



# INSIDE MACINTOSH

---

## Mac OS 8.5 Dialog Manager Reference



January 12, 1999  
Technical Publications  
© 1999 Apple Computer, Inc.

 Apple Computer, Inc.  
© 1998 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 Computer, Inc., except to make a backup copy of any documentation provided on CD-ROM.

The Apple logo is a trademark of Apple Computer, 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 book. Apple retains all intellectual property rights associated with the technology described in this book. This book is intended to assist application developers to develop applications only for Apple-labeled or Apple-licensed computers.

Every effort has been made to ensure that the information in this manual is accurate. Apple is not responsible for typographical errors.

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

Apple, the Apple logo, Mac, and Macintosh are trademarks of Apple Computer, Inc., registered in the United States and other countries. Adobe, Acrobat, and PostScript are trademarks of Adobe Systems Incorporated or its subsidiaries and may be registered in certain jurisdictions.

Adobe, the Adobe logo, Acrobat, the Acrobat logo, Distiller, PostScript,

and the PostScript logo are trademarks of Adobe Systems Incorporated.

Helvetica and Palatino are registered trademarks of Linotype-Hell AG and/or its subsidiaries.

ITC Zapf Dingbats is a registered trademark of International Typeface Corporation.

Simultaneously published in the United States and Canada.

**Even though Apple has reviewed this manual, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS MANUAL, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS MANUAL IS SOLD “AS IS,” AND YOU, THE PURCHASER, 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 MANUAL, 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

Chapter 1	Mac OS 8.5 Dialog Manager Reference	5
	Gestalt Selector for the Mac OS 8.5 Dialog Manager	7
	Functions for the Mac OS 8.5 Dialog Manager	8
	Simulating User Responses in Dialog Boxes	8
	Filtering Dialog Box Events	11
	Result Codes for the Mac OS 8.5 Dialog Manager	13
Appendix A	Document Version History	15
	Index	17



# Mac OS 8.5 Dialog Manager Reference

---

## Contents

<b>Gestalt Selector for the Mac OS 8.5 Dialog Manager</b>	<b>7</b>
<b>Functions for the Mac OS 8.5 Dialog Manager</b>	<b>8</b>
<b>Simulating User Responses in Dialog Boxes</b>	<b>8</b>
GetDialogTimeout	<b>8</b>
SetDialogTimeout	<b>9</b>
<b>Filtering Dialog Box Events</b>	<b>11</b>
GetModalDialogEventMask	<b>11</b>
SetModalDialogEventMask	<b>12</b>
<b>Result Codes for the Mac OS 8.5 Dialog Manager</b>	<b>13</b>



The Dialog Manager provides a simplified human interface that you can use to readily implement dialog boxes and alert boxes. Your program typically uses dialog boxes and alert boxes to present information to and solicit information from the user.

This document describes the Dialog Manager application programming interface (API) introduced with Mac OS 8.5 and Appearance Manager 1.1, as follows:

- “Gestalt Selector for the Mac OS 8.5 Dialog Manager” (page 7)
- “Functions for the Mac OS 8.5 Dialog Manager” (page 8)
- “Result Codes for the Mac OS 8.5 Dialog Manager” (page 13)

Note that pre-existing Dialog Manager functions, types, and constants are not discussed in this document. For a description of the Mac OS 8 Dialog Manager API, see *Mac OS 8 Dialog Manager Reference*. For descriptions of the pre-Mac OS 8 Dialog Manager API, see *Inside Macintosh: Macintosh Toolbox Essentials*.

## Gestalt Selector for the Mac OS 8.5 Dialog Manager

---

Before calling any Dialog Manager functions, your application should pass the selector `gestaltDialogManagerAttr` to the `Gestalt` function to determine which Dialog Manager functions are available.

```
enum {
    gestaltDialogManagerAttr    = 'dlog',
    gestaltDialogManagerPresent = (1L << 0)
};
```

### Constant descriptions

`gestaltDialogManagerAttr`

The `Gestalt` selector passed to determine what features of the Dialog Manager are present. This selector is available with Mac OS 8.5 and later. Passing `gestaltDialogManagerAttr` produces a 32-bit value whose bits you should test to determine what Dialog Manager functionality is available.

`gestaltDialogManagerPresent`

If the bit specified by this mask is set, the Dialog Manager functionality for Appearance Manager 1.1 is available. This bit is set for Mac OS 8.5 and later.

## Functions for the Mac OS 8.5 Dialog Manager

---

The Mac OS 8.5 Dialog Manager provides new functions in the following areas:

- “Simulating User Responses in Dialog Boxes” (page 8)
- “Filtering Dialog Box Events” (page 11)

### Simulating User Responses in Dialog Boxes

---

The Mac OS 8.5 Dialog Manager provides the following functions for simulating user responses in dialog boxes:

- `GetDialogTimeout` (page 8) obtains the original countdown duration, the time remaining, and the item selection to be simulated for a specified modal dialog box.
- `SetDialogTimeout` (page 9) simulates an item selection in a modal dialog box after a specified amount of time elapses.

### GetDialogTimeout

---

Obtains the original countdown duration, the time remaining, and the item selection to be simulated for a specified modal dialog box.

```
pascal OSStatus GetDialogTimeout (
    DialogPtr inDialog,
    SInt16 *outButtonToPress,
    UInt32 *outSecondsToWait,
    UInt32 *outSecondsRemaining);
```

`inDialog`      A value of type `DialogPtr`. Pass a pointer to the dialog box to be examined.

`outButtonToPress`

A pointer to a signed 16-bit integer. On return, a value representing the number within the item list of the item that is to be selected. You may pass `nil` for the `outButtonToPress` parameter if you do not desire this information.

`outSecondsToWait`

A pointer to an unsigned 32-bit integer. On return, a value specifying the number of seconds that were originally set to elapse before the Dialog Manager simulates an item selection. You may pass `nil` for the `outSecondsToWait` parameter if you do not desire this information.

`outSecondsRemaining`

A pointer to an unsigned 32-bit integer. On return, a value specifying the number of seconds remaining before the Dialog Manager simulates an item selection. You may pass `nil` for the `outSecondsRemaining` parameter if you do not desire this information.

*function result* A result code. See “Result Codes for the Mac OS 8.5 Dialog Manager” (page 13).

#### VERSION NOTES

Available with Mac OS 8.5 and later.

#### SEE ALSO

The function `SetDialogTimeout` (page 9).

### SetDialogTimeout

---

Simulates an item selection in a modal dialog box after a specified amount of time elapses.

```
pascal OSStatus SetDialogTimeout (
    DialogPtr inDialog,
    SInt16 inButtonToPress,
    UInt32 inSecondsToWait);
```

- `inDialog` A value of type `DialogPtr`. Pass a pointer to the dialog box for which an item selection is to be simulated.
- `inButtonToPress` A signed 16-bit integer. Pass a value representing the number within the item list of the item that is to be selected.
- `inSecondsToWait` An unsigned 32-bit integer. Pass a value specifying the number of seconds that are to elapse before the Dialog Manager simulates an item selection. Pass 0 to clear a preexisting timeout value and cease the countdown in progress.
- function result* A result code. See “Result Codes for the Mac OS 8.5 Dialog Manager” (page 13).

## DISCUSSION

Your application calls the `SetDialogTimeout` function each time you wish to start a countdown of the specified duration for a given modal dialog box. When the amount of time specified in the `inSecondsToWait` parameter has elapsed, the Dialog Manager simulates a click on the button specified in the `inButtonToPress` parameter. If your application calls `SetDialogTimeout` again, or if any event is received for the dialog box, the countdown is restarted.

In order to use `SetDialogTimeout` with a given modal dialog box, your application must handle events for the dialog box through the `ModalDialog` function. The Dialog Manager will not simulate an item selection for the dialog box until `ModalDialog` processes an event (including null events).

## VERSION NOTES

Available with Mac OS 8.5 and later.

## SEE ALSO

The function `GetDialogTimeout` (page 8).

## Filtering Dialog Box Events

---

The Mac OS 8.5 Dialog Manager provides the following functions to filter dialog box events:

- `GetModalDialogEventMask` (page 11) obtains the events to be received by the `ModalDialog` function.
- `SetModalDialogEventMask` (page 12) specifies the events to be received by the `ModalDialog` function.

### GetModalDialogEventMask

---

Obtains the events to be received by the `ModalDialog` function.

```
pascal OSStatus GetModalDialogEventMask (
    DialogPtr inDialog,
    EventMask *outMask);
```

`inDialog`      A value of type `DialogPtr`. Pass a pointer to the dialog box for which you wish to obtain the event mask.

`outMask`        A pointer to a unsigned 16-bit integer of type `EventMask`. On return, your application may test the bits of this value to determine the event(s) that the dialog box is currently set to receive. See *Inside Macintosh: Macintosh Toolbox Essentials* for a description of event masks.

*function result* A result code. See “Result Codes for the Mac OS 8.5 Dialog Manager” (page 13).

#### VERSION NOTES

Available with Mac OS 8.5 and later.

#### SEE ALSO

The function `SetModalDialogEventMask` (page 12).

**SetModalDialogEventMask**

Specifies the events to be received by the `ModalDialog` function.

```
pascal OSStatus SetModalDialogEventMask (
    DialogPtr inDialog,
    EventMask inMask);
```

`inDialog`     A value of type `DialogPtr`. Pass a pointer to the dialog box for which you wish to set the event mask.

`inMask`        An unsigned 16-bit integer of type `EventMask`. Pass the desired mask value(s) for the event(s) you wish the dialog box to receive. See *Inside Macintosh: Macintosh Toolbox Essentials* for descriptions of event mask constants.

*function result*   A result code. See “Result Codes for the Mac OS 8.5 Dialog Manager” (page 13).

**DISCUSSION**

Your application can use the `SetModalDialogEventMask` function to specify the events received by the `ModalDialog` function for a given modal dialog box. This allows your application to specify additional events that are not by default received by `ModalDialog`, such as disk-inserted events and operating-system events. If you use `SetModalDialogEventMask` to change the `ModalDialog` function’s event mask, you should pass `ModalDialog` a pointer to your own event filter function to handle any added events.

**VERSION NOTES**

Available with Mac OS 8.5 and later.

**SEE ALSO**

The function `GetModalDialogEventMask` (page 11).

## Result Codes for the Mac OS 8.5 Dialog Manager

---

The result codes most commonly returned by the Mac OS 8.5 Dialog Manager are listed below.

noErr	0	No error
paramErr	-50	Error in parameter list
dialogNoTimeoutErr	-5640	No timeout value specified

## CHAPTER 1

### Mac OS 8.5 Dialog Manager Reference

# Document Version History

---

This document has had the following releases:

**Table A-1** *Mac OS 8.5 Dialog Manager Reference* Revision History

---

<b>Version</b>	<b>Notes</b>
Jan. 12, 1999	Initial public release. The following changes were made from the prior version:  Changed “Dialog Manager 2.0” to “Mac OS 8.5 Dialog Manager” throughout to reflect final versioning.  “Gestalt Selector for the Mac OS 8.5 Dialog Manager” (page 7). Added description.
Apr. 28, 1998	First seed draft release.

**A P P E N D I X**

Document Version History

# Index

---

## D

---

dialogNoTimeoutErr **result code** 13

## F

---

filtering dialog box events 11

## G

---

gestaltDialogManagerAttr **constant** 7

gestaltDialogManagerPresent **constant** 8

**Gestalt selector** 7

GetDialogTimeout **function** 8

GetModalDialogEventMask **function** 11

## S

---

SetDialogTimeout **function** 9

SetModalDialogEventMask **function** 12

simulating user responses in dialog boxes 8

This Apple manual was written, edited, and composed on a desktop publishing system using Apple Macintosh computers and FrameMaker software. Line art was created using Adobe™ Illustrator and Adobe Photoshop.

Text type is Palatino® and display type is Helvetica®. Bullets are ITC Zapf Dingbats®. Some elements, such as program listings, are set in Adobe Letter Gothic.

WRITER

Donna S. Lee

PRODUCTION EDITOR

Glen Frank

Acknowledgments to Guy Fullerton,  
Pete Gontier, and Chris Thomas.