
Device Manager Reference

(Legacy)

[Carbon > File Management](#)



2007-07-10



Apple Inc.
© 2003, 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

.Mac is a registered service mark of Apple Inc.

Apple, the Apple logo, Carbon, Mac, and Mac OS are trademarks of Apple Inc., registered in the United States and other countries.

UNIX is a registered trademark of The Open Group

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

Device Manager Reference (Legacy) 5

Overview	5
Data Types	5
AddressSpaceID	5
AuxDCE	6
DCtlEntry	6
DriverFinalInfo	6
DriverInitInfo	6
DriverOpenCount	7
DriverRefNum	7
DriverReplacelInfo	8
DriverSupersededInfo	8
DRVRHeader	8
FileBasedDriverDetailed	9
FileBasedDriverRecord	9
IOCommandContents	9
IOCommandID	9
UnitNumber	9
Constants	10
chooserID	10
chooserInitMsg	10
dNeedLockMask	11
Driver Header and Device Control Entry Flags	12
dVMIImmuneBit	13
dVMIImmuneMask	14
goodbye	14
initMsg	16
Miscellaneous Device Manager constants	17
kOpenCommand	18
Result Codes	19

Appendix A

Deprecated Device Manager Reference (Legacy) Functions 25

Deprecated in Mac OS X v10.5	25
PBCloseAsync	25
PBCloseSync	26
PBReadAsync	26
PBReadSync	27
PBWaitIOComplete	29
PBWriteAsync	29
PBWriteSync	30

Document Revision History 33

Index 35

Device Manager Reference (Legacy)

Framework:	CoreServices/CoreServices.h
Declared in	Devices.h Files.h IOMacOSTypes.h

Overview

Important: The Device Manager is deprecated as of Mac OS X v10.5. You should use the I/O Kit or the File Manager instead.

In Mac OS 9 and earlier, applications used the Device Manager to open, close, and exchange information with device drivers. In addition, device drivers used the Device Manager when they needed to supply a user interface. In Mac OS X, applications that need to communicate directly with hardware devices must use the I/O Kit. For more information about the I/O Kit, see *I/O Kit Fundamentals*.

Mac OS X applications can access serial devices through the device file system. You can use the I/O Kit to obtain a path to a device file in the `/dev` directory, and then use traditional UNIX serial port access through the POSIX `termios` API. For more information, see *Accessing Hardware From Applications*.

Other software products that relied on the Device Manager in the past, such as desk accessories, should be converted into Mac OS X applications.

Note: The Device Manager header file `Devices.h` has been removed in Mac OS X v10.5 and later. The functions previously declared in this header file are now declared in `Files.h`. Documentation for these deprecated functions is available in *File Manager Reference*.

Data Types

AddressSpaceID

```
typedef MPAddressSpaceID AddressSpaceID;
```

Availability

Available in Mac OS X v10.0 and later.

Declared In

IOMacOSTypes.h

AuxDCE

```

struct AuxDCE {
    Ptr dCtlDriver;
    volatile SInt16 dCtlFlags;
    QHdr dCtlQHdr;
    SInt32 dCtlPosition;
    Handle dCtlStorage;
    SInt16 dCtlRefNum;
    SInt32 dCtlCurTicks;
    GrafPtr dCtlWindow;
    SInt16 dCtlDelay;
    SInt16 dCtlEMask;
    SInt16 dCtlMenu;
    SInt8 dCtlSlot;
    SInt8 dCtlSlotId;
    SInt32 dCtlDevBase;
    Ptr dCtlOwner;
    SInt8 dCtlExtDev;
    SInt8 fillByte;
    UInt32 dCtlNodeID;
};
typedef AuxDCE* AuxDCEPtr;
typedef AuxDCEPtr* AuxDCEHandle;

```

DCtlEntry

```

struct DCtlEntry {
    Ptr dCtlDriver;
    volatile SInt16 dCtlFlags;
    QHdr dCtlQHdr;
    volatile SInt32 dCtlPosition;
    Handle dCtlStorage;
    SInt16 dCtlRefNum;
    SInt32 dCtlCurTicks;
    GrafPtr dCtlWindow;
    SInt16 dCtlDelay;
    SInt16 dCtlEMask;
    SInt16 dCtlMenu;
};
typedef DCtlEntry* DCtlPtr;
typedef DCtlPtr* DCtlHandle;

```

DriverFinalInfo

```

struct DriverFinalInfo {
    DriverRefNum refNum;
    RegEntryID deviceEntry;
};
typedef DriverFinalInfo* DriverFinalInfoPtr;

```

DriverInitInfo

```

struct DriverInitInfo {

```

```
    DriverRefNum refNum;  
    RegEntryID deviceEntry;  
};  
typedef DriverInitInfo* DriverInitInfoPtr;
```

DriverOpenCount

```
typedef UInt32 DriverOpenCount;
```

Availability

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared In

Devices.h

DriverRefNum

```
typedef SInt16 DriverRefNum;
```

Availability

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared In

Devices.h

DriverReplaceInfo

```
typedef DriverInitInfo DriverReplaceInfo;  
typedef DriverInitInfo* DriverReplaceInfoPtr;
```

DriverSupersededInfo

```
typedef DriverFinalInfo DriverSupersededInfo;  
typedef DriverFinalInfo* DriverSupersededInfoPtr;
```

DRVRHeader

```
struct DRVRHeader {  
    short drvrFlags;  
    short drvrDelay;  
    short drvrEMask;  
    short drvrMenu;  
    short drvrOpen;  
    short drvrPrime;  
    short drvrCtl;  
    short drvrStatus;  
    short drvrClose;  
    unsigned char drvrName[1];  
};  
typedef struct DRVRHeader DRVRHeader;  
typedef DRVRHeader * DRVRHeaderPtr;
```

Availability

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared In

Devices.h

FileBasedDriverDetailed

```

struct FileBasedDriverDetailed {
    FileBasedDriverRecord fileBasedDriver;
    Str63 fragName;
};
typedef FileBasedDriverDetailed* FileBasedDriverDetailedPtr;

```

FileBasedDriverRecord

```

struct FileBasedDriverRecord {
    FSSpec theSpec;
    MacDriverType theType;
    Boolean compatibleProp;
    UInt8 pad[3];
};
typedef FileBasedDriverRecord* FileBasedDriverRecordPtr;

```

IOCommandContents

```

union IOCommandContents {
    ParmBlkPtr pb;
    DriverInitInfoPtr initialInfo;
    DriverFinalInfoPtr finalInfo;
    DriverReplaceInfoPtr replaceInfo;
    DriverSupersededInfoPtr supersededInfo;
};

```

IOCommandID

```

typedef struct OpaqueIOCommandID * IOCommandID;

```

Availability

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared In

Devices.h

UnitNumber

```

typedef UInt16 UnitNumber;

```

Availability

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared In

Devices.h

Constants

chooserID

Defines the value of the `caller` parameter to a Chooser device package.

```
enum {
    chooserID = 1
};
```

chooserInitMsg

Define values of the `message` parameter to a Chooser device package.

```
enum {
    chooserInitMsg = 11,
    newSelMsg = 12,
    fillListMsg = 13,
    getSelMsg = 14,
    selectMsg = 15,
    deselectMsg = 16,
    terminateMsg = 17,
    buttonMsg = 19
};
```

Constants

`chooserInitMsg`

The user selected this device package.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

`newSelMsg`

The user made new device selections.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

`fillListMsg`

Fill the device list with choices.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

`getSelMsg`

Mark one or more choices as selected.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

`selectMsg`

The user made a selection.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

deselectMsg

The user canceled a selection.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

terminateMsg

Allows device package to clean up.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

buttonMsg

The user selected a button.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

dNeedLockMask

```
enum {
    dNeedLockMask = 0x4000,
    dNeedTimeMask = 0x2000,
    dNeedGoodByeMask = 0x1000,
    dStatEnableMask = 0x0800,
    dCtlEnableMask = 0x0400,
    dWritEnableMask = 0x0200,
    dReadEnableMask = 0x0100
};
```

Constants

dNeedLockMask

Set if driver must be locked in memory as soon as it is opened.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

dNeedTimeMask

Set if driver needs time for performing periodic tasks.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

dNeedGoodByeMask

Set if driver needs to be called before the application heap is initialized.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

dStatEnableMask

Set if driver responds to status requests.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

`dCtlEnableMask`

Set if driver responds to control requests.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

`dWriteEnableMask`

Set if driver responds to write requests.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

`dReadEnableMask`

Set if driver responds to read requests.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

Driver Header and Device Control Entry Flags

Define flags used in the driver header and device control entry.

```
enum {  
    dReadEnable = 0,  
    dWriteEnable = 1,  
    dCtlEnable = 2,  
    dStatEnable = 3,  
    dNeedGoodBye = 4,  
    dNeedTime = 5,  
    dNeedLock = 6  
};
```

Constants

`dReadEnable`

Set if driver responds to read requests.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

`dWriteEnable`

Set if driver responds to write requests.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

`dCtlEnable`

Set if driver responds to control requests.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

`dStatEnable`

Set if driver responds to status requests.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

dNeedGoodBye

Set if driver needs time for performing periodic tasks.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

dNeedTime

Set if driver needs time for performing periodic tasks.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

dNeedLock

Set if driver must be locked in memory as soon as it is opened.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

dVMImmuneBit

Define run-time flags used in the device control entry.

```
enum {
    dVMImmuneBit = 0,
    dOpened = 5,
    dRAMBased = 6,
    drvrActive = 7
};
```

Constants

dVMImmuneBit

Driver does not need VM protection.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

dOpened

Driver is open.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

dRAMBased

dCtlDriver is a handle (1) or pointer (0).

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

drvrActive

Driver is currently processing a request.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

dVMIImmuneMask

```
enum {
    dVMIImmuneMask = 0x0001,
    d0penedMask = 0x0020,
    dRAMBasedMask = 0x0040,
    drvrActiveMask = 0x0080
};
```

Constants

dVMIImmuneMask

Driver does not need VM protection.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

d0penedMask

Driver is open.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

dRAMBasedMask

`dCtlDriver` is a handle (1) or pointer (0).

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

drvrActiveMask

Driver is currently processing a request.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

goodbye

Define control codes for `DeskAccessories`.

```
enum {
    goodbye = -1,
    killCode = 1,
    accEvent = 64,
    accRun = 65,
    accCursor = 66,
    accMenu = 67,
    accUndo = 68,
    accCut = 70,
    accCopy = 71,
    accPaste = 72,
    accClear = 73
};
```

Constants

goodbye

Heap being reinitialized.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

killCode

KillIO requested.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

accEvent

Handle an event.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

accRun

Time for periodic action.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

accCursor

Change cursor shape.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

accMenu

Handle menu item.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

accUndo

Handle undo command.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

accCut

Handle cut command.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

accCopy

Handle copy command.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

accPaste

Handle paste command.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

accClear

Handle clear command.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

initMsg

Define values of the message parameter to a Monitor 'mntn'.

```
enum {
    initMsg = 1,
    okMsg = 2,
    cancelMsg = 3,
    hitMsg = 4,
    nulMsg = 5,
    updateMsg = 6,
    activateMsg = 7,
    deactivateMsg = 8,
    keyEvtMsg = 9,
    superMsg = 10,
    normalMsg = 11,
    startupMsg = 12
};
```

Constants

initMsg

Initialization.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

okMsg

User clicked OK button.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

cancelMsg

User clicked Cancel button.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

hitMsg

User clicked control in Options dialog.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

nulMsg

Periodic event.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

updateMsg

Update event.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

activateMsg

Not used.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

`deactivateMsg`

Not used.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

`keyEvtMsg`

Keyboard event.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

`superMsg`

Show superuser controls.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

`normalMsg`

Show only normal controls.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

`startupMsg`

Code has been loaded.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

Miscellaneous Device Manager constants

```
enum {
    ioInProgress = 1,
    aRdCmd = 2,
    aWrCmd = 3,
    asyncTrpBit = 10,
    noQueueBit = 9
};
```

Constants

`ioInProgress`

Predefined value of `ioResult` while I/O is pending.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

`aRdCmd`

Low byte of `ioTrap` for Read calls.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

`aWrCmd`

Low byte of `ioTrap` for Write calls.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

asyncTrpBit

Trap word modifier.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

noQueueBit

Trap word modifier.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

kOpenCommand

```
enum {
    kOpenCommand = 0,
    kCloseCommand = 1,
    kReadCommand = 2,
    kWriteCommand = 3,
    kControlCommand = 4,
    kStatusCommand = 5,
    kKillIOCommand = 6,
    kInitializeCommand = 7,
    kFinalizeCommand = 8,
    kReplaceCommand = 9,
    kSupersededCommand = 10,
    kSuspendCommand = 11,
    kResumeCommand = 12,
    kPowerManagementCommand = 13
};
```

Constants

kOpenCommand

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

kCloseCommand

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

kReadCommand

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

kWriteCommand

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

kControlCommand

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

kStatusCommand

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

`kKillIOCommand`

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

`kInitializeCommand`

Init driver and device.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

`kFinalizeCommand`

Shutdown driver and device.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

`kReplaceCommand`

Replace an old driver.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

`kSupersededCommand`

Prepare to be replaced by a new driver.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

`kSuspendCommand`

Prepare driver to go to sleep.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

`kResumeCommand`

Wake up sleeping driver.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

`kPowerManagementCommand`

Power management command, supersedes `kSuspendCommand` and `kResumeCommand`.

Available in Mac OS X v10.0 through Mac OS X v10.4.

Declared in `Devices.h`.

Result Codes

The table below shows the most common result codes returned by the Device Manager.

Result Code	Value	Description
<code>controlErr</code>	-17	Driver does not respond to this control request Available in Mac OS X v10.0 and later.

Result Code	Value	Description
statusErr	-18	Driver does not respond to this status request Available in Mac OS X v10.0 and later.
readErr	-19	Driver does not respond to read requests Available in Mac OS X v10.0 and later.
writErr	-20	Driver does not respond to write requests Available in Mac OS X v10.0 and later.
badUnitErr	-21	Driver reference number does not match unit table Available in Mac OS X v10.0 and later.
unitEmptyErr	-22	Driver reference number specifies a nil handle in unit table Available in Mac OS X v10.0 and later.
openErr	-23	Requested read/write permission does not match driver's open permission Available in Mac OS X v10.0 and later.
closErr	-24	Driver unable to complete close request Available in Mac OS X v10.0 and later.
dRemovErr	-25	Tried to remove an open driver Available in Mac OS X v10.0 and later.
dInstErr	-26	DrvInstAll couldn't find driver in resources Available in Mac OS X v10.0 and later.
abortErr	-27	IO call aborted by KillIO Available in Mac OS X v10.0 and later.
iIOAbort	-27	Available in Mac OS X v10.0 and later.
iIOAbortErr	-27	IO abort error (Printing Manager) Available in Mac OS X v10.0 and later.
notOpenErr	-28	Driver not open Available in Mac OS X v10.0 and later.
unitTblFullErr	-29	Unit table has no more entries Available in Mac OS X v10.0 and later.
dceExtErr	-30	dce extension error Available in Mac OS X v10.0 and later.

Result Code	Value	Description
offLinErr	-65	r/w requested for an off-line drive Available in Mac OS X v10.0 and later.
fontSubErr	-66	Font substitution occurred Available in Mac OS X v10.0 and later.
noNybErr	-66	Couldn't find 5 nybbles in 200 tries Available in Mac OS X v10.0 and later.
noAdrMkErr	-67	Couldn't find valid addr mark Available in Mac OS X v10.0 and later.
dataVerErr	-68	Read verify compare failed Available in Mac OS X v10.0 and later.
badCksmErr	-69	addr mark checksum didn't check Available in Mac OS X v10.0 and later.
badBtSlpErr	-70	bad addr mark bit slip nibbles Available in Mac OS X v10.0 and later.
noDtaMkErr	-71	couldn't find a data mark header Available in Mac OS X v10.0 and later.
badDCKsum	-72	bad data mark checksum Available in Mac OS X v10.0 and later.
badDBtSlp	-73	bad data mark bit slip nibbles Available in Mac OS X v10.0 and later.
wrUnderrun	-74	write underrun occurred Available in Mac OS X v10.0 and later.
cantStepErr	-75	step handshake failed Available in Mac OS X v10.0 and later.
tk0BadErr	-76	track 0 detect doesn't change Available in Mac OS X v10.0 and later.
initIWMErr	-77	unable to initialize IWM Available in Mac OS X v10.0 and later.
twoSideErr	-78	tried to read 2nd side on a 1-sided drive Available in Mac OS X v10.0 and later.

Result Code	Value	Description
spdAdjErr	-79	unable to correctly adjust disk speed Available in Mac OS X v10.0 and later.
seekErr	-80	track number wrong on address mark Available in Mac OS X v10.0 and later.
sectNFErr	-81	sector number never found on a track Available in Mac OS X v10.0 and later.
fmt1Err	-82	can't find sector 0 after track format Available in Mac OS X v10.0 and later.
fmt2Err	-83	can't get enough sync Available in Mac OS X v10.0 and later.
clkRdErr	-85	unable to read same clock value twice Available in Mac OS X v10.0 and later.
clkWrErr	-86	time written did not verify Available in Mac OS X v10.0 and later.
prWrErr	-87	parameter ram written didn't read-verify Available in Mac OS X v10.0 and later.
prInitErr	-88	InitUtil found the parameter ram uninitialized Available in Mac OS X v10.0 and later.
rcvrErr	-89	SCC receiver error (framing; parity; OR) Available in Mac OS X v10.0 and later.
breakRecd	-90	Break received (SCC) Available in Mac OS X v10.0 and later.
ddpSktErr	-91	error in socket number Available in Mac OS X v10.0 and later.
eMultiErr	-91	Multicast address error ddpSktErr Available in Mac OS X v10.0 and later.
ddpLenErr	-92	data length too big Available in Mac OS X v10.0 and later.
eLenErr	-92	Length error ddpLenErr Available in Mac OS X v10.0 and later.

Result Code	Value	Description
noBridgeErr	-93	no network bridge for non-local send Available in Mac OS X v10.0 and later.
lapProtErr	-94	error in attaching/detaching protocol Available in Mac OS X v10.0 and later.
excessCollsns	-95	excessive collisions on write Available in Mac OS X v10.0 and later.
portNotPwr	-96	serial port not currently powered Available in Mac OS X v10.0 and later.
portInUse	-97	driver Open error code (port is in use) Available in Mac OS X v10.0 and later.
portNotCf	-98	driver Open error code (parameter RAM not configured for this connection) Available in Mac OS X v10.0 and later.
nrLockedErr	-2536	Available in Mac OS X v10.0 and later.
nrNotEnoughMemoryErr	-2537	Available in Mac OS X v10.0 and later.
nrInvalidNodeErr	-2538	Available in Mac OS X v10.0 and later.
nrNotFoundErr	-2539	Available in Mac OS X v10.0 and later.
nrNotCreatedErr	-2540	Available in Mac OS X v10.0 and later.
nrNameErr	-2541	Available in Mac OS X v10.0 and later.
nrNotSlotDeviceErr	-2542	Available in Mac OS X v10.0 and later.
nrDataTruncatedErr	-2543	Available in Mac OS X v10.0 and later.
nrPowerErr	-2544	Available in Mac OS X v10.0 and later.
nrPowerSwitchAbortErr	-2545	Available in Mac OS X v10.0 and later.
nrTypeMismatchErr	-2546	Available in Mac OS X v10.0 and later.
nrNotModifiedErr	-2547	Available in Mac OS X v10.0 and later.
nrOverrunErr	-2548	Available in Mac OS X v10.0 and later.
nrResultCodeBase	-2549	Available in Mac OS X v10.0 and later.
nrPathNotFound	-2550	a path component lookup failed Available in Mac OS X v10.0 and later.

Result Code	Value	Description
nrPathBufferTooSmall	-2551	buffer for path is too small Available in Mac OS X v10.0 and later.
nrInvalidEntryIterationOp	-2552	invalid entry iteration operation Available in Mac OS X v10.0 and later.
nrPropertyAlreadyExists	-2553	property already exists Available in Mac OS X v10.0 and later.
nrIterationDone	-2554	iteration operation is done Available in Mac OS X v10.0 and later.
nrExitedIteratorScope	-2555	outer scope of iterator was exited Available in Mac OS X v10.0 and later.
nrTransactionAborted	-2556	transaction was aborted Available in Mac OS X v10.0 and later.
nrCallNotSupported	-2557	This call is not available or supported on this machine Available in Mac OS X v10.0 and later.

Deprecated Device Manager Reference (Legacy) Functions

A function identified as deprecated has been superseded and may become unsupported in the future.

Deprecated in Mac OS X v10.5

PBCloseAsync

Closes an open file. (Deprecated in Mac OS X v10.5. Use `PBCloseForkAsync` instead.)

```
OSErr PBCloseAsync (
    ParmBlkPtr paramBlock
);
```

Parameters

paramBlock

A pointer to a basic File Manager parameter block.

Return Value

A result code. See “[Device Manager Result Codes](#)” (page 19).

Discussion

The relevant fields of the parameter block are:

- `ioCompletion` On input, a pointer to a completion routine.
- `ioResult` On output, the result code of the function.
- `ioRefNum` On input, a file reference number to the file to close.

The `PBCloseAsync` function writes the contents of the access path buffer specified by the `ioRefNum` field to the volume and removes the access path.

Special Considerations

Some information stored on the volume won't be updated until `PBFlushVolAsync` is called.

Do not call `PBCloseAsync` with a file reference number of a file that has already been closed. Attempting to close the same file twice may result in loss of data on a volume. .

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Not available to 64-bit applications.

Declared In

`Files.h`

PBCloseSync

Closes an open file. (Deprecated in Mac OS X v10.5. Use `PBCloseForkSync` instead.)

```
OSErr PBCloseSync (
    ParmBlkPtr paramBlock
);
```

Parameters

paramBlock

A pointer to a basic File Manager parameter block.

Return Value

A result code. See “[Device Manager Result Codes](#)” (page 19).

Discussion

The relevant fields of the parameter block are:

- `ioCompletion` On input, a pointer to a completion routine.
- `ioResult` On output, the result code of the function.
- `ioRefNum` On input, a file reference number to the file to close.

The `PBCloseSync` function writes the contents of the access path buffer specified by the `ioRefNum` field to the volume and removes the access path.

Special Considerations

Some information stored on the volume won't be updated until `PBFlushVolSync` is called.

Do not call `PBCloseSync` with a file reference number of a file that has already been closed. Attempting to close the same file twice may result in loss of data on a volume. .

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Not available to 64-bit applications.

Declared In

`Files.h`

PBReadAsync

Reads any number of bytes from an open file. (Deprecated in Mac OS X v10.5. Use `PBReadForkAsync` instead.)

```
OSErr PBReadAsync (
    ParmBlkPtr paramBlock
);
```

Parameters

paramBlock

A pointer to a basic File Manager parameter block.

Return Value

A result code. See “[Device Manager Result Codes](#)” (page 19).

Discussion

The relevant fields of the parameter block are:

- `ioCompletion` On input, a pointer to a completion routine.
- `ioResult` On output, the result code of the function.
- `ioRefNum` On input, a file reference number for an open file to be read.
- `ioBuffer` On input, a pointer to a data buffer into which the bytes are read.
- `ioReqCount` On input, the number of bytes requested. The value that you pass in this field should be greater than zero.
- `ioActCount` On output, the number of bytes actually read.
- `ioPosMode` On input, the positioning mode.
- `ioPosOffset` On input, the positioning offset. On output, the new position of the mark.

This function attempts to read `ioReqCount` bytes from the open file whose access path is specified in the `ioRefNum` field and transfer them to the data buffer pointed to by the `ioBuffer` field. The position of the mark is specified by `ioPosMode` and `ioPosOffset`. If your application tries to read past the logical end-of-file, `PBReadAsync` reads the data, moves the mark to the end-of-file, and returns `eofErr` as its function result. Otherwise, `PBReadAsync` moves the file mark to the byte following the last byte read and returns `noErr`.

You can specify that `PBReadAsync` read the file data 1 byte at a time until the requested number of bytes have been read or until the end-of-file is reached. To do so, set bit 7 of the `ioPosMode` field. Similarly, you can specify that `PBReadAsync` should stop reading data when it reaches an application-defined newline character. To do so, place the ASCII code of that character into the high-order byte of the `ioPosMode` field; you must also set bit 7 of that field to enable newline mode.

When reading data in newline mode, `PBReadAsync` returns the newline character as part of the data read and sets `ioActCount` to the actual number of bytes placed into the buffer (which includes the newline character).

Special Considerations

In Mac OS 8 and 9, it is possible to call `PBReadAsync` with a value of 0 in the `ioReqCount` field. In Mac OS X, `PBReadAsync` returns a `paramErr` error if the value in the `ioReqCount` field is 0.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Not available to 64-bit applications.

Declared In

`Files.h`

PBReadSync

Reads any number of bytes from an open file. (Deprecated in Mac OS X v10.5. Use `PBReadForkSync` instead.)

Deprecated Device Manager Reference (Legacy) Functions

```
OSErr PBReadSync (
    ParmBlkPtr paramBlock
);
```

Parameters*paramBlock*

A pointer to a basic File Manager parameter block.

Return Value

A result code. See “[Device Manager Result Codes](#)” (page 19).

Discussion

The relevant fields of the parameter block are:

- `ioCompletion` On input, a pointer to a completion routine.
- `ioResult` On output, the result code of the function.
- `ioRefNum` On input, a file reference number for an open file to be read.
- `ioBuffer` On input, a pointer to a data buffer into which the bytes are read.
- `ioReqCount` On input, the number of bytes requested. The value that you pass in this field should be greater than zero.
- `ioActCount` On output, the number of bytes actually read.
- `ioPosMode` On input, the positioning mode.
- `ioPosOffset` On input, the positioning offset. On output, the new position of the mark.

This function attempts to read `ioReqCount` bytes from the open file whose access path is specified in the `ioRefNum` field and transfer them to the data buffer pointed to by the `ioBuffer` field. The position of the mark is specified by `ioPosMode` and `ioPosOffset`. If your application tries to read past the logical end-of-file, `PBReadSync` reads the data, moves the mark to the end-of-file, and returns `eofErr` as its function result. Otherwise, `PBReadSync` moves the file mark to the byte following the last byte read and returns `noErr`.

You can specify that `PBReadSync` read the file data 1 byte at a time until the requested number of bytes have been read or until the end-of-file is reached. To do so, set bit 7 of the `ioPosMode` field. Similarly, you can specify that `PBReadSync` should stop reading data when it reaches an application-defined newline character. To do so, place the ASCII code of that character into the high-order byte of the `ioPosMode` field; you must also set bit 7 of that field to enable newline mode.

When reading data in newline mode, `PBReadSync` returns the newline character as part of the data read and sets `ioActCount` to the actual number of bytes placed into the buffer (which includes the newline character).

Special Considerations

In Mac OS 8 and 9, it is possible to call `PBReadSync` with a value of 0 in the `ioReqCount` field. In Mac OS X, `PBReadSync` returns a `paramErr` error if the value in the `ioReqCount` field is 0.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Not available to 64-bit applications.

Declared In

`Files.h`

PBWaitIOComplete

Keeps the system idle until either an interrupt occurs or the specified timeout value is reached. (Deprecated in Mac OS X v10.5. There is no replacement function.)

```
OSErr PBWaitIOComplete (
    ParmBlkPtr paramBlock,
    Duration timeout
);
```

Parameters

paramBlock

A pointer to a basic File Manager parameter block.

timeout

The maximum length of time you want the system to be kept idle.

Return Value

A result code. See “Device Manager Result Codes” (page 19). If the timeout value is reached, returns `kMPTimeoutErr`.

Special Considerations

This function is not implemented in Mac OS X.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Not available to 64-bit applications.

Declared In

`Files.h`

PBWriteAsync

Writes any number of bytes to an open file. (Deprecated in Mac OS X v10.5. Use `PBWriteForkAsync` instead.)

```
OSErr PBWriteAsync (
    ParmBlkPtr paramBlock
);
```

Parameters

paramBlock

A pointer to a basic File Manager parameter block.

Return Value

A result code. See “Device Manager Result Codes” (page 19).

Discussion

The relevant fields of the parameter block are:

- `ioCompletion` On input, a pointer to a completion routine.
- `ioResult` On output, the result code of the function.
- `ioRefNum` On input, a file reference number for the open file to which to write.
- `ioBuffer` On input, a pointer to a data buffer containing the bytes to write.

Deprecated Device Manager Reference (Legacy) Functions

- `ioReqCount` On input, the number of bytes requested.
- `ioActCount` On output, the number of bytes actually written.
- `ioPosMode` On input, the positioning mode.
- `ioPosOffset` On input, the positioning offset. On output, the new position of the mark.

The `PBWriteAsync` function takes `ioReqCount` bytes from the buffer pointed to by `ioBuffer` and attempts to write them to the open file whose access path is specified by `ioRefNum`. The position of the mark is specified by `ioPosMode` and `ioPosOffset`. If the write operation completes successfully, `PBWriteAsync` moves the file mark to the byte following the last byte written and returns `noErr`.

If you try to write past the logical end-of-file, `PBWriteAsync` moves the logical end-of-file. If you try to write past the physical end-of-file, `PBWriteAsync` adds one or more clumps to the file and moves the physical end-of-file accordingly.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Not available to 64-bit applications.

Declared In

`Files.h`

PBWriteSync

Writes any number of bytes to an open file. (Deprecated in Mac OS X v10.5. Use `PBWriteForkSync` instead.)

```
OSErr PBWriteSync (
    ParmBlkPtr paramBlock
);
```

Parameters

paramBlock

A pointer to a basic File Manager parameter block.

Return Value

A result code. See “[Device Manager Result Codes](#)” (page 19).

Discussion

The relevant fields of the parameter block are:

- `ioCompletion` On input, a pointer to a completion routine.
- `ioResult` On output, the result code of the function.
- `ioRefNum` On input, a file reference number for the open file to which to write.
- `ioBuffer` On input, a pointer to a data buffer containing the bytes to write.
- `ioReqCount` On input, the number of bytes requested.
- `ioActCount` On output, the number of bytes actually written.
- `ioPosMode` On input, the positioning mode.
- `ioPosOffset` On input, the positioning offset. On output, the new position of the mark.

Deprecated Device Manager Reference (Legacy) Functions

The `PBWriteSync` function takes `ioReqCount` bytes from the buffer pointed to by `ioBuffer` and attempts to write them to the open file whose access path is specified by `ioRefNum`. The position of the mark is specified by `ioPosMode` and `ioPosOffset`. If the write operation completes successfully, `PBWriteSync` moves the file mark to the byte following the last byte written and returns `noErr`.

If you try to write past the logical end-of-file, `PBWriteSync` moves the logical end-of-file. If you try to write past the physical end-of-file, `PBWriteSync` adds one or more clumps to the file and moves the physical end-of-file accordingly.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

Not available to 64-bit applications.

Declared In

`Files.h`

Document Revision History

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

Date	Notes
2007-07-10	Moved to the legacy area of the ADC Reference Library.
2003-04-01	Added documentation for the function <code>PBWaitIOComplete</code> .
2003-01-01	Updated formatting

REVISION HISTORY

Document Revision History

Index

A

abortErr [constant](#) 20
accClear [constant](#) 15
accCopy [constant](#) 15
accCursor [constant](#) 15
accCut [constant](#) 15
accEvent [constant](#) 15
accMenu [constant](#) 15
accPaste [constant](#) 15
accRun [constant](#) 15
accUndo [constant](#) 15
activateMsg [constant](#) 16
AddressSpaceID [data type](#) 5
aRdCmd [constant](#) 17
asyncTrpBit [constant](#) 18
AuxDCE [structure](#) 6
aWrCmd [constant](#) 17

B

badBtSlpErr [constant](#) 21
badCksmErr [constant](#) 21
badDBtSlp [constant](#) 21
badDcksum [constant](#) 21
badUnitErr [constant](#) 20
breakRecd [constant](#) 22
buttonMsg [constant](#) 11

C

cancelMsg [constant](#) 16
cantStepErr [constant](#) 21
chooserID 10
chooserInitMsg 10
chooserInitMsg [constant](#) 10
clkRdErr [constant](#) 22
clkWrErr [constant](#) 22

closErr [constant](#) 20
controlErr [constant](#) 19

D

dataVerErr [constant](#) 21
dceExtErr [constant](#) 20
dCtlEnable [constant](#) 12
dCtlEnableMask [constant](#) 12
DctlEntry [structure](#) 6
ddpLenErr [constant](#) 22
ddpSktErr [constant](#) 22
deactivateMsg [constant](#) 17
deselectMsg [constant](#) 11
dInstErr [constant](#) 20
dNeedGoodBye [constant](#) 13
dNeedGoodByeMask [constant](#) 11
dNeedLock [constant](#) 13
dNeedLockMask 11
dNeedLockMask [constant](#) 11
dNeedTime [constant](#) 13
dNeedTimeMask [constant](#) 11
dOpened [constant](#) 13
dOpenedMask [constant](#) 14
dRAMBased [constant](#) 13
dRAMBasedMask [constant](#) 14
dReadEnable [constant](#) 12
dReadEnableMask [constant](#) 12
dRemovErr [constant](#) 20
Driver Header and Device Control Entry Flags 12
DriverFinalInfo [structure](#) 6
DriverInitInfo [structure](#) 6
DriverOpenCount [data type](#) 7
DriverRefNum [data type](#) 7
DriverReplaceInfo [data type](#) 8
DriverSupersededInfo [data type](#) 8
drvActive [constant](#) 13
drvActiveMask [constant](#) 14
DRVRHeader [structure](#) 8
dStatEnable [constant](#) 12
dStatEnableMask [constant](#) 11

dVMIImmuneBit [13](#)
 dVMIImmuneBit constant [13](#)
 dVMIImmuneMask [14](#)
 dVMIImmuneMask constant [14](#)
 dWritEnable constant [12](#)
 dWritEnableMask constant [12](#)

E

eLenErr constant [22](#)
 eMultiErr constant [22](#)
 excessCollsns constant [23](#)

F

FileBasedDriverDetailed structure [9](#)
 FileBasedDriverRecord structure [9](#)
 fillListMsg constant [10](#)
 fmt1Err constant [22](#)
 fmt2Err constant [22](#)
 fontSubErr constant [21](#)

G

getSelMsg constant [10](#)
 goodbye [14](#)
 goodbye constant [14](#)

H

hitMsg constant [16](#)

I

iIOAbort constant [20](#)
 iIOAbortErr constant [20](#)
 initIWMErr constant [21](#)
 initMsg [16](#)
 initMsg constant [16](#)
 IOCommandContents structure [9](#)
 IOCommandID data type [9](#)
 ioInProgress constant [17](#)

K

kCloseCommand constant [18](#)
 kControlCommand constant [18](#)
 keyEvtMsg constant [17](#)
 kFinalizeCommand constant [19](#)
 killCode constant [15](#)
 kInitializeCommand constant [19](#)
 kKillIOCommand constant [19](#)
 kOpenCommand [18](#)
 kOpenCommand constant [18](#)
 kPowerManagementCommand constant [19](#)
 kReadCommand constant [18](#)
 kReplaceCommand constant [19](#)
 kResumeCommand constant [19](#)
 kStatusCommand constant [18](#)
 kSupersededCommand constant [19](#)
 kSuspendCommand constant [19](#)
 kWriteCommand constant [18](#)

L

lapProtErr constant [23](#)

M

Miscellaneous Device Manager constants [17](#)

N

newSelMsg constant [10](#)
 noAdrMkErr constant [21](#)
 noBridgeErr constant [23](#)
 noDtaMkErr constant [21](#)
 noNybErr constant [21](#)
 noQueueBit constant [18](#)
 normalMsg constant [17](#)
 notOpenErr constant [20](#)
 nrCallNotSupported constant [24](#)
 nrDataTruncatedErr constant [23](#)
 nrExitedIteratorScope constant [24](#)
 nrInvalidEntryIterationOp constant [24](#)
 nrInvalidNodeErr constant [23](#)
 nrIterationDone constant [24](#)
 nrLockedErr constant [23](#)
 nrNameErr constant [23](#)
 nrNotCreatedErr constant [23](#)
 nrNotEnoughMemoryErr constant [23](#)

nrNotFoundErr **constant** [23](#)
 nrNotModifiedErr **constant** [23](#)
 nrNotSlotDeviceErr **constant** [23](#)
 nrOverrunErr **constant** [23](#)
 nrPathBufferTooSmall **constant** [24](#)
 nrPathNotFound **constant** [23](#)
 nrPowerErr **constant** [23](#)
 nrPowerSwitchAbortErr **constant** [23](#)
 nrPropertyAlreadyExists **constant** [24](#)
 nrResultCodeBase **constant** [23](#)
 nrTransactionAborted **constant** [24](#)
 nrTypeMismatchErr **constant** [23](#)
 nulMsg **constant** [16](#)

O

offLinErr **constant** [21](#)
 okMsg **constant** [16](#)
 openErr **constant** [20](#)

P

PBCloseAsync **function** (Deprecated in Mac OS X v10.5)
[25](#)
 PBCloseSync **function** (Deprecated in Mac OS X v10.5)
[26](#)
 PBReadAsync **function** (Deprecated in Mac OS X v10.5)
[26](#)
 PBReadSync **function** (Deprecated in Mac OS X v10.5) [27](#)
 PBWaitIOComplete **function** (Deprecated in Mac OS X
 v10.5) [29](#)
 PBWriteAsync **function** (Deprecated in Mac OS X v10.5)
[29](#)
 PBWriteSync **function** (Deprecated in Mac OS X v10.5)
[30](#)
 portInUse **constant** [23](#)
 portNotCf **constant** [23](#)
 portNotPwr **constant** [23](#)
 prInitErr **constant** [22](#)
 prWrErr **constant** [22](#)

R

rcvrErr **constant** [22](#)
 readErr **constant** [20](#)

S

sectNFErr **constant** [22](#)
 seekErr **constant** [22](#)
 selectMsg **constant** [10](#)
 spdAdjErr **constant** [22](#)
 startupMsg **constant** [17](#)
 statusErr **constant** [20](#)
 superMsg **constant** [17](#)

T

terminateMsg **constant** [11](#)
 tk0BadErr **constant** [21](#)
 twoSideErr **constant** [21](#)

U

unitEmptyErr **constant** [20](#)
 UnitNumber **data type** [9](#)
 unitTblFullErr **constant** [20](#)
 updateMsg **constant** [16](#)

W

writErr **constant** [20](#)
 wrUnderrun **constant** [21](#)