
NSDecimalNumberHandler Class Reference

[Cocoa](#) > [Data Management](#)



2007-04-30



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 Cocoa are trademarks of Apple Inc., registered in the United States and other countries.

iPhone and Numbers are trademarks of Apple Inc.

Times is a registered trademark of Heidelberger Druckmaschinen AG, available from Linotype Library GmbH.

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

NSDecimalNumberHandler Class Reference 5

Overview 5

Adopted Protocols 5

Tasks 6

 Creating a Decimal Number Handler 6

 Initializing a Decimal Number Handler 6

Class Methods 6

 decimalNumberHandlerWithRoundingMode:scale:raiseOnExactness:raiseOnOverflow:
 raiseOnUnderflow:raiseOnDivideByZero: 6

 defaultDecimalNumberHandler 7

Instance Methods 7

 initWithRoundingMode:scale:raiseOnExactness:raiseOnOverflow:raiseOnUnderflow:
 raiseOnDivideByZero: 7

Document Revision History 9

Index 11

NSDecimalNumberHandler Class Reference

| | |
|------------------------|---|
| Inherits from | NSObject |
| Conforms to | NSCoding NSDecimalNumberBehaviors NSObject (NSObject) |
| Framework | /System/Library/Frameworks/Foundation.framework |
| Availability | Available in Mac OS X v10.0 and later. |
| Companion guide | Number and Value Programming Topics for Cocoa |
| Declared in | NSDecimalNumber.h |

Overview

`NSDecimalNumberHandler` is a class that adopts the `NSDecimalNumberBehaviors` protocol. This class allows you to set the way an `NSDecimalNumber` object rounds off and handles errors, without having to create a custom class.

You can use an instance of this class as an argument to any of the `NSDecimalNumber` methods that end with `...Behavior:`. If you don't think you need special behavior, you probably don't need this class—it is likely that `NSDecimalNumber`'s default behavior will suit your needs.

For more information, see the `NSDecimalNumberBehaviors` protocol specification.

Adopted Protocols

NSDecimalNumberBehaviors

- `roundingMode`
- `scale`
- `exceptionDuringOperation:error:leftOperand:rightOperand:`

NSCoding

- `encodeWithCoder:`
- `initWithCoder:`

Tasks

Creating a Decimal Number Handler

- + [defaultDecimalNumberHandler](#) (page 7)
Returns the default instance of `NSDecimalNumberHandler`.
- + [decimalNumberHandlerWithRoundingMode:scale:raiseOnExactness:raiseOnOverflow:raiseOnUnderflow:raiseOnDivideByZero:](#) (page 6)
Returns an `NSDecimalNumberHandler` object with customized behavior.

Initializing a Decimal Number Handler

- [initWithRoundingMode:scale:raiseOnExactness:raiseOnOverflow:raiseOnUnderflow:raiseOnDivideByZero:](#) (page 7)
Returns an `NSDecimalNumberHandler` object initialized so it behaves as specified by the method's arguments.

Class Methods

decimalNumberHandlerWithRoundingMode:scale:raiseOnExactness:raiseOnOverflow:raiseOnUnderflow:raiseOnDivideByZero:

Returns an `NSDecimalNumberHandler` object with customized behavior.

```
+ (id)decimalNumberHandlerWithRoundingMode:(NSRoundingMode)roundingMode
  scale:(short)scale raiseOnExactness:(BOOL)raiseOnExactness
  raiseOnOverflow:(BOOL)raiseOnOverflow raiseOnUnderflow:(BOOL)raiseOnUnderflow
  raiseOnDivideByZero:(BOOL)raiseOnDivideByZero
```

Parameters

roundingMode

The rounding mode to use. There are four possible values: `NSRoundUp`, `NSRoundDown`, `NSRoundPlain`, and `NSRoundBankers`.

scale

The number of digits a rounded value should have after its decimal point.

raiseOnExactness

If YES, in the event of an exactness error the handler will raise an exception, otherwise it will ignore the error and return control to the calling method.

raiseOnOverflow

If YES, in the event of an overflow error the handler will raise an exception, otherwise it will ignore the error and return control to the calling method.

raiseOnUnderflow

If YES, in the event of an underflow error the handler will raise an exception, otherwise it will ignore the error and return control to the calling method.

raiseOnDivideByZero

If YES, in the event of a divide by zero error the handler will raise an exception, otherwise it will ignore the error and return control to the calling method

Return Value

An `NSDecimalNumberHandler` object with customized behavior.

Discussion

See the `NSDecimalNumberBehaviors` protocol specification for a complete explanation of the possible behaviors.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSDecimalNumber.h`

defaultDecimalNumberHandler

Returns the default instance of `NSDecimalNumberHandler`.

```
+ (id)defaultDecimalNumberHandler
```

Return Value

The default instance of `NSDecimalNumberHandler`.

Discussion

This default decimal number handler rounds to the closest possible return value. It assumes your need for precision does not exceed 38 significant digits, and it raises an exception when its `NSDecimalNumber` object tries to divide by 0 or when its `NSDecimalNumber` object produces a number too big or too small to be represented.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSDecimalNumber.h`

Instance Methods

initWithRoundingMode:scale:raiseOnExactness:raiseOnOverflow:raiseOnUnderflow:raiseOnDivideByZero:

Returns an `NSDecimalNumberHandler` object initialized so it behaves as specified by the method's arguments.

```
- (id)initWithRoundingMode:(NSRoundingMode)roundingMode scale:(short)scale
  raiseOnExactness:(BOOL)raiseOnExactness raiseOnOverflow:(BOOL)raiseOnOverflow
  raiseOnUnderflow:(BOOL)raiseOnUnderflow
  raiseOnDivideByZero:(BOOL)raiseOnDivideByZero
```

Parameters*roundingMode*

The rounding mode to use. There are four possible values: NSRoundUp, NSRoundDown, NSRoundPlain, and NSRoundBankers.

scale

The number of digits a rounded value should have after its decimal point.

raiseOnExactness

If YES, in the event of an exactness error the handler will raise an exception, otherwise it will ignore the error and return control to the calling method.

raiseOnOverflow

If YES, in the event of an overflow error the handler will raise an exception, otherwise it will ignore the error and return control to the calling method

raiseOnUnderflow

If YES, in the event of an underflow error the handler will raise an exception, otherwise it will ignore the error and return control to the calling method

raiseOnDivideByZero

If YES, in the event of a divide by zero error the handler will raise an exception, otherwise it will ignore the error and return control to the calling method

Return Value

An initialized NSDecimalNumberHandler object initialized with customized behavior. The returned object might be different than the original receiver.

Discussion

See the NSDecimalNumberBehaviors protocol specification for a complete explanation of the possible behaviors.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSDecimalNumber.h

Document Revision History

This table describes the changes to *NSNumberFormatter Class Reference*.

| Date | Notes |
|------------|---|
| 2007-04-30 | Corrected "Conforms to" section to include NSCodering. |
| 2006-05-23 | First publication of this content as a separate document. |

REVISION HISTORY

Document Revision History

Index

D

`decimalNumberHandlerWithRoundingMode:scale:
raiseOnExactness:raiseOnOverflow:raiseOnUnderflow:
raiseOnDivideByZero:` **class method** [6](#)
`defaultDecimalNumberHandler` **class method** [7](#)

I

`initWithRoundingMode:scale:raiseOnExactness:
raiseOnOverflow:raiseOnUnderflow:
raiseOnDivideByZero:` **instance method** [7](#)