScalar width = paint.measureText(text.c\_str(), text.length() \* 2);

 //  SkScalar height = paint.getTextSize();

 //  rect.setXYWH(100, 100, width, height);

 //  paint.setStyle(SkPaint::kStroke\_Style);

 //  canvas.drawRect(rect, paint);

 //  paint.setStyle(SkPaint::kFill\_Style);



     //文字旋转

 //  wstring text = L"123456";

 //  SkPath path;

 //  SkPoint pts[2];

 //  pts[0].set(200, 200);

 //  pts[1].set(200, 100);

 //  path.addPoly(pts, 2, false);

 //  paint.setStyle(SkPaint::kStroke\_Style);//轮廓

 //  paint.setColor(SK\_ColorRED);

 //  canvas.drawPath(path, paint);//画路径

 //  paint.setAntiAlias(true);//平滑

 //  paint.setStyle(SkPaint::kFill\_Style);//填充

 //  paint.setTextSize(20);

 //  paint.setTextEncoding(SkPaint::kUTF16\_TextEncoding);//UNICODE编码

 //  paint.setColor(SK\_ColorBLACK);

 //  canvas.drawTextOnPath(text.c\_str(), text.size() \* 2, path, NULL, paint);//画竖直文字



     //路径

 //  SkPath path[2];

 //  SkPoint pts[4];

 //  SkMatrix mx;

 //

 //  pts[0].set(50, 50);

 //  pts[1].set(75, 25);

 //  pts[2].set(125, 25);

 //  pts[3].set(150, 50);

 //  path[0].addPoly(pts, 4, true);

 //  mx.reset();

 //  mx.setTranslateX(100);//X坐标平移100

 //  path[1].addPoly(pts, 4, true);

 //  path[0].addPath(path[1], mx);//添加新路径

 //  canvas.drawPath(path[0], paint);





     //设置剪切区

 //  SkPath pa[2];

 //  SkPoint pts[4];

 //

 //  pts[0].set(100, 100);

 //  pts[1].set(200, 100);

 //  pts[2].set(200, 200);

 //  pts[3].set(100, 200);

 //  pa[0].addPoly(pts, 4, true);

 //  pts[0].set(150, 125);

 //  pts[1].set(250, 125);

 //  pts[2].set(250, 175);

 //  pts[3].set(150, 175);

 //  pa[1].addPoly(pts, 4, true);

 //

 //  paint.setColor(SK\_ColorBLUE);

 //  canvas.save(SkCanvas::kClip\_SaveFlag);//保存剪切区

 //  canvas.clipPath(pa[0], SkRegion::kReplace\_Op);

 //  canvas.clipPath(pa[1], SkRegion::kXOR\_Op);

 //  canvas.drawPath(pa[0], paint);

 //  canvas.drawPath(pa[1], paint);

 //  canvas.restore();//还原剪切区



     //绕中心点旋转矩形

 //  SkPoint ptCenter;



 //  paint.setAntiAlias(true);

 //  paint.setStyle(SkPaint::kStroke\_Style);

 //  paint.setColor(SK\_ColorBLACK);

 //  paint.setStrokeWidth(1.0f);



 //  rect.setXYWH(50, 50, 100, 50);

 //  ptCenter.fX = rect.centerX();

 //  ptCenter.fY = rect.centerY();



 //  canvas.save(SkCanvas::kMatrix\_SaveFlag);

 //  canvas.translate(ptCenter.fX, ptCenter.fY);//平移坐标系到矩形中心点

 //  canvas.rotate(30.0f);

 //  rect.offset(-ptCenter.fX, -ptCenter.fY);//将矩形中心点平移到原点(0,0)

 //  canvas.drawRect(rect, paint);//绘画旋转后的矩形

 //  canvas.restore();



 //  rect.offset(ptCenter.fX, ptCenter.fY);

 //  paint.setColor(SK\_ColorRED);

 //  canvas.drawRect(rect, paint);