Forms

Cocoa > User Experience

2002-11-12

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

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

Introduction to Forms

A form is group of related text fields. Unlike individual text fields, the fields in a form work together without any intervention from the developer. For example, you can tab from one field to another without writing any code or doing any work in Interface Builder.

Organization of This Document

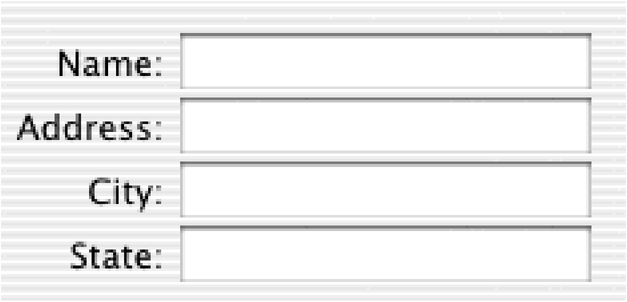
This topic discusses forms. "How Forms Work" (page 7) gives basic information on them. "Setting a Form's Appearance" (page 9) describes how to modify the form's appearance. "Managing Form Entries" (page 11) describes how to access and modify individual entries in a form.

A form is implemented as a type of matrix. For more information, see Matrices.

How Forms Work

A form is group of related text fields. Unlike individual text fields, the fields in a form work together without any intervention from the developer. For example, you can tab from one field to another without writing any code or doing any work in Interface Builder.

A form is implemented as a NSForm, a subclass of NSMatrix, which contains a column of NSFormCells. Here's an example:



Name:

Address:

City:

State:

In NSForm's methods, each NSFormCell is called an "entry" (or, sometimes, a "cell" or "item"). The left part of each entry is called the "title," and the right part is called the "text." Methods that refer to individual entries use an one-dimensional "index"; the indexing system starts at the top of the top of form, with zero.

Generally, you'll create and modify a form in Interface Builder. You can also create one programmatically with one of NSForm's constructors. To add an entry to the end of a form, use `addEntry:`. To add an entry at a specific index, use `insertEntry:atIndex:`. To remove an entry, use `removeEntryAtIndex:`.

Any entry in the form can be "selected." When an entry is selected, its text area responds to the user's keystrokes. You can select an entry using the `selectTextAtIndex:` method, or you can let the user select an entry by clicking it with the mouse. Once an entry is selected, the user can select the next entry by pressing Tab, or select the previous entry by pressing Shift-Tab.

To initiate the action of a selected entry, the user presses Return or Enter. In response, the entry sends an action message to its target. If the entry has no target, the NSForm sends an action message to its target.

Setting a Form's Appearance

Generally, you'll use Interface Builder to modify the appearance of a form. But you can also modify it programmatically, with the methods described here.

Use these NSForm methods to set the appearance of all the entries in a form:

- To set the font of the entries' titles or text fields, use `setTitleFont:` or `setTextFont:`.
- To set the alignment of the entries' titles or text fields, use `setTitleAlignment:` or `setTextAlignment:`., with one of these as the argument: `NSRightTextAlignment`, `NSCenterTextAlignment`, or `NSLeftTextAlignment`.
- To set the spacing between entries (in pixels), use `setInterlineSpacing:`.
- To set the width of the entries (including their titles and text fields), use `setEntryWidth:`.
- To set how the text fields are outlined, use `setBordered:` and `setBezeled:`. A border is a thin line around the field. A bezel is an outline shaded to look three-dimensional. If both of these are set to `NO`, then the text fields are not outlined.

Use the NSFormCell methods in the following list to set the appearance of one of the entries in a form. You can access a particular entry with `cellAtIndex:`; for example, `[myForm cellAtIndex:1]`.

- To set the font of an entry's title or text field, use `setTitleFont:` or `setTextFont:`.
- To set the alignment of the entries' titles or text fields, use `setTitleAlignment:` or `setTextAlignment:`., with one of these as the argument: `NSRightTextAlignment`, `NSCenterTextAlignment`, or `NSLeftTextAlignment`.
- To set the title of an entry, use `setTitle:`. If you want the title to contained styled text, use `setAttributedTitle:`.

Managing Form Entries

NSForm has several methods for manipulating its entries. These methods dynamically add and remove entries:

- To add an entry, use `addEntry:` and `insertEntry:atIndex:`.
- To remove an entry, use `removeEntryAtIndex:`.

These methods find and select a particular entry:

- To get the index of the entry with a specified tag, use `indexOfCellWithTag:`.
- To get the index of the entry that contains the insertion point, use `indexOfSelectedItem`.
- To get the entry at a specified index, use `cellAtIndex:`
- To select all the text in a specified entry, `selectTextAtIndex:`.

|

Document Revision History

This table describes the changes to *Forms*.

Date	Notes
2002-11-12	Revision history was added to existing topic. It will be used to record changes to the content of the topic.