

---

# NSComboBoxCellDataSource Protocol Reference

[Cocoa](#) > [User Experience](#)



2006-05-23



Apple Inc.  
© 2006 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, Cocoa, Mac, and Mac OS 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

---

## **NSComboBoxCellDataSource Protocol Reference 5**

---

Overview 5

Tasks 5

    Returning Information About Combo Box Items 5

    Working with Entered Strings 5

Instance Methods 6

    comboBoxCell:completedString: 6

    comboBoxCell:indexOfItemWithStringValue: 6

    comboBoxCell:objectValueForItemAtIndex: 7

    numberOfItemsInComboBoxCell: 7

---

## **Document Revision History 9**

---

## **Index 11**

---



# NSComboBoxCellDataSource Protocol Reference

(informal protocol)

---

<b>Framework</b>	/System/Library/Frameworks/AppKit.framework
<b>Companion guide</b>	Combo Box Programming Topics
<b>Declared in</b>	NSComboBoxCell.h

## Overview

The `NSComboBoxCellDataSource` category declares the methods that an `NSComboBoxCell` uses to access the contents of its data source object.

For more information, see “Providing Data for a Combo Box”.

## Tasks

### Returning Information About Combo Box Items

- [comboBoxCell:objectValueForItemAtIndex:](#) (page 7)  
Returns the object that corresponds to the item at the given index in the combo box cell.
- [numberOfItemsInComboBoxCell:](#) (page 7)  
Returns the number of items managed for the combo box cell by your data source object.

### Working with Entered Strings

- [comboBoxCell:completedString:](#) (page 6)  
Returns the item from the combo box's pop-up list that matches the text entered by the user.
- [comboBoxCell:indexOfItemWithStringValue:](#) (page 6)  
Invoked by an `NSComboBoxCell` object to synchronize the pop-up list's selected item with the text field's contents.

## Instance Methods

### **comboBoxCell:completedString:**

Returns the item from the combo box's pop-up list that matches the text entered by the user.

```
- (NSString *)comboBoxCell:(NSComboBoxCell *)aComboBoxCell completedString:(NSString *)uncompletedString
```

#### **Parameters**

*aComboBoxCell*

The combo box cell.

*uncompletedString*

The substring containing the text the user typed into the text field of the combo box cell.

#### **Return Value**

The completed string, from the items in the pop-up list, that matches the text entered by the user. Your implementation should return the first complete string that starts with *uncompletedString*.

#### **Discussion**

An `NSComboBoxCell` object uses this method to perform incremental—or “smart”—searches when the user types into the text field.

As the user types in the text field, the receiver uses this method to search for items from the pop-up list that start with what the user has typed. The receiver adds the new text to the end of the field and selects the new text, so when the user types another character, it replaces the new text.

This method is optional. If you don't implement it, the receiver does not perform incremental searches.

#### **Availability**

Available in Mac OS X v10.0 and later.

#### **Declared In**

`NSComboBoxCell.h`

### **comboBoxCell:indexOfItemWithStringValue:**

Invoked by an `NSComboBoxCell` object to synchronize the pop-up list's selected item with the text field's contents.

```
- (NSInteger)comboBoxCell:(NSComboBoxCell *)aComboBoxCell  
  indexOfItemWithStringValue:(NSString *)aString
```

#### **Parameters**

*aComboBoxCell*

The combo box cell.

*aString*

The string to match. If `comboBoxCell:completedString:` (page 6) is implemented, *aString* is the string returned by that method. Otherwise, *aString* is the text that the user has typed.

#### **Return Value**

The index for the pop-up list item matching *aString*, or `NSNotFound` if no item matches.

**Discussion**

This method is optional. If you don't implement it, the receiver does not synchronize the pop-up list's selected item with the text field's contents.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSComboBoxCell.h

**comboBoxCell:objectValueForItemAtIndex:**

Returns the object that corresponds to the item at the given index in the combo box cell.

```
- (id)comboBoxCell:(NSComboBoxCell *)aComboBoxCell
  objectValueForItemAtIndex:(NSInteger)index
```

**Parameters**

*aComboBoxCell*

The combo box cell for which to return the item.

*index*

The index of the item to return.

**Return Value**

The object corresponding to the item at the specified index in the given combo box cell.

**Discussion**

Your data source must implement this method.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSComboBoxCell.h

**numberOfItemsInComboBoxCell:**

Returns the number of items managed for the combo box cell by your data source object.

```
- (NSInteger)numberOfItemsInComboBoxCell:(NSComboBoxCell *)aComboBoxCell
```

**Parameters**

*aComboBoxCell*

The combo box cell for which your data source manages items.

**Return Value**

The number of items your data source object manages.

**Discussion**

An `NSComboBoxCell` object uses this method to determine how many items it should display in its pop-up list. Your data source must implement this method.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSComboBoxCell.h



# Document Revision History

---

This table describes the changes to *NSComboBoxCellDataSource Protocol Reference*.

Date	Notes
2006-05-23	First publication of this content as a separate document.

## REVISION HISTORY

### Document Revision History

# Index

---

## C

---

comboBoxCell:completedString: <NSObject>  
instance method [6](#)

comboBoxCell:indexOfItemWithStringValue:  
<NSObject> instance method [6](#)

comboBoxCell:objectValueForItemAtIndex:  
<NSObject> instance method [7](#)

## N

---

numberOfItemsInComboBoxCell: <NSObject> instance  
method [7](#)