

---

# NSView Interface Builder Kit Additions Reference

[Tools > Interface Builder](#)



2007-05-10



Apple Inc.  
© 2007 Apple 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, and Aqua 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

---

## **NSView Interface Builder Kit Additions Reference 5**

---

Overview 5

Tasks 5

    Specifying the Container View 5

    Providing Suggested Sizing Information 5

    Providing Design-time Layout Information 6

    Providing Baseline Information 6

Instance Methods 6

    ibBaselineAtIndex: 6

    ibBaselineCount 7

    ibDesignableContentView 7

    ibLayoutInset 7

    ibMaximumSize 8

    ibMinimumSize 8

    ibPreferredDesignSize 8

    ibPreferredResizeDirection 9

---

## **Document Revision History 11**

---

## **Index 13**

---



# NSView Interface Builder Kit Additions Reference

---

<b>Inherits from</b>	NSResponder : NSObject
<b>Conforms to</b>	NSObject (NSObject)
<b>Framework</b>	/System/Library/Frameworks/InterfaceBuilderKit.framework
<b>Declared in</b>	InterfaceBuilderKit/IBViewIntegration.h
<b>Companion guide</b>	Interface Builder Plug-In Programming Guide

## Overview

The Interface Builder Kit framework extends the `NSView` class by adding the following:

- Methods for determining the position of Aqua guides
- Methods for determining design-time size information

Interface Builder uses these methods to discover relevant information about the views provided by your plug-in. You typically do not call these methods directly. Instead, you override them to customize the design-time information for your views.

Because these methods are relevant only in the context of Interface Builder, it is recommended that you implement these methods in a category on your class and include the category code only in your Interface Builder plug-in.

## Tasks

### Specifying the Container View

- [ibDesignableContentView](#) (page 7)  
Returns the container view for the children of the receiver.

### Providing Suggested Sizing Information

- [ibPreferredDesignSize](#) (page 8)  
Returns the recommended size for the receiver.

- [ibPreferredResizeDirection](#) (page 9)  
Returns the direction in which the receiver should grow when its size changes programmatically.

## Providing Design-time Layout Information

- [ibLayoutInset](#) (page 7)  
Returns the inset values for the receiver.
- [ibMinimumSize](#) (page 8)  
Returns the minimum size of the receiver.
- [ibMaximumSize](#) (page 8)  
Returns the maximum size of the receiver.

## Providing Baseline Information

- [ibBaselineCount](#) (page 7)  
Returns the number of baselines available for guide alignment in the receiver.
- [ibBaselineAtIndex:](#) (page 6)  
Returns the distance from the view's y origin to the specified baseline.

## Instance Methods

### **ibBaselineAtIndex:**

Returns the distance from the view's y origin to the specified baseline.

```
- (float)ibBaselineAtIndex:(int) index
```

#### **Parameters**

*index*

The index of the desired baseline. This value is between 0 and the number of baselines returned by the `ibBaselineCount` method.

#### **Return Value**

The distance from the view's y origin to the specified baseline, measured in the view's local coordinate system.

#### **Discussion**

Interface Builder calls this method to retrieve the baseline positions associated with the receiver. You should not need to call this method directly.

Baselines are used for the alignment of text in a view. Interface Builder uses this information to compute the locations of Aqua guides that appear during layout.

#### **See Also**

- [ibBaselineCount](#) (page 7)

## ibBaselineCount

Returns the number of baselines available for guide alignment in the receiver.

```
- (int)ibBaselineCount
```

### Return Value

The total number of baselines available with this view. The default implementation returns 0.

### Discussion

Interface Builder calls this method to retrieve the number of baselines associated with the receiver. If your view defines any baselines other than its basic boundaries (plus any relevant inset), you should override this method and return the number of additional baselines.

## ibDesignableContentView

Returns the container view for the children of the receiver.

```
- (NSView *)ibDesignableContentView
```

### Return Value

The container view for the receiver's children, or `nil` if the view does not contain children.

### Discussion

If you are implementing a container view, you should override this method and return the view used to contain any child views. For most views, you would simply return `self` from this method. If the receiver exposes child views but does not directly contain them, however, you should return the view that does contain the exposed child views. For example, a scroll view returns its clip view, which is the actual container for the document view.

## ibLayoutInset

Returns the inset values for the receiver.

```
- (IBInset)ibLayoutInset
```

### Return Value

A structure containing the inset values (measured in points) for each edge of the receiver. Positive inset values shrink the size of your view while negative values expand it.

### Discussion

Inset values let you specify the amount by which to shrink your view's bounds rectangle in order to lay it out visually. You should provide custom inset boundaries if the visible part of your view does not occupy the entire view bounds. The inset values combined with the view bounds should result in a rectangle that just encloses the relevant portion of your view.

For example, imagine a control that contains a shadow effect. Suppose the control occupies the entire view except for a small border along the bottom and right side that contain a shadow. If the shadow extends 3 points to the bottom and right of the control, this method should return a `IBInset` structure with the values `{0, 0, 3, 3}`.

For more information about layout boundaries, see *Interface Builder Plug-In Programming Guide*.

## **ibMaximumSize**

Returns the maximum size of the receiver.

- (NSSize)ibMaximumSize

### **Return Value**

The maximum size of the receiver, specified using the view's own local (bounds) coordinate space.

### **Discussion**

For views with no practical size limit, you can return an arbitrarily large value, such as 10000, for each dimension. This value affects only the design-time size of the view. You can still modify the size of the view at runtime using the view's `setFrame:` and `setSize:` methods.

The size values you return should use the same coordinate system as your view's bounds rectangle. Normally, a view's bounds rectangle is tied to its frame rectangle and coordinates are measured in points (1/72 of an inch). It is possible, however, to detach a view's bounds coordinates from its frame coordinates. You might do this so that your view can use a fixed coordinate system internally, regardless of the view's actual size. Interface Builder automatically converts the bounds coordinates you return to window coordinates before performing any layout.

## **ibMinimumSize**

Returns the minimum size of the receiver.

- (NSSize)ibMinimumSize

### **Return Value**

The minimum size of the receiver, specified using the view's own local (bounds) coordinate space.

### **Discussion**

This value affects only the design-time minimum size of the view. You can still modify the size of the view at runtime using the view's `setFrame:` and `setFrameSize:` methods.

The size values you return should use the same coordinate system as your view's bounds rectangle. Normally, a view's bounds rectangle is tied to its frame rectangle and coordinates are measured in points (1/72 of an inch). It is possible, however, to detach a view's bounds coordinates from its frame coordinates. You might do this so that your view can use a fixed coordinate system internally, regardless of the view's actual size. Interface Builder automatically converts the bounds coordinates you return to window coordinates before performing any layout.

## **ibPreferredSize**

Returns the recommended size for the receiver.

- (NSSize)ibPreferredSize

### **Return Value**

The suggested size for this view. This method returns the view's current size by default.



## **ibPreferredResizeDirection**

Returns the direction in which the receiver should grow when its size changes programmatically.

- (IBDirection)ibPreferredResizeDirection

### **Return Value**

The suggested growth direction. For a list of possible values, see *Interface Builder Kit Constants Reference*.

### **Discussion**

Programmatic changes to the size of a view might come from Interface Builder or from the user changing values in the inspector. System controls return the suggested horizontal growth direction associated with their cells by default.



# Document Revision History

---

This table describes the changes to *NSView Interface Builder Kit Additions Reference*.

Date	Notes
2007-05-10	New document describing methods added to the NSView class to support their manipulation in Interface Builder.

## REVISION HISTORY

### Document Revision History

# Index

---

## I

---

`ibBaselineAtIndex:` **instance method** [6](#)  
`ibBaselineCount` **instance method** [7](#)  
`ibDesignableContentView` **instance method** [7](#)  
`ibLayoutInset` **instance method** [7](#)  
`ibMaximumSize` **instance method** [8](#)  
`ibMinimumSize` **instance method** [8](#)  
`ibPreferredDesignSize` **instance method** [8](#)  
`ibPreferredResizeDirection` **instance method** [9](#)