

---

# HIShape Reference



2004-06-28



Apple Inc.  
© 2004 Apple Computer, Inc.  
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.  
1 Infinite Loop  
Cupertino, CA 95014  
408-996-1010

Apple, the Apple logo, Carbon, Mac, Mac OS, Quartz, and QuickDraw are trademarks of Apple Inc., registered in the United States and other countries.

Simultaneously published in the United States and Canada.

**Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.**

**IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY**

**DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.**

**THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.**

**Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.**

# Contents

---

## **HIShape Reference 5**

Overview	5
Functions by Task	6
Creating Immutable Shapes	6
Creating Mutable Shapes	6
Querying Shapes	6
Manipulating Shapes	7
Functions	7
HIShapeContainsPoint	7
HIShapeCreateCopy	8
HIShapeCreateDifference	8
HIShapeCreateEmpty	8
HIShapeCreateIntersection	9
HIShapeCreateMutable	9
HIShapeCreateMutableCopy	10
HIShapeCreateUnion	10
HIShapeCreateWithQDRgn	10
HIShapeCreateWithRect	11
HIShapeDifference	11
HIShapeGetAsQDRgn	12
HIShapeGetBounds	13
HIShapeGetTypeID	13
HIShapeIntersect	13
HIShapeIntersectsRect	14
HIShapeIsEmpty	14
HIShapeIsRectangular	15
HIShapeOffset	15
HIShapeReplacePathInCGContext	16
HIShapeSetEmpty	16
HIShapeSetQDClip	17
HIShapeUnion	17
Data Types	18
HIShapeRef	18
HIMutableShapeRef	18

---

## **Document Revision History 19**

---

## **Index 21**

---



# HIShape Reference

---

<b>Framework:</b>	Carbon/Carbon.h
<b>Declared in</b>	HIShape.h

## Overview

HIShape is an abstract shape object that replaces the old QuickDraw region handle. HIShape objects are the preferred way to describe regions in HView views that use Quartz.

One advantage of HIShape objects is that they can be mutable or immutable. Using immutable (that is, fixed) shapes improves performance because copying it simply requires incrementing the reference count, avoiding the handle-to-handle copies required with the older region handles.

Currently, HIShape objects cannot handle floating point–based shape descriptions. Therefore, any coordinates you specify when defining shapes must end on integer boundaries. The purpose of the shape often dictates whether rounding fractional values up or down will give you the best results. For example, shapes that describe a maximum allowable area, such as the structure region of a window, should be rounded up (or outward) to the nearest integer. Shapes describing a minimum allowable area, such as the opaque region, should be rounded down (or inward). By default, any fractional coordinate values are truncated.

Typical shapes are rectangular, but you can build more complex shapes by combining them with other shapes using the union, intersection, or difference functions. The Quartz-savvy HTheme APIs also let you obtain shapes for various standard control elements.

In most cases, applications will obtain or set shapes in response to a Carbon event using `GetEventParameter` or `SetEventParameter` with the `typeHIShapeRef` parameter type.

The following functions also take HIShape references as parameters:

- `HViewNewTrackingArea` and `HViewChangeTrackingArea`, which let you set or change an HIShape-based tracking area in a view (comparable to window tracking regions).
- `HViewSetNeedsDisplayInShape`, which is the HIShape-compatible version of `HViewSetNeedsDisplayInRegion`.

Except where noted, HIShape functions return only the result codes `noErr` or `paramErr`.

HIShape objects are available in Mac OS X version 10.2 and later. However, many of the functions that take HIShape parameters are available only in Mac OS X version 10.4 and later.

## Functions by Task

### Creating Immutable Shapes

[HIShapeCreateWithQDRgn](#) (page 10)

Creates an immutable shape from a QuickDraw region.

[HIShapeCreateWithRect](#) (page 11)

Creates an immutable shape from a Quartz rectangle.

[HIShapeCreateEmpty](#) (page 8)

Creates an empty immutable shape.

[HIShapeCreateCopy](#) (page 8)

Creates an immutable copy of a mutable or immutable shape.

[HIShapeCreateIntersection](#) (page 9)

Creates an immutable shape that is the intersection of two shapes.

[HIShapeCreateDifference](#) (page 8)

Creates an immutable shape that is the difference of two shapes.

[HIShapeCreateUnion](#) (page 10)

Creates an immutable shape that is the union of two shapes.

### Creating Mutable Shapes

[HIShapeCreateMutable](#) (page 9)

Creates an empty mutable shape.

[HIShapeCreateMutableCopy](#) (page 10)

Creates a mutable copy of an existing shape.

[HIShapeSetEmpty](#) (page 16)

Sets the specified mutable shape to be empty.

[HIShapeIntersect](#) (page 13)

Sets a shape to be the intersection of two other shapes.

[HIShapeDifference](#) (page 11)

Sets a shape to be the difference of two other shapes.

[HIShapeUnion](#) (page 17)

Sets a shape to be the union of two other shapes.

[HIShapeOffset](#) (page 15)

Moves a shape by a specified offset.

### Querying Shapes

[HIShapeIsEmpty](#) (page 14)

Determines whether a given shape is empty.

[HIShapeIsRectangular](#) (page 15)

Determines whether a given shape is rectangular.

[HIShapeContainsPoint](#) (page 7)

Determines whether a shape contains the given point.

## Manipulating Shapes

[HIShapeGetTypeID](#) (page 13)

Obtains the Core Foundation type ID for the HIShape class.

[HIShapeGetBounds](#) (page 13)

Obtains the bounding rectangle for the given shape.

[HIShapeIntersectsRect](#) (page 14)

Determines whether a given shape intersects a given rectangle.

[HIShapeGetAsQDRgn](#) (page 12)

Creates a QuickDraw region with the same shape as the specified HIShape.

[HIShapeReplacePathInCGContext](#) (page 16)

Make the current path in a graphics context match a shape.

[HIShapeSetQDClip](#) (page 17)

Sets the clip region in the current QuickDraw port to be the specified shape.

## Functions

### HIShapeContainsPoint

Determines whether a shape contains the given point.

```
Boolean HIShapeContainsPoint (
    HIShapeRef inShape,
    const CGPoint *inPoint
);
```

#### Parameters

*inShape*

The shape to test.

*inPoint*

The point to test.

#### Return Value

True if the shape contains the point, False otherwise.

#### Discussion

This function is comparable to the old QuickDraw function `PtInRgn`.

#### Availability

Available in Mac OS X v10.2 and later.

#### Declared In

HIShape.h

## **HIShapeCreateCopy**

Creates an immutable copy of a mutable or immutable shape.

```
HIShapeRef HIShapeCreateCopy (  
    HIShapeRef inShape  
);
```

### **Parameters**

*inShape*

The shape you want to copy.

### **Return Value**

The newly-created copy.

### **Discussion**

Copying an immutable shape simply increments its reference count. Copying a mutable shape actually duplicates it in memory.

### **Availability**

Available in Mac OS X v10.2 and later.

### **Declared In**

HIShape.h

## **HIShapeCreateDifference**

Creates an immutable shape that is the difference of two shapes.

```
HIShapeRef HIShapeCreateDifference (  
    HIShapeRef inShape1,  
    HIShapeRef inShape2  
);
```

### **Parameters**

*inShape1*

The first shape.

*inShape2*

The shape to subtract from the first shape.

### **Return Value**

The newly-created difference shape.

### **Availability**

Available in Mac OS X v10.2 and later.

### **Declared In**

HIShape.h

## **HIShapeCreateEmpty**

Creates an empty immutable shape.

```
HIShapeRef HIShapeCreateEmpty (  
    void  
);
```

**Return Value**

The newly-created empty shape.

**Availability**

Available in Mac OS X v10.4 and later.

**Declared In**

HIShape.h

## HIShapeCreateIntersection

Creates an immutable shape that is the intersection of two shapes.

```
HIShapeRef HIShapeCreateIntersection (  
    HIShapeRef inShape1,  
    HIShapeRef inShape2  
);
```

**Parameters**

*inShape1*

An existing shape.

*inShape2*

Another existing shape.

**Return Value**

The newly-created intersection shape.

**Discussion**

If the two shapes do not intersect, the shape returned is empty.

**Availability**

Available in Mac OS X v10.2 and later.

**Declared In**

HIShape.h

## HIShapeCreateMutable

Creates an empty mutable shape.

```
HIMutableShapeRef HIShapeCreateMutable (  
    void  
);
```

**Return Value**

An empty, mutable shape.

**Availability**

Available in Mac OS X v10.2 and later.

**Declared In**

HIShape.h

### **HIShapeCreateMutableCopy**

Creates a mutable copy of an existing shape.

```
HIMutableShapeRef HIShapeCreateMutableCopy (  
    HIShapeRef inOrig  
);
```

**Parameters**

*inOrig*

The shape to copy.

**Return Value**

The newly-created copy.

**Availability**

Available in Mac OS X v10.2 and later.

**Declared In**

HIShape.h

### **HIShapeCreateUnion**

Creates an immutable shape that is the union of two shapes.

```
HIShapeRef HIShapeCreateUnion (  
    HIShapeRef inShape1,  
    HIShapeRef inShape2  
);
```

**Parameters**

*inShape1*

An existing shape.

*inShape2*

Another existing shape.

**Return Value**

The newly-created union shape.

**Availability**

Available in Mac OS X v10.2 and later.

**Declared In**

HIShape.h

### **HIShapeCreateWithQDRgn**

Creates an immutable shape from a QuickDraw region.

```
HIShapeRef HIShapeCreateWithQDRgn (  
    RgnHandle inRgn  
);
```

**Parameters**

*inRgn*

The region to convert to an immutable shape.

**Return Value**

The newly created shape.

**Discussion**

You can use this function to convert handle-based regions to HIShape objects. However, you should be judicious about how often you convert, as this function does require memory allocation.

**Availability**

Available in Mac OS X v10.2 and later.

**Declared In**

HIShape.h

## HIShapeCreateWithRect

Creates an immutable shape from a Quartz rectangle.

```
HIShapeRef HIShapeCreateWithRect (  
    const CGRect *inRect  
);
```

**Parameters**

*inRect*

The Quartz-based rectangle to convert.

**Return Value**

The newly-created shape.

**Discussion**

This function is comparable to the old QuickDraw function `RectRgn`.

**Availability**

Available in Mac OS X v10.2 and later.

**Declared In**

HIShape.h

## HIShapeDifference

Sets a shape to be the difference of two other shapes.

```
OSStatus HIShapeDifference (  
    HIShapeRef inShape1,  
    HIShapeRef inShape2,  
    HIMutableShapeRef outResult  
);
```

**Parameters**

*inShape1*

The first shape.

*inShape2*

The shape to subtract from the first.

*outResult*

The shape to hold the difference of the two shapes. This parameter can be one of the source shapes.

**Return Value**

A result code.

**Discussion**

This function is comparable to the old QuickDraw function `DiffRgn`.

**Availability**

Available in Mac OS X v10.2 and later.

**Declared In**

HIShape.h

## HIShapeGetAsQDRgn

Creates a QuickDraw region with the same shape as the specified HIShape.

```
OSStatus HIShapeGetAsQDRgn (  
    HIShapeRef inShape,  
    RgnHandle outRgn  
);
```

**Parameters**

*inShape*

The shape to convert.

*outRgn*

A valid region handle. You must obtain this handle by calling `NewRgn` before calling this function.

**Return Value**

A result code.

**Discussion**

You can use this function to create a handle-based region from an HIShape object. This conversion requires memory allocation, so you should convert only when necessary.

**Availability**

Available in Mac OS X v10.2 and later.

**Declared In**

HIShape.h

## HIShapeGetBounds

Obtains the bounding rectangle for the given shape.

```
CGRect * HIShapeGetBounds (
    HIShapeRef inShape,
    CGRect *outRect
);
```

### Parameters

*inShape*

The shape whose bounds you want to obtain.

*inRect*

A pointer to the `HIRect` structure you want filled with the shape bounds.

### Return Value

A pointer to the rectangle you passed in the `inRect` parameter, now set to the shape's bounding rectangle.

### Discussion

The function result is a pointer to the `HIRect` structure you passed in the `inRect` parameter.

### Availability

Available in Mac OS X v10.2 and later.

### Declared In

`HIShape.h`

## HIShapeGetTypeID

Obtains the Core Foundation type ID for the `HIShape` class.

```
CTypeID HIShapeGetTypeID (
    void
);
```

### Return Value

The type ID for the `HIShape` class.

### Availability

Available in Mac OS X v10.2 and later.

### Declared In

`HIShape.h`

## HIShapeIntersect

Sets a shape to be the intersection of two other shapes.

```
OSStatus HIShapeIntersect (  
    HISHapeRef inShape1,  
    HISHapeRef inShape2,  
    HIMutableShapeRef outResult  
);
```

**Parameters**

*inShape1*

The first shape.

*inShape2*

The second shape.

*outResult*

The shape to hold the intersection of the two shapes. This parameter can be one of the source shapes.

**Return Value**

A result code.

**Discussion**

This function is comparable to the old QuickDraw function `SectRgn`.

**Availability**

Available in Mac OS X v10.2 and later.

**Declared In**

HIShape.h

## HIShapeIntersectsRect

Determines whether a given shape intersects a given rectangle.

```
Boolean HIShapeIntersectsRect (  
    HISHapeRef inShape,  
    const CGRect *inRect  
);
```

**Parameters**

*inShape*

The shape to test.

*inRect*

The rectangle to test against.

**Return Value**

Returns `True` if the area of the shape intersects the rectangle, `False` otherwise.

**Availability**

Available in Mac OS X v10.4 and later.

**Declared In**

HIShape.h

## HIShapesEmpty

Determines whether a given shape is empty.

```
Boolean HIShapeIsEmpty (  
    HIShapeRef inShape  
);
```

**Parameters**

*inShape*

The shape to test.

**Return Value**

True if the shape's area is empty, False otherwise.

**Discussion**

This function is comparable to the old QuickDraw function EmptyRgn.

**Availability**

Available in Mac OS X v10.2 and later.

**Declared In**

HIShape.h

## HIShapeIsRectangular

Determines whether a given shape is rectangular.

```
Boolean HIShapeIsRectangular (  
    HIShapeRef inShape  
);
```

**Parameters**

*inShape*

The shape to test.

**Return Value**

True if the shape's area is rectangular, False otherwise.

**Availability**

Available in Mac OS X v10.2 and later.

**Declared In**

HIShape.h

## HIShapeOffset

Moves a shape by a specified offset.

```
OSStatus HIShapeOffset (  
    HIMutableShapeRef inShape,  
    CGFloat inDX,  
    CGFloat inDY  
);
```

**Parameters**

*inShape*

The shape to offset.

*inDx*

The desired x-coordinate offset.

*inDy*

The desired y-coordinate offset.

**Return Value**

A result code.

**Discussion**

This function is comparable to the old QuickDraw function `OffsetRgn`.

**Availability**

Available in Mac OS X v10.2 and later.

**Declared In**

`HIShape.h`

## **HIShapeReplacePathInCGContext**

Make the current path in a graphics context match a shape.

```
OSStatus HIShapeReplacePathInCGContext (  
    HIShapeRef inShape,  
    CGContextRef inContext  
);
```

**Parameters**

*inShape*

The shape to apply to the path.

*inContext*

The context to apply the shape to.

**Return Value**

A result code.

**Availability**

Available in Mac OS X v10.2 and later.

**Declared In**

`HIShape.h`

## **HIShapeSetEmpty**

Sets the specified mutable shape to be empty.

```
OSStatus HIShapeSetEmpty (  
    HIMutableShapeRef inShape  
);
```

**Parameters**

*inShape*

The shape to empty.

**Return Value**

A result code.

**Discussion**

This function is comparable to the old QuickDraw function `SetEmptyRgn`.

**Availability**

Available in Mac OS X v10.2 and later.

**Declared In**

`HIShape.h`

## **HIShapeSetQDClip**

Sets the clip region in the current QuickDraw port to be the specified shape.

```
OSStatus HIShapeSetQDClip (  
    HIShapeRef inShape,  
    CGrafPtr inPort  
);
```

**Parameters**

*inShape*

The shape to apply to the clip region.

*inPort*

The clip region to apply the shape to.

**Return Value**

A result code. If `HIShapeSetQDClip` cannot allocate a QuickDraw region, it returns `memFullErr`.

**Availability**

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

**Declared In**

`HIShape.h`

## **HIShapeUnion**

Sets a shape to be the union of two other shapes.

```
OSStatus HIShapeUnion (  
    HIShapeRef inShape1,  
    HIShapeRef inShape2,  
    HIMutableShapeRef outResult  
);
```

**Parameters**

*inShape1*

An existing shape.

*inShape2*

Another existing shape.

*outResult*

The shape to hold the union of the two shapes. This parameter can be one of the source shapes.

**Return Value**

A result code.

**Discussion**

This function is comparable to the old QuickDraw function `UnionRgn`.

**Availability**

Available in Mac OS X v10.2 and later.

**Declared In**

`HIShape.h`

## Data Types

### **HIShapeRef**

Defines a shape object.

```
typedef const struct __HIShape *HIShapeRef;
```

**Discussion**

The structure pointed to by this reference is opaque.

**Availability**

Available in Mac OS X v10.2 and later.

**Declared In**

`HIShape.h`

### **HIMutableShapeRef**

Defines a mutable shape object.

```
typedef struct __HIShape *HIMutableShapeRef;
```

**Discussion**

The structure pointed to by this reference is opaque.

**Availability**

Available in Mac OS X v10.2 and later.

**Declared In**

`HIShape.h`

# Document Revision History

---

This table describes the changes to *HIShape Reference*.

Date	Notes
2004-06-28	First version of this document.

**REVISION HISTORY**

Document Revision History

# Index

---

## H

---

`HImmutableShapeRef` **data type** 18  
`HShapeContainsPoint` **function** 7  
`HShapeCreateCopy` **function** 8  
`HShapeCreateDifference` **function** 8  
`HShapeCreateEmpty` **function** 8  
`HShapeCreateIntersection` **function** 9  
`HShapeCreateMutable` **function** 9  
`HShapeCreateMutableCopy` **function** 10  
`HShapeCreateUnion` **function** 10  
`HShapeCreateWithQDRgn` **function** 10  
`HShapeCreateWithRect` **function** 11  
`HShapeDifference` **function** 11  
`HShapeGetAsQDRgn` **function** 12  
`HShapeGetBounds` **function** 13  
`HShapeGetTypeID` **function** 13  
`HShapeIntersect` **function** 13  
`HShapeIntersectsRect` **function** 14  
`HShapeIsEmpty` **function** 14  
`HShapeIsRectangular` **function** 15  
`HShapeOffset` **function** 15  
`HShapeRef` **data type** 18  
`HShapeReplacePathInCGContext` **function** 16  
`HShapeSetEmpty` **function** 16  
`HShapeSetQDClip` **function** 17  
`HShapeUnion` **function** 17