# HIGeometry Reference

**Carbon > Human Interface Toolbox**

**2007-06-11**

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



Simultaneously published in the United States and Canada.

# Contents

# HIGeometry Reference

**Framework:** Carbon/Carbon.h

**Declared in** HIGeometry.h

## Overview

HIGeometry is a Quartz-compatible API for describing and manipulating basic geometric objects such as points, rectangles, and sizes. HIGeometry expresses all coordinates using floating-point numbers. This API provides functions to convert an object's coordinates into a different coordinate space. These functions support resolution independence mode drawing by taking into account the scale factor of your application's user interface.

## Functions by Task

### Getting the Scale Factor

`HIGetScaleFactor` (page 6)
Returns the scale factor of an application's user interface.

### Converting Coordinates

These functions allow conversion between global pixel coordinates, global virtual (72-dpi) coordinates, window coordinates, and view coordinates.

`HIPointConvert` (page 6)
Converts a point from one coordinate space to another.

`HIRectConvert` (page 7)
Converts a rectangle from one coordinate space to another.

`HISizeConvert` (page 8)
Converts a size structure from one coordinate space to another.

# Functions

## HIGetScaleFactor

Returns the scale factor of an application's user interface.

```
float HIGetScaleFactor (
   void
);
```

**Return Value**
A positive number that represents the scale factor of your application.

**Discussion**
The scale factor of your application's user interface is the ratio between device space units and user space units. For example, if the scale factor is 2.0 and you draw a horizontal line with a length of 10 units, the graphics system will display a line with a length of 20 pixels.

The scale factor is initialized when your application launches. In Mac OS X v10.4, the scale factor is always 1.0 (unless you use the Quartz Debug application to change it). In future versions of Mac OS X, the scale factor may vary based on user preference or the resolution of the main display.

Application frameworks such as Cocoa and Carbon use the scale factor to adjust the size of your application's user interface so that it has the appropriately scaled appearance when viewed on the main display.

**Availability**
Available in Mac OS X v10.4 and later.
Not available to 64-bit applications.

**See Also**
`HIWindowGetScaleMode`

**Declared In**
`HIGeometry.h`

## HIPointConvert

Converts a point from one coordinate space to another.

```
void HIPointConvert (
   HIPoint *ioPoint,
   HICoordinateSpace inSourceSpace,
   void *inSourceObject,
   HICoordinateSpace inDestinationSpace,
   void *inDestinationObject
);
```

**Parameters**
*ioPoint*
A pointer to the point to convert. On output, the point contains the destination coordinates.

*inSourceSpace*

A constant specifying the source coordinate space from which the point is to be converted. Some coordinate spaces require the caller to pass extra information in the `inSourceObject` parameter.

*inSourceObject*

A pointer to an object defining the source coordinate space from which the point is to be converted. You may pass an object of type `WindowRef` or `HIViewRef`. If no object is necessary, you must pass `NULL`. See "Coordinate Space Constants" (page 10) for details on which coordinate spaces require objects.

*inDestinationSpace*

A constant specifying the destination coordinate space to which the point is to be converted. Some coordinate spaces require the caller to pass extra information in the `inDestinationObject` parameter.

*inDestinationObject*

A pointer to an object defining the destination coordinate space to which the point is to be converted. You may pass an object of type `WindowRef` or `HIViewRef`. If no object is necessary, you must pass `NULL`. See "Coordinate Space Constants" (page 10) for details on which coordinate spaces require objects.

**Discussion**

When converting a point from one coordinate space to another, this function takes into account the scale factor of your application's user interface. If the source or destination object is a view, the view should already be embedded in a window. If both the source and destination objects are views, the views are not required to be embedded in the same window.

**Availability**

Available in Mac OS X v10.4 and later.

**Declared In**

`HIGeometry.h`

## HIRectConvert

Converts a rectangle from one coordinate space to another.

```
void HIRectConvert (
   HIRect *ioRect,
   HICoordinateSpace inSourceSpace,
   void *inSourceObject,
   HICoordinateSpace inDestinationSpace,
   void *inDestinationObject
);
```

**Parameters**

*ioRect*

A pointer to the rectangle to convert. On output, the rectangle contains the destination origin and size.

*inSourceSpace*

A constant specifying the source coordinate space from which the rectangle is to be converted. Some coordinate spaces require the caller to pass extra information in the `inSourceObject` parameter.

*inSourceObject*

A pointer to an object defining the source coordinate space from which the rectangle is to be converted. You may pass an object of type `WindowRef` or `HIViewRef`. If no object is necessary, you must pass `NULL`. See "Coordinate Space Constants" (page 10) for details on which coordinate spaces require objects.

*inDestinationSpace*

A constant specifying the destination coordinate space to which the rectangle is to be converted. Some coordinate spaces require the caller to pass extra information in the `inDestinationObject` parameter.

*inDestinationObject*

A pointer to an object defining the destination coordinate space to which the rectangle is to be converted. You may pass an object of type `WindowRef` or `HIViewRef`. If no object is necessary, you must pass `NULL`. See "Coordinate Space Constants" (page 10) for details on which coordinate spaces require objects.

**Discussion**

When converting a rectangle from one coordinate space to another, this function takes into account the scale factor of your application's user interface. If the source or destination object is a view, the view should already be embedded in a window. If both the source and destination objects are views, the views are not required to be embedded in the same window.

**Availability**

Available in Mac OS X v10.4 and later.

**Declared In**

`HIGeometry.h`

## HISizeConvert

Converts a size structure from one coordinate space to another.

```
void HISizeConvert (
   HISize *ioSize,
   HICoordinateSpace inSourceSpace,
   void *inSourceObject,
   HICoordinateSpace inDestinationSpace,
   void *inDestinationObject
);
```

**Parameters**

*ioSize*

A pointer to the size structure to convert. On output, the structure contains the destination size.

*inSourceSpace*

A constant specifying the source coordinate space from which the size is to be converted. Some coordinate spaces require the caller to pass extra information in the `inSourceObject` parameter.

*inSourceObject*

A pointer to an object defining the source coordinate space from which the size is to be converted. You may pass an object of type `WindowRef` or `HIViewRef`. If no object is necessary, you must pass `NULL`. See "Coordinate Space Constants" (page 10) for details on which coordinate spaces require objects.

*inDestinationSpace*

A constant specifying the destination coordinate space to which the size is to be converted. Some coordinate spaces require the caller to pass extra information in the `inDestinationObject` parameter.

*inDestinationObject*

A pointer to an object defining the destination coordinate space to which the size is to be converted. You may pass an object of type `WindowRef` or `HIViewRef`. If no object is necessary, you must pass `NULL`. See "Coordinate Space Constants" (page 10) for details on which coordinate spaces require objects.

**Discussion**
When converting a size structure (width and height) from one coordinate space to another, this function takes into account the scale factor of your application's user interface. If the source or destination object is a view, the view should already be embedded in a window. If both the source and destination objects are views, the views are not required to be embedded in the same window.

**Availability**
Available in Mac OS X v10.4 and later.

**Declared In**
`HIGeometry.h`

# Data Types

## HIPoint

Defines the position of a point using floating-point coordinates.

```
typedef CGPoint HIPoint;
```

**Discussion**
The `HIPoint` type is a data structure that defines the position of a point (x,y) in a floating-point coordinate space. When you obtain a point of type `HIPoint` from an HIToolbox function or a Carbon event, typically the y-axis of the drawing coordinate space is inverted with the origin (0,0) in the upper-left corner of the main display. Note that although it replaces the QuickDraw `Point` data structure, the `HIPoint` data structure does not contain the same fields.

**Availability**
Available in Mac OS X v10.1 and later.

**Declared In**
`HIGeometry.h`

## HISize

Defines the width and height of an object using floating-point coordinates.

```
typedef CGSize HISize;
```

**Availability**
Available in Mac OS X v10.2 and later.

**Declared In**
`HIGeometry.h`

### HIRect

Defines the position and size of a rectangle using floating-point coordinates.

```
typedef CGRect HIRect;
```

**Discussion**
The `HIRect` type is a data structure that defines the position and size (width and height) of a rectangle in a floating-point coordinate space. When you obtain a rectangle of type `HIRect` from an HIToolbox function or a Carbon event, typically the y-axis of the drawing coordinate space is inverted with the origin (0,0) in the upper-left corner of the main display. In this case, the position or origin of the rectangle is its upper-left corner. Note that although it replaces the QuickDraw `Rect` data structure, the `HIRect` data structure does not contain the same fields.

**Availability**
Available in Mac OS X v10.2 and later.

**Declared In**
`HIGeometry.h`

## Constants

### Coordinate Space Constants

Specify coordinate spaces used in HIToolbox.

```
typedef UInt32 HICoordinateSpace;
enum {
    kHICoordSpace72DPIGlobal      = 1,
    kHICoordSpaceScreenPixel      = 2,
    kHICoordSpaceWindow           = 3,
    kHICoordSpaceView             = 4
};
```

**Constants**
`kHICoordSpace72DPIGlobal`
Specifies a global coordinate space that has been adjusted by the scale factor of your application's user interface. For example, if a user interface object is 125 x 100 pixels on the screen and the scale factor is 1.25, the size of the object in this coordinate space is 100 x 80. The origin of this coordinate space is the upper-left corner of the main display, and the y-axis is inverted. This is the compatibility coordinate space; Carbon functions that do not take an explicit `HICoordinateSpace` parameter, such as existing Window Manager, QuickDraw, and Display Manager functions, assume that coordinate parameters are expressed in this space. When the scale factor is 1.0, this coordinate space and `kHICoordSpaceScreenPixel` are the same.

Available in Mac OS X v10.4 and later.

Declared in `HIGeometry.h`.

`kHICoordSpaceScreenPixel`

Specifies a coordinate space defined by the screen size in pixels. The origin of this coordinate space is the upper-left corner of the main display, and the y-axis is inverted. When the scale factor of your application's user interface is 1.0, this coordinate space and `kHICoordSpace72DPIGlobal` are the same.

Available in Mac OS X v10.4 and later.

Declared in `HIGeometry.h`.

`kHICoordSpaceWindow`

Specifies the coordinate space of a window of type `WindowRef`. The origin of this coordinate space is the upper-left corner of the window's structure region, and the y-axis is inverted. When this constant is passed to a function as a source or destination coordinate space, you must also pass a window as a source or destination object.

Available in Mac OS X v10.4 and later.

Declared in `HIGeometry.h`.

`kHICoordSpaceView`

Specifies the coordinate space of a view of type `HIViewRef`. The origin of this coordinate space is the upper-left corner of the view's bounds, and the y-axis is inverted. When this constant is passed to a function as a source or destination coordinate space, you must also pass a view as a source or destination object.

Available in Mac OS X v10.4 and later.

Declared in `HIGeometry.h`.

**Declared In**

`HIGeometry.h`

# Document Revision History

This table describes the changes to *HIGeometry Reference*.

| Date | Notes |
| --- | --- |
| 2007-06-11 | Made minor editorial changes. |
| 2006-06-28 | New document that defines basic geometric objects for HIToolbox. |

Document Revision History

# Index

## C



## H



## K