

[Core Graphics / CGPath](#)

Class

# CGPath

An immutable graphics path: a mathematical description of shapes or lines to be drawn in a graphics context.

iOS | iPadOS | Mac Catalyst | macOS | tvOS | visionOS | watchOS

```
class CGPath
```

## Overview

Neither `CGPath` nor `CGMutablePath` define functions to draw a path. To draw a Core Graphics path to a graphics context, you add the path to the graphics context by calling `addPath(_:_:)` and then call one of the context's drawing functions—see [CGContext](#).

Each figure in the graphics path is constructed with a connected set of lines and Bézier curves, called a *subpath*. A subpath has an ordered set of *path elements* that represent single steps in the construction of the subpath. (For example, a line segment from one corner of a rectangle to another corner is a path element. Every subpath includes a *starting point*, which is the first point in the subpath. The path also maintains a *current point*, which is the last point in the last subpath.

## Topics

### Creating Graphics Paths

`init(rect: CGRect, transform: UnsafePointer<CGAffineTransform>?)`

Create an immutable path of a rectangle.

```
init(ellipseIn: CGRect, transform: UnsafePointer<CGAffineTransform>?)  
Create an immutable path of an ellipse.
```

```
init(roundedRect: CGRect, cornerWidth: CGFloat, cornerHeight: CGFloat,  
transform: UnsafePointer<CGAffineTransform>?)  
Create an immutable path of a rounded rectangle.
```

## Copying a Graphics Path

```
func copy() -> CGPath?  
Creates an immutable copy of a graphics path.
```

```
func copy(using: UnsafePointer<CGAffineTransform>?) -> CGPath?  
Creates an immutable copy of a graphics path transformed by a transformation matrix.
```

```
func copy(dashingWithPhase: CGFloat, lengths: [CGFloat], transform:  
CGAffineTransform) -> CGPath
```

Returns a new path equivalent to the results of drawing the path with a dashed stroke.

```
func copy strokingWithWidth: CGFloat, lineCap: CGLineCap, lineJoin:  
CGLineJoin, miterLimit: CGFloat, transform: CGAffineTransform) ->  
CGPath
```

Returns a new path equivalent to the results of drawing the path with a solid stroke.

```
func mutableCopy() -> CGMutablePath?  
Creates a mutable copy of an existing graphics path.
```

```
func mutableCopy(using: UnsafePointer<CGAffineTransform>?) -> CGMutable  
Path?
```

Creates a mutable copy of a graphics path transformed by a transformation matrix.

## Examining a Graphics Path

```
var boundingBox: CGRect  
Returns the bounding box containing all points in a graphics path.
```

```
var boundingBoxOfPath: CGRect  
Returns the bounding box of a graphics path.
```

```
var currentPoint: CGPoint  
Returns the current point in a graphics path.
```

```
func contains(CGPoint, using: CGPathFillRule, transform: CGAffineTransform) -> Bool
```

Returns whether the specified point is interior to the path.

```
var isEmpty: Bool
```

Indicates whether or not a graphics path is empty.

```
func isRect(UnsafeMutablePointer<CGRect>?) -> Bool
```

Indicates whether or not a graphics path represents a rectangle.

## Applying a Function to the Elements of a Path

```
func apply(info: UnsafeMutableRawPointer?, function: CGPathApplierFunction)
```

For each element in a graphics path, calls a custom applier function.

```
typealias CGPathApplierFunction
```

Defines a callback function that can view an element in a graphics path.

```
struct CGPathElement
```

A data structure that provides information about a path element.

```
enum CGPathElementType
```

The type of element found in a path.

## Working with Core Foundation Types

```
class var typeID: CFTTypeID
```

Returns the Core Foundation type identifier for Core Graphics paths.

## Instance Methods

```
func applyWithBlock((UnsafePointer<CGPathElement>) -> Void)
```

```
func componentsSeparated(using: CGPathFillRule) -> [CGPath]
```

```
func flattened(threshold: CGFloat) -> CGPath
```

```
func intersection(CGPath, using: CGPathFillRule) -> CGPath
```

```
func intersects(CGPath, using: CGPathFillRule) -> Bool
```

```
func lineIntersection(CGPath, using: CGPathFillRule) -> CGPath
```

```
func lineSubtracting(CGPath, using: CGPathFillRule) -> CGPath
func normalized(using: CGPathFillRule) -> CGPath
func subtracting(CGPath, using: CGPathFillRule) -> CGPath
func symmetricDifference(CGPath, using: CGPathFillRule) -> CGPath
func union(CGPath, using: CGPathFillRule) -> CGPath
```

---

## Relationships

### Inherited By

CGMutablePath

### Conforms To

Equatable, Hashable

---

## See Also

### Related Documentation

Quartz 2D Programming Guide

## 2D Drawing

class CGContext

A Quartz 2D drawing environment.

class CGImage

A bitmap image or image mask.

class CGMutablePath

A mutable graphics path: a mathematical description of shapes or lines to be drawn in a graphics context.

```
class CGLayer
```

An offscreen context for reusing content drawn with Core Graphics.