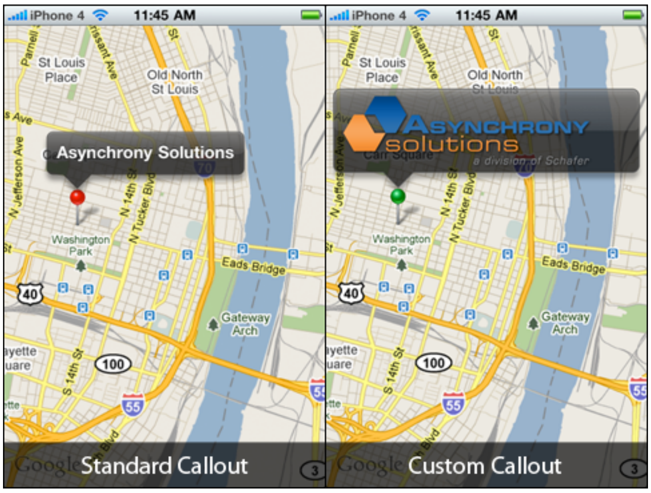
## Building Custom Map Annotation Callouts – Part 1

*by James Rantanen on September 1st, 2010 at 2:50 am*

The iPhone’s Map Annotation Callouts are very useful for displaying small amounts of information when a map pin (annotation) is selected. One problem with the standard callouts present in iOS is the inability to change the height of the callout.

For example, you may want to display a logo or other image that is taller than the default callout. Or you may want to display an address and phone number on separate lines under the title. Both of these scenarios are impossible using the standard iOS callouts. There are many steps to building a good replacement callout with the proper look and behavior, but it can be done.



*Part 1 (explained here) will explain how to build a custom map callout.*

*Part 2 covers adding a button to the custom callout, which is not as simple as it sounds.*

### Put it on the map (and take it off)

For this example we will create two simple map annotations in the view controller – one will display the standard callout and the other will display the custom callout.

To place the “custom callout annotation” on the map we will add the custom annotation when the mapView calls the `mapView:didSelectAnnotationView:` method, and we will remove the callout on the corresponding deselect method, `mapView:didDeselectAnnotationView:`. In `mapView:viewForAnnotation:` we return an instance of our custom MKAnnotationView subclass. Also, we disable the standard callout on the “parent” annotation view, which we will show the custom callout for.

```
- (void)mapView:(MKMapView *)mapView didSelectAnnotationView:(MKAnnotationView *)view {
        if (view.annotation == self.customAnnotation) {
                if (self.calloutAnnotation == nil) {
                        self.calloutAnnotation = [[CalloutMapAnnotation alloc]
                                initWithLatitude:view.annotation.coordinate.latitude
                                andLongitude:view.annotation.coordinate.longitude];
                } else {
                        self.calloutAnnotation.latitude = view.annotation.coordinate.latitude;
                        self.calloutAnnotation.longitude = view.annotation.coordinate.longitude;
                }
                [self.mapView addAnnotation:self.calloutAnnotation];
                self.selectedAnnotationView = view;
        }
}

- (void)mapView:(MKMapView *)mapView didDeselectAnnotationView:(MKAnnotationView *)view {
        if (self.calloutAnnotation && view.annotation == self.customAnnotation) {
                [self.mapView removeAnnotation: self.calloutAnnotation];
        }
}

- (MKAnnotationView *)mapView:(MKMapView *)mapView viewForAnnotation:(id<MKAnnotation>)annotation {
        if (annotation == self.calloutAnnotation) {
                CalloutMapAnnotationView *calloutMapAnnotationView = (CalloutMapAnnotationView *)[self.mapView dequ
                if (!calloutMapAnnotationView) {
                        calloutMapAnnotationView = [[[CalloutMapAnnotationView alloc] initWithAnnotation:annotation
                                reuseIdentifier:@"CalloutAnnotation"] autorelease];
                }
                calloutMapAnnotationView.parentAnnotationView = self.selectedAnnotationView;
                calloutMapAnnotationView.mapView = self.mapView;
                return calloutMapAnnotationView;
        } else if (annotation == self.customAnnotation) {
                MKPinAnnotationView *annotationView = [[[MKPinAnnotationView alloc] initWithAnnotation:annotation
                                reuseIdentifier:@"CustomAnnotation"] autorelease];
                annotationView.canShowCallout = NO;
                annotationView.pinColor = MKPinAnnotationColorGreen;
                return annotationView;
        } else if (annotation == self.normalAnnotation) {
                MKPinAnnotationView *annotationView = [[[MKPinAnnotationView alloc] initWithAnnotation:annotation
                                reuseIdentifier:@"NormalAnnotation"] autorelease];
                annotationView.canShowCallout = YES;
                annotationView.pinColor = MKPinAnnotationColorPurple;
                return annotationView;
        }

        return nil;
}
```

*Note: If building for iOS 3.x you will need to determine annotation selection another way (KVO, notifications, etc.).*

**Draw the callout (in the right place)**

Now that we have the callout annotation placed on the map at the same coordinate as the parent annotation, we need to adjust the width and height of the callout view and adjust the center offset so that the view spans the entire width of the map and sits above the parent annotation. These calculations will be done during `setAnnotation:` because our `contentHeight`, `offsetFromParent`, and `mapView` properties should have been set by then. `setNeedsDisplay` will also be called in `setAnnotation:` so that the callout is redrawn to match up with the annotation.

```
- (void)setAnnotation:(id <MKAnnotation>)annotation {
        [super setAnnotation:annotation];
        [self prepareFrameSize];
        [self prepareOffset];
        [self setNeedsDisplay];
}

- (void)prepareFrameSize {
        CGRect frame = self.frame;
        CGFloat height =        self.contentHeight +
        CalloutMapAnnotationViewContentHeightBuffer +
        CalloutMapAnnotationViewBottomShadowBufferSize -
        self.offsetFromParent.y;

        frame.size = CGSizeMake(self.mapView.frame.size.width, height);
        self.frame = frame;
}

- (void)prepareOffset {
        CGPoint parentOrigin = [self.mapView
                                convertPoint:self.parentAnnotationView.frame.origin
                                fromView:self.parentAnnotationView.superview];

        CGFloat xOffset =       (self.mapView.frame.size.width / 2) -
                                                (parentOrigin.x + self.offsetFromParent.x);

        //Add half our height plus half of the height of the annotation we are tied to so that our bottom lines up
        //Then take into account its offset and the extra space needed for our drop shadow
        CGFloat yOffset = -(self.frame.size.height / 2 +
                                                self.parentAnnotationView.frame.size.height / 2) +
                                                self.offsetFromParent.y +
                                                CalloutMapAnnotationViewBottomShadowBufferSize;

        self.centerOffset = CGPointMake(xOffset, yOffset);
}
```

The shape of the callout bubble is basically a round-rectangle with a triangle that points to the parent annotation. Determining where that point should be is a matter of finding the x-coordinate of the parent relative to it and adding the `offsetFromParent.x` property. Luckily `UIView` contains the handy `convertPoint:fromView:` method to handle the conversion between coordinate systems.

The steps to draw something similar to the standard callout are as follows:

- Create the shape (path) of the callout bubble with the point in the right position to match up with the parent
- Fill the path and add the shadow (adding the shadow here and then restoring the context prevents the shadow from being redrawn with each subsequent step)
- Apply a stroke to the path (more opaque than the fill)
- Create a round rectangle path to appear as the “gloss”

- Fill the gloss path with a gradient
- Convert the glass path to a “stroked path” (this will allow us to apply a gradient to the stroke)
- Apply a gradient (light to transparent) to the stroked path

In code:

```
- (void)drawRect:(CGRect)rect {
        CGFloat stroke = 1.0;
        CGFloat radius = 7.0;
        CGMutablePathRef path = CGPathCreateMutable();
        UIColor *color;
        CGColorSpaceRef space = CGColorSpaceCreateDeviceRGB();
        CGContextRef context = UIGraphicsGetCurrentContext();
        CGFloat parentX = [self relativeParentXPosition];

        //Determine Size
        rect = self.bounds;
        rect.size.width -= stroke + 14;
        rect.size.height -= stroke + CalloutMapAnnotationViewHeightAboveParent - self.offsetFromParent.y + CalloutM
        rect.origin.x += stroke / 2.0 + 7;
        rect.origin.y += stroke / 2.0;

        //Create Path For Callout Bubble
        CGPathMoveToPoint(path, NULL, rect.origin.x, rect.origin.y + radius);
        CGPathAddLineToPoint(path, NULL, rect.origin.x, rect.origin.y + rect.size.height - radius);
        CGPathAddArc(path, NULL, rect.origin.x + radius, rect.origin.y + rect.size.height - radius,
                                 radius, M_PI, M_PI / 2, 1);
        CGPathAddLineToPoint(path, NULL, parentX - 15,
                                                 rect.origin.y + rect.size.height);
        CGPathAddLineToPoint(path, NULL, parentX,
                                                 rect.origin.y + rect.size.height + 15);
        CGPathAddLineToPoint(path, NULL, parentX + 15,
                                                 rect.origin.y + rect.size.height);
        CGPathAddLineToPoint(path, NULL, rect.origin.x + rect.size.width - radius,
                                                 rect.origin.y + rect.size.height);
        CGPathAddArc(path, NULL, rect.origin.x + rect.size.width - radius,
                                 rect.origin.y + rect.size.height - radius, radius, M_PI / 2, 0.0f, 1);
        CGPathAddLineToPoint(path, NULL, rect.origin.x + rect.size.width, rect.origin.y + radius);
        CGPathAddArc(path, NULL, rect.origin.x + rect.size.width - radius, rect.origin.y + radius,
                                 radius, 0.0f, -M_PI / 2, 1);
        CGPathAddLineToPoint(path, NULL, rect.origin.x + radius, rect.origin.y);
        CGPathAddArc(path, NULL, rect.origin.x + radius, rect.origin.y + radius, radius,
                                 -M_PI / 2, M_PI, 1);
        CGPathCloseSubpath(path);

        //Fill Callout Bubble & Add Shadow
        color = [[UIColor blackColor] colorWithAlphaComponent:.6];
        [color setFill];
        CGContextAddPath(context, path);
        CGContextSaveGState(context);
        CGContextSetShadowWithColor(context, CGSizeMake (0, self.yShadowOffset), 6, [UIColor colorWithWhite:0 alpha
        CGContextFillPath(context);
        CGContextRestoreGState(context);

        //Stroke Callout Bubble
        color = [[UIColor darkGrayColor] colorWithAlphaComponent:.9];
        [color setStroke];
        CGContextSetLineWidth(context, stroke);
        CGContextSetLineCap(context, kCGLineCapSquare);
        CGContextAddPath(context, path);
        CGContextStrokePath(context);

        //Determine Size for Gloss
        CGRect glossRect = self.bounds;
        glossRect.size.width = rect.size.width - stroke;
        glossRect.size.height = (rect.size.height - stroke) / 2;
        glossRect.origin.x = rect.origin.x + stroke / 2;
        glossRect.origin.y += rect.origin.y + stroke / 2;

        CGFloat glossTopRadius = radius - stroke / 2;
        CGFloat glossBottomRadius = radius / 1.5;

        //Create Path For Gloss
        CGMutablePathRef glossPath = CGPathCreateMutable();
        CGPathMoveToPoint(glossPath, NULL, glossRect.origin.x, glossRect.origin.y + glossTopRadius);
        CGPathAddLineToPoint(glossPath, NULL, glossRect.origin.x, glossRect.origin.y + glossRect.size.height - glos
        CGPathAddArc(glossPath, NULL, glossRect.origin.x + glossBottomRadius, glossRect.origin.y + glossRect.size.h
                                 glossBottomRadius, M_PI, M_PI / 2, 1);
        CGPathAddLineToPoint(glossPath, NULL, glossRect.origin.x + glossRect.size.width - glossBottomRadius,
                                                 glossRect.origin.y + glossRect.size.height);
        CGPathAddArc(glossPath, NULL, glossRect.origin.x + glossRect.size.width - glossBottomRadius,
                                 glossRect.origin.y + glossRect.size.height - glossBottomRadius, glossBottomRadius,
        CGPathAddLineToPoint(glossPath, NULL, glossRect.origin.x + glossRect.size.width, glossRect.origin.y + gloss
        CGPathAddArc(glossPath, NULL, glossRect.origin.x + glossRect.size.width - glossTopRadius, glossRect.origin.
                                 glossTopRadius, 0.0f, -M_PI / 2, 1);
        CGPathAddLineToPoint(glossPath, NULL, glossRect.origin.x + glossTopRadius, glossRect.origin.y);
        CGPathAddArc(glossPath, NULL, glossRect.origin.x + glossTopRadius, glossRect.origin.y + glossTopRadius, glc
                                 -M_PI / 2, M_PI, 1);
        CGPathCloseSubpath(glossPath);

        //Fill Gloss Path
        CGContextAddPath(context, glossPath);
        CGContextClip(context);
        CGFloat colors[] =
        {
                1, 1, 1, .3,
                1, 1, 1, .1,
        };
        CGFloat locations[] = { 0, 1.0 };
        CGGradientRef gradient = CGGradientCreateWithColorComponents(space, colors, locations, 2);
        CGPoint startPoint = glossRect.origin;
        CGPoint endPoint = CGPointMake(glossRect.origin.x, glossRect.origin.y + glossRect.size.height);
        CGContextDrawLinearGradient(context, gradient, startPoint, endPoint, 0);

        //Gradient Stroke Gloss Path
        CGContextAddPath(context, glossPath);
        CGContextSetLineWidth(context, 2);
        CGContextReplacePathWithStrokedPath(context);
        CGContextClip(context);
        CGFloat colors2[] =
        {
                1, 1, 1, .3,
                1, 1, 1, .1,
                1, 1, 1, .0,
        };
        CGFloat locations2[] = { 0, .1, 1.0 };
        CGGradientRef gradient2 = CGGradientCreateWithColorComponents(space, colors2, locations2, 3);
        CGPoint startPoint2 = glossRect.origin;
        CGPoint endPoint2 = CGPointMake(glossRect.origin.x, glossRect.origin.y + glossRect.size.height);
        CGContextDrawLinearGradient(context, gradient2, startPoint2, endPoint2, 0);

        //Cleanup
        CGPathRelease(path);
        CGPathRelease(glossPath);
        CGColorSpaceRelease(space);
        CGGradientRelease(gradient);
        CGGradientRelease(gradient2);
}

- (CGFloat)yShadowOffset {
        if (!_yShadowOffset) {
                float osVersion = [[[UIDevice currentDevice] systemVersion] floatValue];
                if (osVersion >= 3.2) {
                        _yShadowOffset = 6;
                } else {
                        _yShadowOffset = -6;
                }

        }
        return _yShadowOffset;
}

- (CGFloat)relativeParentXPosition {
        CGPoint parentOrigin = [self.mapView convertPoint:self.parentAnnotationView.frame.origin
                                                                                         fromView:self.parentAnnota
        return parentOrigin.x + self.offsetFromParent.x;
}
```

*Note: in iOS 3.2* `CGContextSetShadowWithColor` *reversed the direction of the y-axis offset, thus requiring the*`yShadowOffset` *method seen above.*

### Let’s Add Some Content

To allow the addition of content we will create a content view as a read-only property, which will allow our consumers to access it. An additional method, `prepareContentFrame` will be added and invoked from `setAnnotation:` to set the content frame.

```
- (void)setAnnotation:(id <MKAnnotation>)annotation {
    [super setAnnotation:annotation];
        [self prepareFrameSize];
        [self prepareOffset];
        [self prepareContentFrame];
    [self setNeedsDisplay];
}

- (void)prepareContentFrame {
        CGRect contentFrame = CGRectMake(self.bounds.origin.x + 10,
                                                                         self.bounds.origin.y + 3,
                                                                         self.bounds.size.width - 20,
                                                                         self.contentHeight);

        self.contentView.frame = contentFrame;
}
```

```
- (UIView *)contentView {
        if (!_contentView) {
                _contentView = [[UIView alloc] init];
                self.contentView.backgroundColor = [UIColor clearColor];
                self.contentView.autoresizingMask = UIViewAutoresizingFlexibleHeight | UIViewAutoresizingFlexibleWi
                [self addSubview:self.contentView];
        }
        return _contentView;
}
```

In our map view controller we will add the following code in `mapView:viewForAnnotation` to place an image in the callout and set the proper content height.

```
                        calloutMapAnnotationView.contentHeight = 78.0f;
                        UIImage *asynchronyLogo = [UIImage imageNamed:@"asynchrony-logo-small.png"];
                        UIImageView *asynchronyLogoView = [[[UIImageView alloc] initWithImage:asynchronyLogo] autor
                        asynchronyLogoView.frame = CGRectMake(5, 2, asynchronyLogoView.frame.size.width, asynchrony
                        [calloutMapAnnotationView.contentView addSubview:asynchronyLogoView];
```

**Animation**

So far the callout looks similar to the native callout, but it is still lacking some of the behavior of the original. The callout needs to animate out from the parent annotation. Also, when the parent annotation is near the edge of the map view, the map should be adjusted to move the parent annotation in from the edge of the view.

The animation would be fairly simple if we could just adjust the frame of the callout view, however that will not scale the contents of the callout. Thus, we must use a CGAffineTransform. Apple has a good introducton to affine transforms. The transform will need to both scale the view and translate the view to make it appear to grow out of the parent annotation. Scaling is simple – a value of 1 is normal size and other values act as a multiplier, so smaller values shrink the view and larger values expand the view. If the parent is off-center on the x-axis the callout needs to be translated to keep the point fixed directly over the parent annotation. Likewise the y-axis must be translated so that it appears that the callout grows upward from parent. We need to hold on to the frame for these calculations because `self.frame` cannot be trusted during the animations. The calculations are done in the following two methods:

```
- (CGFloat)xTransformForScale:(CGFloat)scale {
        CGFloat xDistanceFromCenterToParent = self.endFrame.size.width / 2 - [self relativeParentXPosition];
        return (xDistanceFromCenterToParent * scale) - xDistanceFromCenterToParent;
}

- (CGFloat)yTransformForScale:(CGFloat)scale {
        CGFloat yDistanceFromCenterToParent = (((self.endFrame.size.height) / 2) + self.offsetFromParent.y + Callou
        return yDistanceFromCenterToParent - yDistanceFromCenterToParent * scale;
}
```

There will be three steps to the animation to create the bounce-like effect of the standard callout. We cannot begin the animation with a scale of 0 because a transformation matrix with a scale of 0 cannot be inverted.

1. Grow from very small to slightly larger than the final size
2. Shrink to slightly smaller than the final size
3. Grow to the final size

These three steps will be separate animations chained together using UIView's `setAnimationDidStopSelector:` and `setAnimationDelegate:` methods.

```
- (void)animateIn {
        self.endFrame = self.frame;
        CGFloat scale = 0.001f;
        self.transform = CGAffineTransformMake(scale, 0.0f, 0.0f, scale, [self xTransformForScale:scale], [self yTr
        [UIView beginAnimations:@"animateIn" context:nil];
        [UIView setAnimationCurve:UIViewAnimationCurveEaseOut];
        [UIView setAnimationDuration:0.075];
        [UIView setAnimationDidStopSelector:@selector(animateInStepTwo)];
        [UIView setAnimationDelegate:self];

        scale = 1.1;
        self.transform = CGAffineTransformMake(scale, 0.0f, 0.0f, scale, [self xTransformForScale:scale], [self yTr

        [UIView commitAnimations];
}

- (void)animateInStepTwo {
        [UIView beginAnimations:@"animateInStepTwo" context:nil];
        [UIView setAnimationCurve:UIViewAnimationCurveEaseInOut];
        [UIView setAnimationDuration:0.1];
        [UIView setAnimationDidStopSelector:@selector(animateInStepThree)];
        [UIView setAnimationDelegate:self];

        CGFloat scale = 0.95;
        self.transform = CGAffineTransformMake(scale, 0.0f, 0.0f, scale, [self xTransformForScale:scale], [self yTr

        [UIView commitAnimations];
}

- (void)animateInStepThree {
        [UIView beginAnimations:@"animateInStepThree" context:nil];
        [UIView setAnimationCurve:UIViewAnimationCurveEaseInOut];
        [UIView setAnimationDuration:0.075];

        CGFloat scale = 1.0;
        self.transform = CGAffineTransformMake(scale, 0.0f, 0.0f, scale, [self xTransformForScale:scale], [self yTr

        [UIView commitAnimations];
}
```

### Shifting the Map

When the parent annotation is near the edge of the map, the map needs to be shifted so that the parent annotation and the callout remain a certain distance away from the edge of the view. To do this we need to calculate the distance to the edge of the view, the number of degrees latitude and longitude per pixel, and then set the new center point for the map. This adjustment should be made when `didMoveToSuperview` is called.

```
- (void)adjustMapRegionIfNeeded {
        //Longitude
        CGFloat xPixelShift = 0;
        if ([self relativeParentXPosition] < 38) {
                xPixelShift = 38 - [self relativeParentXPosition];
        } else if ([self relativeParentXPosition] > self.frame.size.width - 38) {
                xPixelShift = (self.frame.size.width - 38) - [self relativeParentXPosition];
        }

        //Latitude
        CGPoint mapViewOriginRelativeToParent = [self.mapView convertPoint:self.mapView.frame.origin toView:self.pa
        CGFloat yPixelShift = 0;
        CGFloat pixelsFromTopOfMapView = -(mapViewOriginRelativeToParent.y + self.frame.size.height - CalloutMapAnn
        CGFloat pixelsFromBottomOfMapView = self.mapView.frame.size.height + mapViewOriginRelativeToParent.y - self
        if (pixelsFromTopOfMapView < 7) {
                yPixelShift = 7 - pixelsFromTopOfMapView;
        } else if (pixelsFromBottomOfMapView < 10) {
                yPixelShift = -(10 - pixelsFromBottomOfMapView);
        }

        //Calculate new center point, if needed
        if (xPixelShift || yPixelShift) {
                CGFloat pixelsPerDegreeLongitude = self.mapView.frame.size.width / self.mapView.region.span.longitu
                CGFloat pixelsPerDegreeLatitude = self.mapView.frame.size.height / self.mapView.region.span.latitud

                CLLocationDegrees longitudinalShift = -(xPixelShift / pixelsPerDegreeLongitude);
                CLLocationDegrees latitudinalShift = yPixelShift / pixelsPerDegreeLatitude;

                CLLocationCoordinate2D newCenterCoordinate = {self.mapView.region.center.latitude + latitudinalShif
                        self.mapView.region.center.longitude + longitudinalShift};

                [self.mapView setCenterCoordinate:newCenterCoordinate animated:YES];

                //fix for now
                self.frame = CGRectMake(self.frame.origin.x - xPixelShift,
                                                                self.frame.origin.y - yPixelShift,
                                                                self.frame.size.width,
                                                                self.frame.size.height);
                //fix for later (after zoom or other action that resets the frame)
                self.centerOffset = CGPointMake(self.centerOffset.x - xPixelShift, self.centerOffset.y);
        }
}
```

### Conclusion

It takes a bit of work to replicate the iOS map annotation callout, but it is worth the effort if you need a larger space for content. You may download the full source code to see a working example.

**James Rantanen**

Website - More Posts

Tags: affine transform, animation, callout, iOS, iPhone, mapkit, mkannotation, mkannotationview, mobile

**23 comments**

1. First, I'm amazed that no-one else has commented – this is extremely useful code, and I thank you for making it available.
   I've implemented it in a project that has multiple instances of the custom callout. It works fine in the 4.1 simulator, and works for the most part on a 3.1.3 iPhone.
   I appreciate your comment that getting select/deselect events in 3.x requires manual notifications. On the 3.1.3 device, I am not getting the accessory tap events – is that also something that needs a manual notification? Any ideas on how to do that?
   Thanks.
   -Mike

   Mike on November 8, 2010

2. I love this custom View. One Problem I want to make this much smaller so that it doesn't cover the entire width of the Map View. Is there an easy way to do this? I tried changing values in your coding only to get some pretty bad results.

   The thing I'm looking to do is make it have a yellow background which I managed and add a Name and two buttons, which I have also done. The width is just big for my liking.

   Any help on reducing the width would be greatly appreciated! Thanks

   Roy Remington on November 9, 2010

3. Oh, Never mind. Disregard my previous comment. I managed to figure it out. Had to change the X Origin and then had it automatically zoom into the annotation when it was selected to resolve the problem.

   Roy Remington on November 9, 2010

4. To Roy Remington, That's exactly the problem I need to solve – could you post some code that shows how you changed the X origin and what you mean by "automatically zoom into the annotation"?
   Thanks!
   -Mike

   Mike on December 13, 2010

5. Your code is the best map callout solution I have ever seen on the net. thanks so much.

   Xiaoyu Chen on January 1, 2011

6. First of all, thank you very much for this great tutorial. I am new to iphone programming, and I would like to add multiple custom annotation view to my map.

   When I try to edit this tutorial to add multiple custom annotation view, only the last annotation become custom annotation and the rest of them become normal annotation.

   How can i add multiple custom annotation view?

   Sun on January 2, 2011

7.

Thanks a lot for this great code.
But I have a problem in it. How can I change the width of the callout?.
I try to do that but I got a strange behavior (the callout moved from its position???)

Any help would be appreciated 🙂

Thanks in advance.
Mohamed

Mohamed Adel on January 4, 2011

8.

Hi, this code has really put my project forward! Thank you so much.
One problem I've encountered is that if I have several annotations, only the first annotations will display a Custom Callout. How do you do that so if the user clicks on a second annotation following the first one, the custom callout will also be displayed?

Again, thank you so much for your help, this is awesome!

Martin on January 9, 2011

9.

Great mapping code by the way. The best I've seen on the net and in some of the books I have. I encountered a similar problem as Martin on how to have multiple annotations with the CustomCallout. Currently it only works for the first annotation and not the others. Any help would be appreciated. Thanks.

Pettis Hartfield on February 4, 2011

10.

Caution – didSelectAnnotationView is only available from iOS 4.0 onwards. Ideally there would be another method for selecting the annotation, probably by handling the touches in the view manually. Still researching…

Gavin on February 18, 2011

11.

Great work on this. Pretty much everyone who asks about this on Stack Overflow gets sent here.

I have a question… why has Apple made this so hard to do?

How about setCalloutHeight: 100… and you're done?

Jonathan on February 27, 2011

12.

Hi,
Its really nice code.But I have one problem. I have added button in Calloutmap annotation view but button seems less clickable.Button click method not called. It's called hardly when I keep pressing upto 2 or 3 seconds. Any help will be appreciated.
Thnaks

aArmaan on March 4, 2011

13. [...] Part 1 showed how to build a custom map callout that provides more content flexibility than the native callout, but maintains the expected look and behavior. In part 2 we will add a very common element of the map interface into our custom callout – the accessory button. At first glance this seems simple: just add a button to the callout. However, MapKit intercepts touch events and causes undesired callout behavior. The code used to add an accessory button is also applicable to any other button(s) or responders you may want to add to a callout, giving you the flexibility to do what you feel is best for your users. [...]

Building Custom Map Annotation Callouts – Part 2 | Advanced App Development on March 25, 2011

14.

Thanks for the code. Looks nice and the description is very thorough.
I need to develop an annotation-like bubbles to use without MapKit. I think I will use your ideas for this as well as some of the ideas from replies in this post – http://stackoverflow.com/questions/1619245/using-mkannotationview-without-mkmapview

Unfortunately the code provided by one of the repliers is not quite usable and doesn't have a triangle below the view

Kostiantyn Sokolinskyi on April 1, 2011

15.

This is a good solution, but I can't help but feel it's an overkill for me unfortunately, I am only trying to replace the grey callout graphic which is drawn to my custom graphic I've made in Photoshop.

I tried this with your code but had no success, I then tried just overriding drawRect of MKAnnotationView for myself, but no luck with the name and subtitle of the annotation.

If you could suggest a way forward, all I really need to do is replace the graphic, maybe reduce the height slightly too… (to match my graphic)

Daniel Phillips on April 1, 2011

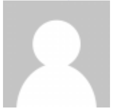
16.

James,
Can you please commit to github?

I think you need these properties to be assign (weak references) not retain in CalloutMapAnnotationView.

```
@property (nonatomic, retain) MKAnnotationView *parentAnnotationView;
@property (nonatomic, retain) MKMapView *mapView;
```

John Pope on April 11, 2011

17. 

This code is fabulous, thank you James,

But, did anyone managed to change the width of the callout?

If y change the width in – (void)prepareFrameSize :

```
frame.size = CGSizeMake(285, height);
instead of:
frame.size = CGSizeMake(self.mapView.frame.size.width, height);
```

… the callout is placed wrong. I can solve this by changing in – (void)prepareOffset :

```
CGFloat xOffset = 0;
instead of:
CGFloat xOffset = (self.mapView.frame.size.width / 2) –
(parentOrigin.x + self.offsetFromParent.x);
```

But then the triangle under the bubble places wrong.

Can anyone help me.

victor on May 27, 2011

18.

Great tutorial, works like a charm, tks!!!

i was playing a little bit and i've tried with a moving annotation but there is a little issue, the callout doesn't stick to the annotation, do you have any hints for implementing this behavior?

kocisky on May 28, 2011

19.

I'm speechless at such an elegant solution. I was freaking out about wether I would be able to handle the code because it's a little out of my league but it turns out implementation was a piece of cake. Still smoothing out some minor glitches but overall a great piece of code to work with. Thanks!

Max on June 4, 2011

20.

Thanks, this helps a lot! A perplexing bug I have found is if you initially zoom in on the map, then pan the custom pin to the very edge and click on it — it will correctly animate away from the edge but the callout will not be shown. (If you then pan the map some more it will usually pop up).

Shawn on June 10, 2011

21. [...] I, along with other SO users, are using this awesome solution to create a custom callout annotation: http://blog.asolutions.com/2010/09/building-custom-map-annotation-callouts-part-1/ [...]

    Can't adjust coordinates on custom callout annotation view | taking a bite into Apple on June 17, 2011

22. [...] I, along with other SO users, are using this awesome solution to create a custom callout annotation: http://blog.asolutions.com/2010/09/building-custom-map-annotation-callouts-part-1/ [...]

    MKAnnotationView – Lock custom annotation view to pin on location updates | taking a bite into Apple on June 18, 2011

23. This is fantastic, and exactly what I needed for a project I'm working on. The only difference between my project and this example is that CoreLocation updates are constantly changing the coordinates of the pin, so the pin annotation moves around. I can't figure out how to get the callout annotation to "lock" to the pin, and move around with it onscreen. Any ideas? http://stackoverflow.com/q/6392931/607876

    kevin on June 22, 2011

**This Post is Open for Discussion**

Name

Mail (never published)

Website

Your Comment

September 2010

| M | T | W | T | F | S | S |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| 27 | 28 | 29 | 30 | | | |

« Aug Oct »

**Categories**

- Agile
- Asynchrony
- Change
- Design
- Development
- Visual Thinking

**Archives**

- June 2011
- May 2011
- April 2011
- March 2011
- February 2011
- January 2011
- December 2010
- November 2010
- October 2010
- September 2010
- August 2010
- July 2010
- June 2010
- May 2010

**Tags**

acronyms agile android Apple buckminster callout change changeviz CMMI comics continuous integration CRUD cucumber design dev eclipse hudson interface iOS iPad iPhone Kanban liferay mapkit metrics mkannotation mkannotationview mobile p2 pairing photoshop project management rcp retrospectives risk management Security selenium testing tutorial UCD ui urban design usability video vizthink

**Interesting Links**

- Rogue Project Leader

**Asynchrony Twitter Feed**

- asynchrony: New Blog Post: Going Faster With a Back-Seat (Web) Driver: Slow, long running test suites can be almost as dange... http://bit.ly/j1zgKv June 17, 2011
- asynchrony: Are you a whole team? Read the article by Asynchrony's Lori Cross and alumni @mattphilip and find out! http://bit.ly/izaEiW June 9, 2011
- asynchrony: New Blog Post: Today I tried TotalFinder and found a finer Finder: For the most part, OS X's Finder app looks ni... http://bit.ly/iZUECK June 1, 2011
- asynchrony: New Blog Post: ACMP ChangeViz Presentation: This is the presentation Bill Howard and I gave at the 2011 Associat... http://bit.ly/jnIiOJ May 9, 2011
- asynchrony: New Blog Post: Sketchnotes as Collaborative Mashup: The images above are my sketchnotes from the 2011 Associatio... http://bit.ly/mbMUxF May 6, 2011

Come visit us sometime at:
*Our Main Office* **1701 Washington Avenue**
**Saint Louis, Missouri 63103**
**314 / 678-2200**
*Our Scott Air Force Base Office* **703 Seibert Road, Suite 3**
**Scott AFB, IL 62225**
**618 / 212-4580**

Check out our parent site: **asolutions.com**

**Asynchrony Blog Disclaimer**