
Notification Manager Reference

[Carbon > Events & Other Input](#)





Apple Inc.
© 2002 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, Carbon, 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

Notification Manager Reference 5

Overview	5
Functions	6
DisposeNMUPP	6
InvokeNMUPP	6
NewNMUPP	6
NMInstall	7
NMRemove	7
Callbacks	8
NMProcPtr	8
Data Types	8
NMRec	8
Result Codes	10
Gestalt Constants	10

Document Revision History 11

Index 13

Notification Manager Reference

Framework:	Carbon/Carbon.h
Declared in	Notification.h

Overview

The Notification Manager provides a notification service. It allows software running in the background (or otherwise unseen by the user) to communicate information to the user. For example, applications that manage lengthy background tasks (such as printing many documents or transferring large amounts of data to other machines) might need to inform the user that the operation is complete. These applications cannot use the standard methods of communicating with the user, such as alert or dialog boxes, because such windows might easily be obscured by the windows of other applications. Moreover, even if those windows are visible, the background application cannot be certain that the user is aware of the change. A more reliable method is needed to manage the communication between a background application and the user, who might be awaiting the completion of the background task while running some other application in the foreground.

In the same way, relatively invisible operations such as Time Manager tasks, VBL tasks, or device drivers might need to inform the user that some previously started routine is complete or perhaps that some error has rendered further execution undesirable or impossible.

In all these cases, the communication generally needs to occur in one direction only, from the background application (or task, or driver) to the user. The Notification Manager, included in system software versions 6.0 and later, allows you to alert the user by posting a notification, which is an audible or visible indication that your application (or other piece of software) requires the user's attention. You post a notification by issuing a notification request to the Notification Manager, which places your request in a queue. When your request reaches the top of the queue, the Notification Manager posts a notification to the user.

You can request three types of notification:

- **Polite notification.** A small icon blinks, by periodically alternating with the Apple menu icon (the Apple logo) or the Application menu icon in the menu bar.
- **Audible notification.** The Sound Manager plays the system alert sound or a sound contained in an 'snd' resource.
- **Alert notification.** An alert box containing a short message appears on the screen. The user must dismiss the alert box (by clicking the OK button) before foreground processing can continue.

These types of notification are not mutually exclusive. For example, an application can request both audible and alert notifications. Moreover, if the requesting software is listed in the Application menu (and hence represents a process that is loaded into memory), you can instruct the Notification Manager to place a

diamond-shaped mark next to the name of the requesting process. The mark is usually intended to prompt the user to switch the marked application into the foreground. Finally, you can request that the Notification Manager execute a notification response procedure, which is executed as the final step in a notification.

Functions

DisposeNMUPP

Releases a universal procedure pointer (UPP) to a response callback function.

```
void DisposeNMUPP (
    NMUPP userUPP
);
```

Parameters

userUPP

A valid UPP, created by calling the function [NewNMUPP](#) (page 6).

Availability

CarbonLib 1.0 and later. Mac OS X version 10.0 and later.

Declared In

Notification.h

InvokeNMUPP

Invokes a response callback function.

```
void InvokeNMUPP (
    NMRecPtr nmReqPtr,
    NMUPP userUPP
);
```

Parameters

nmReqPtr

A pointer to a notification request. For information about defining a notification request, see [NMRec](#) (page 8).

userUPP

A response callback UPP, created by calling the function [NewNMUPP](#) (page 6).

Availability

CarbonLib 1.0 and later. Mac OS X version 10.0 and later.

Declared In

Notification.h

NewNMUPP

Creates a universal procedure pointer (UPP) to a response callback function.

```
NMUPP NewNMUPP (  
    NMProcPtr userRoutine  
);
```

Parameters

userRoutine

A pointer to a response callback function. For more information about response callbacks, see [NMProcPtr](#) (page 8).

Return Value

A new UPP. When you no longer need this UPP, you should call [DisposeNMUPP](#) (page 6) to release it.

Availability

CarbonLib 1.0 and later. Mac OS X version 10.0 and later.

Declared In

Notification.h

NMInstall

Installs a notification request in the notification queue.

```
OSErr NMInstall (  
    NMRecPtr nmReqPtr  
);
```

Parameters

request

A pointer to a notification request record that the caller provides.

Return Value

A result code. If the result is non-zero, the Notification Manager cannot install the request because it contains invalid information. See [“Notification Manager Result Codes”](#) (page 10).

Availability

CarbonLib 1.0 and later. Mac OS X version 10.0 and later.

Not available to 64-bit applications.

Declared In

Notification.h

NMRemove

Removes a notification request from the notification queue.

```
OSErr NMRemove (  
    NMRecPtr nmReqPtr  
);
```

Parameters

nmReqPtr

A pointer to a notification request record.

Return Value

A result code. If the result is non-zero, the Notification Manager cannot remove the request. See [“Notification Manager Result Codes”](#) (page 10).

Availability

CarbonLib 1.0 and later. Mac OS X version 10.0 and later.

Not available to 64-bit applications.

Declared In

`Notification.h`

Callbacks

NMProcPtr

Defines a pointer to a response callback function. Your response callback function is executed as the final stage of a notification.

```
typedef void (*NMProcPtr) (
    NMRecPtr request
);
```

If you name your function `MyNMProc`, you would declare it like this:

```
void MyNMProc (
    NMRecPtr request
);
```

Parameters

request

A pointer to a notification request record.

Discussion

The `nmResp` field of the notification record defines a response function executed as the final stage of a notification. If no processing is necessary in response to the notification, then you can supply the value `NULL` in that field. If you supply a UPP to your own response function in the `nmResp` field, the Notification Manager passes it one parameter, a pointer to your notification request.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`Notification.h`

Data Types

NMRec

Defines the record format of a notification request.


```
struct NMRec {
    QElemPtr qLink;
    short qType;
    short nmFlags;
    long nmPrivate;
    short nmReserved;
    short nmMark;
    Handle nmIcon;
    Handle nmSound;
    StringPtr nmStr;
    NMUPP nmResp;
    long nmRefCon;
};
typedef struct NMRec NMRec;
typedef NMRec * NMRecPtr;
```

Fields

qLink

This field is used internally by the Notification Manager. You do not need to assign a value to this field.

qType

Indicates the type of queue. You should assign the value `nmType` to this field.

nmFlags

Reserved.

nmPrivate

Reserved.

nmReserved

Reserved.

nmMark

In Mac OS, the item to mark in the Application menu. If you do not want to display a mark, assign 0 to this field.

For additional information about the use of this field in Mac OS X, see the discussion below.

nmIcon

In Mac OS, a handle to a small icon that is to blink periodically in the menu bar. If you do not want to display an icon, assign `NULL` to this field.

nmSound

In Mac OS, a handle to a sound resource to be played. If you do not want to play a sound, assign `NULL` to this field.

nmStr

A string to appear in the notification window. In Mac OS, if you do not want a notification window you should assign `NULL` to this field.

nmResp

A UPP to a response callback function. If no processing is necessary in response to the notification, assign the value `NULL` to this field.

nmRefCon

A long integer for private use by your application.

Discussion

In Mac OS X version 10.0, only the alert notification is supported.

In Mac OS X version 10.1 and later, mark and alert notifications are both supported. If you set the `nmMark` field to a non-zero value, your process icon will bounce in the dock.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`Notification.h`

Result Codes

The most common result codes returned by the Notification Manager are listed in the table below.

Result Code	Value	Description
<code>nmTypErr</code>	-299	The notification request contains an invalid queue type. Available in Mac OS X v10.0 and later.

Gestalt Constants

You can check for version and feature availability information by using the Notification Manager selectors defined in the Gestalt Manager. For more information, see *Inside Mac OS X: Gestalt Manager Reference*.

Document Revision History

This table describes the changes to *Notification Manager Reference*.

Date	Notes
2002-12-01	Updated to include new Carbon functions.
	Added new information about notifications in Mac OS X.

REVISION HISTORY

Document Revision History

Index

D

DisposeNMUPP **function 6**

I

InvokeNMUPP **function 6**

N

NewNMUPP **function 6**
NMInstall **function 7**
NMProcPtr **callback 8**
NMRec **structure 8**
NMRemove **function 7**
nmTypErr **constant 10**