
Printing Plug-in Interfaces Reference

[Printing > Carbon](#)



2003-07-31



Apple Inc.
© 2002, 2003 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, AppleTalk, Carbon, Mac, Mac OS, and Pages are trademarks of Apple Inc., registered in the United States and other countries.

DEC is a trademark of Digital Equipment Corporation.

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

Printing Plug-in Interfaces Reference 7

Overview	7
Functions	7
PMCreateLocalizedPaperSizeCFString	7
PMCreatePaperSizeCFString	8
Callbacks by Task	9
Printing Plug-in Callbacks	9
Printing Dialog Extension Callbacks	9
Printer Module Callbacks	9
Printer Browser Module Callbacks	11
I/O Module Callbacks	11
Callbacks	12
GetConnInfoProcPtr	12
PMBeginJobProcPtr	13
PMCancelJobProcPtr	13
PMCOMAddRefProcPtr	13
PMCOMQueryInterfaceProcPtr	14
PMCOMReleaseProcPtr	14
PMCreatePrinterBrowserModuleInfoProcPtr	14
PMCreatePrinterTicketsProcPtr	15
PMCreatePrintingDialogExtensionsPathsProcPtr	15
PMEndJobProcPtr	15
PMImageAccessProcPtr	16
PMInitializeProcPtr	16
PMIOCloseProcPtr	16
PMIOGetAttributeProcPtr	17
PMIOModuleCloseProcPtr	17
PMIOModuleGetAttributeProcPtr	18
PMIOModuleGetConnectionInfoProcPtr	18
PMIOModuleInitializeProcPtr	18
PMIOModuleOpenProcPtr	19
PMIOModuleReadProcPtr	19
PMIOModuleSetAttributeProcPtr	19
PMIOModuleStatusProcPtr	20
PMIOModuleTerminateProcPtr	20
PMIOModuleWriteProcPtr	20
PMIOOpenProcPtr	21
PMIOReadProcPtr	21
PMIOSetAttributeProcPtr	22
PMIOStatusProcPtr	22
PMIOWriteProcPtr	23

PMJobStreamGetNextBandProcPtr	23
PMJobStreamGetPosProcPtr	24
PMJobStreamOpenProcPtr	24
PMJobStreamReadWriteProcPtr	24
PMJobStreamSetPosProcPtr	25
PMNotificationProcPtr	25
PMPDECloseProcPtr	26
PMPDEGetSummaryTextProcPtr	26
PMPDEInitializeProcPtr	27
PMPDEOpenProcPtr	29
PMPDEPrologueProcPtr	29
PMPDESyncProcPtr	31
PMPDETerminateProcPtr	32
PMPluginGetAPIVersionProcPtr	33
PMPluginReleaseProcPtr	33
PMPluginRetainProcPtr	34
PMPrBrowserAPIVersionProcPtr	34
PMPrBrowserGetLookupSpecProcPtr	34
PMPrBrowserGetSelectedPrintersProcPtr	35
PMPrBrowserInitializeProcPtr	35
PMPrBrowserPrologueProcPtr	36
PMPrBrowserResizeProcPtr	36
PMPrBrowserSelectionStatusProcPtr	37
PMPrBrowserSyncProcPtr	37
PMPrBrowserSyncRequestProcPtr	38
PMPrBrowserTerminateProcPtr	38
PMPrBrowserWorksetPrintersProcPtr	38
PMPrintJobProcPtr	39
PMPrintPageProcPtr	39
PMTerminateProcPtr	40
Data Types	40
Printing Plug-in Data Types	40
Printing Dialog Extension Data Types	42
Printer Module Data Types	45
Printer Browser Module Data Types	47
I/O Module Data Types	50
Constants	51
PDE Feature Flags	51
PDE Interface Identifier	52
PDE Interface Version	52
PDE Pane Kind Identifiers	53
PDE Ticket Identifiers	54
PDE Type Identifiers	54
I/O Module Interface Version	55
Printer Module Interface Version	55
Printer Module Status Codes	55

CONTENTS

Other Printer Module Constants	55
Print Center Feature Flags	56
Print Center Signatures	56
Result Codes	56

Document Revision History 59

Index 61

Printing Plug-in Interfaces Reference

Framework:	ApplicationServices/ApplicationServices.h, Carbon/Carbon.h
Declared in	PMIOModule.h PMPluginHeader.h PMPrinterBrowsers.h PMPrinterModuleDeprecated.h PMPrintingDialogExtensionsDeprecated.h

Overview

As printer vendors and application developers extend the printing capabilities of their hardware and software products, they need a way to extend the Mac OS X printing system to make new printing features available to their customers. To address this need, Mac OS X has introduced the **printing plug-in** —a component architecture based on Core Foundation Plug-in Services. There are four types of printing plug-ins in Mac OS X:

- **I/O modules** are used by the printing system to communicate with a printer using a standard transport-layer interface, such as AppleTalk or TCP/IP.
- **Printer browsers** provide a way for people to discover available local and network printers.
- **Printer modules** are used by the printing system to convert the graphics content in a print job for output to a specific printer or family of printers.
- **Printing dialog extensions** provide a way for people to view and change the settings for a set of related printing features. The user interface of a printing dialog extension is a pane in one of the printing dialogs.

This reference document is relevant for anyone writing a plug-in that provides support for printing.

If you're writing a printing dialog extension, you should refer to this document as you implement the required callback functions. For conceptual information about printing dialog extensions, see *Extending Printing Dialogs*.

Functions

PMCreateLocalizedPaperSizeCFString

Returns a text description of the physical dimensions of the paper specified in a paper information ticket. The unit of measure is based on the current language and script system.

```

CFStringRef PMCreateLocalizedPaperSizeCFString (
    PMTicketRef listofPaperTickets,
    UInt16 paperToUse
);

```

Parameters

listofPaperTickets

A ticket that contains a list of paper information tickets. Typically you would obtain this list ticket from a printer module template ticket.

paperToUse

A one-based index that specifies an entry in a list ticket. Upon entering, this parameter should specify a valid entry in *listofPaperTickets*.

Return Value

A text description of the physical dimensions of the paper specified in a paper information ticket. Numeric values are localized to English or metric measure, based on the current language and script system. The caller assumes ownership of the string and is responsible for releasing it. The return value `NULL` indicates that the function failed to create the string.

Discussion

The name `PMCreateLocalizedPaperSizeCFString` suggests that this function does some additional localization, but that is misleading. This function takes an array of paper information tickets and an index, finds the desired paper information ticket, and simply calls `PMCreatePaperSizeCFString` to get the physical paper size.

Availability

Available in Mac OS X version 10.0 and later.

Not available to 64-bit applications.

Declared In

`PMPrintingDialogExtensionsDeprecated.h`

PMCreatePaperSizeCFString

Returns a text description of the physical dimensions of the paper specified in a paper information ticket. The unit of measure is based on the current language and script system.

```

CFStringRef PMCreatePaperSizeCFString (
    PMTicketRef selectedPaper
);

```

Parameters

selectedPaper

A paper information ticket. Typically you would obtain this ticket from a page format ticket, or possibly from a printer module template ticket.

Return Value

A text description of the physical dimensions of the paper specified in a paper information ticket. Numeric values are localized to English or metric measure, based on the current language and script system. The caller assumes ownership of the string, and is responsible for releasing it. The return value `NULL` indicates that the function failed to create the string.

Availability

Available in Mac OS X version 10.0 and later.

Not available to 64-bit applications.

Declared In

`PMPrintingDialogExtensionsDeprecated.h`

Callbacks by Task

Printing Plug-in Callbacks

The three callback functions described in this section are used by the printing system to manage a printing plug-in after it is loaded. Printing plug-ins (except for printer browsers) are required to implement these functions.

[PMPluginRetainProcPtr](#) (page 34)

[PMPluginReleaseProcPtr](#) (page 33)

Defines a pointer to the retain function in a printing plug-in interface. Your custom retain function increments the reference count of an instance of one of these interfaces.

[PMPluginGetAPIVersionProcPtr](#) (page 33)

Printing Dialog Extension Callbacks

The callback functions described in this section must be implemented by all printing dialog extensions.

[PMPDEPrologueProcPtr](#) (page 29)

[PMPDEInitializeProcPtr](#) (page 27)

[PMPDESyncProcPtr](#) (page 31)

[PMPDEGetSummaryTextProcPtr](#) (page 26)

[PMPDEOpenProcPtr](#) (page 29)

[PMPDECloseProcPtr](#) (page 26)

[PMPDETerminateProcPtr](#) (page 32)

Printer Module Callbacks

The callback functions described in this section must be implemented by all printer modules.

- [GetConnInfoProcPtr](#) (page 12)
- [PMBeginJobProcPtr](#) (page 13)
- [PMCancelJobProcPtr](#) (page 13)
- [PMCreatePrinterBrowserModuleInfoProcPtr](#) (page 14)
- [PMCreatePrinterTicketsProcPtr](#) (page 15)
- [PMCreatePrintingDialogExtensionsPathsProcPtr](#) (page 15)
- [PMEndJobProcPtr](#) (page 15)
- [PMImageAccessProcPtr](#) (page 16)
- [PMInitializeProcPtr](#) (page 16)
- [PMIOCloseProcPtr](#) (page 16)
- [PMIOGetAttributeProcPtr](#) (page 17)
- [PMIOOpenProcPtr](#) (page 21)
- [PMIOReadProcPtr](#) (page 21)
- [PMIOSetAttributeProcPtr](#) (page 22)
- [PMIOStatusProcPtr](#) (page 22)
- [PMIOWriteProcPtr](#) (page 23)
- [PMJobStreamGetNextBandProcPtr](#) (page 23)
- [PMJobStreamGetPosProcPtr](#) (page 24)
- [PMJobStreamOpenProcPtr](#) (page 24)
- [PMJobStreamReadWriteProcPtr](#) (page 24)
- [PMJobStreamSetPosProcPtr](#) (page 25)
- [PMNotificationProcPtr](#) (page 25)

[PMPrintJobProcPtr](#) (page 39)

[PMPrintPageProcPtr](#) (page 39)

[PMTerminateProcPtr](#) (page 40)

Printer Browser Module Callbacks

The callback functions described in this section must be implemented by all printer browser modules.

[PMCOMAddRefProcPtr](#) (page 13)

[PMCOMQueryInterfaceProcPtr](#) (page 14)

[PMCOMReleaseProcPtr](#) (page 14)

[PMPrBrowserAPIVersionProcPtr](#) (page 34)

[PMPrBrowserGetLookupSpecProcPtr](#) (page 34)

[PMPrBrowserGetSelectedPrintersProcPtr](#) (page 35)

[PMPrBrowserInitializeProcPtr](#) (page 35)

[PMPrBrowserPrologueProcPtr](#) (page 36)

[PMPrBrowserResizeProcPtr](#) (page 36)

[PMPrBrowserSelectionStatusProcPtr](#) (page 37)

[PMPrBrowserSyncProcPtr](#) (page 37)

[PMPrBrowserSyncRequestProcPtr](#) (page 38)

[PMPrBrowserTerminateProcPtr](#) (page 38)

[PMPrBrowserWorksetPrintersProcPtr](#) (page 38)

I/O Module Callbacks

The callback functions described in this section must be implemented by all I/O modules.

[PMIOModuleCloseProcPtr](#) (page 17)

[PMIOModuleGetAttributeProcPtr](#) (page 18)

[PMIOModuleGetConnectionInfoProcPtr](#) (page 18)

[PMIOModuleInitializeProcPtr](#) (page 18)

[PMIOModuleOpenProcPtr](#) (page 19)

[PMIOModuleReadProcPtr](#) (page 19)

[PMIOModuleSetAttributeProcPtr](#) (page 19)

[PMIOModuleStatusProcPtr](#) (page 20)

[PMIOModuleTerminateProcPtr](#) (page 20)

[PMIOModuleWriteProcPtr](#) (page 20)

Callbacks

GetConnInfoProcPtr

```
typedef OSStatus(* GetConnInfoProcPtr) (  
    const void *jobContext,  
    CFStringRef *connectionType,  
    CFStringRef *pbmPath  
);
```

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

```
OSStatus MyGetConnInfoCallback (  
    const void *jobContext,  
    CFStringRef *connectionType,  
    CFStringRef *pbmPath  
);
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

`PMPrinterModuleDeprecated.h`

PMBeginJobProcPtr

```
typedef OSStatus(* PMBeginJobProcPtr) (
    PMContext printerModuleContext,
    const void *jobContext,
    PMTicketRef jobTicket,
    PMTicketRef *converterSetup
);
```

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

```
OSStatus MyPMBeginJobCallback (
    PMContext printerModuleContext,
    const void *jobContext,
    PMTicketRef jobTicket,
    PMTicketRef *converterSetup
);
```

PMCancelJobProcPtr

```
typedef OSStatus(* PMCancelJobProcPtr) (
    PMContext printerModuleContext,
    const void *jobContext
);
```

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

```
OSStatus MyPMCancelJobCallback (
    PMContext printerModuleContext,
    const void *jobContext
);
```

PMCOMAddRefProcPtr

```
typedef UInt32(* PMCOMAddRefProcPtr) (
    void *thisPointer
);
```

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

```
UInt32 MyPMCOMAddRefCallback (
    void *thisPointer
);
```

PMCOMQueryInterfaceProcPtr

```
typedef Sint32(* PMCOMQueryInterfaceProcPtr) (
    void *thisPointer,
    CFUUIDBytes iid,
    void **ppv
);
```

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

```
Sint32 MyPMCOMQueryInterfaceCallback (
    void *thisPointer,
    CFUUIDBytes iid,
    void **ppv
);
```

PMCOMReleaseProcPtr

```
typedef UInt32(* PMCOMReleaseProcPtr) (
    void *thisPointer
);
```

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

```
UInt32 MyPMCOMReleaseCallback (
    void *thisPointer
);
```

PMCreatePrinterBrowserModuleInfoProcPtr

```
typedef OSStatus(* PMCreatePrinterBrowserModuleInfoProcPtr) (
    CFStringRef connectionType,
    CFArrayRef *printerBrowserInfo
);
```

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

```
OSStatus MyPMCreatePrinterBrowserModuleInfoCallback (
    CFStringRef connectionType,
    CFArrayRef *printerBrowserInfo
);
```

PMCreatePrinterTicketsProcPtr

```
typedef OSStatus(* PMCreatePrinterTicketsProcPtr) (
    PMContext printerModuleContext,
    PMTicketRef *printerInfo,
    PMTemplateRef *jobTemplate
);
```

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

```
OSStatus MyPMCreatePrinterTicketsCallback (
    PMContext printerModuleContext,
    PMTicketRef *printerInfo,
    PMTemplateRef *jobTemplate
);
```

PMCreatePrintingDialogExtensionsPathsProcPtr

```
typedef OSStatus(* PMCreatePrintingDialogExtensionsPathsProcPtr) (
    PMContext printerModuleContext,
    CFArrayRef *pdePaths
);
```

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

```
OSStatus MyPMCreatePrintingDialogExtensionsPathsCallback (
    PMContext printerModuleContext,
    CFArrayRef *pdePaths
);
```

PMEndJobProcPtr

```
typedef OSStatus(* PMEndJobProcPtr) (
    PMContext printerModuleContext,
    const void *jobContext
);
```

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

```
OSStatus MyPMEndJobCallback (
    PMContext printerModuleContext,
    const void *jobContext
);
```

PMImageAccessProcPtr

```
typedef OSStatus(* PMImageAccessProcPtr) (
    PMContext printerModuleContext,
    const void *jobContext,
    CFStringRef grafBase,
    PMDrawingCtx drawingCtx,
    PMImageRef imageRef,
    PMImageRef *outImageRefPtr
);
```

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

```
OSStatus MyPMImageAccessCallback (
    PMContext printerModuleContext,
    const void *jobContext,
    CFStringRef grafBase,
    PMDrawingCtx drawingCtx,
    PMImageRef imageRef,
    PMImageRef *outImageRefPtr
);
```

PMInitializeProcPtr

```
typedef OSStatus(* PMInitializeProcPtr) (
    CFDataRef printerAddress,
    const void *jobContext,
    const PMIOProcs *pmIOProcs,
    const PMNotificationProcPtr pmNotificationProc,
    PMContext *printerModuleContext
);
```

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

```
OSStatus MyPMInitializeCallback (
    CFDataRef printerAddress,
    const void *jobContext,
    const PMIOProcs *pmIOProcs,
    const PMNotificationProcPtr pmNotificationProc,
    PMContext *printerModuleContext
);
```

PMIOCloseProcPtr

```
typedef OSStatus(* PMIOCloseProcPtr) (
    const void *jobContext
);
```

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

```
OSStatus MyPMIOCloseCallback (
    const void *jobContext
);
```



```
);
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

PMPrinterModuleDeprecated.h

PMIOGetAttributeProcPtr

```
typedef OSStatus(* PMIOGetAttributeProcPtr) (  
    const void *jobContext,  
    CFStringRef attribute,  
    CTypeRef *result  
);
```

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

```
OSStatus MyPMIOGetAttributeCallback (  
    const void *jobContext,  
    CFStringRef attribute,  
    CTypeRef *result  
);
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

PMPrinterModuleDeprecated.h

PMIOModuleCloseProcPtr

```
typedef OSStatus(* PMIOModuleCloseProcPtr) (  
    IOContext ioModuleContext,  
    Boolean abort  
);
```

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

```
OSStatus MyCallback (  
    IOContext ioModuleContext,  
    Boolean abort  
);
```

PMIOModuleGetAttributeProcPtr

```
typedef OSStatus(* PMIOModuleGetAttributeProcPtr) (
    IOMContext ioModuleContext,
    CFStringRef attribute,
    CFTYPERef *result
);
```

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

```
OSStatus MyCallback (
    IOMContext ioModuleContext,
    CFStringRef attribute,
    CFTYPERef *result
);
```

PMIOModuleGetConnectionInfoProcPtr

```
typedef OSStatus(* PMIOModuleGetConnectionInfoProcPtr) (
    CFStringRef *connectionType,
    CFStringRef *pbmPath
);
```

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

```
OSStatus MyCallback (
    CFStringRef *connectionType,
    CFStringRef *pbmPath
);
```

PMIOModuleInitializeProcPtr

```
typedef OSStatus(* PMIOModuleInitializeProcPtr) (
    CFDataRef printerAddress,
    IOMContext *ioModuleContextPtr
);
```

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

```
OSStatus MyCallback (
    CFDataRef printerAddress,
    IOMContext *ioModuleContextPtr
);
```

PMIOModuleOpenProcPtr

```
typedef OSStatus(* PMIOModuleOpenProcPtr) (
    IOMContext ioModuleContext,
    PMTicketRef jobTicket,
    UInt32 *bufferSize
);
```

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

```
OSStatus MyCallback (
    IOMContext ioModuleContext,
    PMTicketRef jobTicket,
    UInt32 *bufferSize
);
```

PMIOModuleReadProcPtr

```
typedef OSStatus(* PMIOModuleReadProcPtr) (
    IOMContext ioModuleContext,
    Ptr buffer,
    UInt32 *size,
    Boolean *eoj
);
```

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

```
OSStatus MyCallback (
    IOMContext ioModuleContext,
    Ptr buffer,
    UInt32 *size,
    Boolean *eoj
);
```

PMIOModuleSetAttributeProcPtr

```
typedef OSStatus(* PMIOModuleSetAttributeProcPtr) (
    IOMContext ioModuleContext,
    CFStringRef attribute,
    CTypeRef data
);
```

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

```
OSStatus MyCallback (
    IOMContext ioModuleContext,
    CFStringRef attribute,
    CTypeRef data
);
```

PMIOModuleStatusProcPtr

```
typedef OSStatus(* PMIOModuleStatusProcPtr) (
    IOMContext ioModuleContext,
    CFStringRef *status
);
```

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

```
OSStatus MyCallback (
    IOMContext ioModuleContext,
    CFStringRef *status
);
```

PMIOModuleTerminateProcPtr

```
typedef OSStatus(* PMIOModuleTerminateProcPtr) (
    IOMContext *ioModuleContextPtr
);
```

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

```
OSStatus MyCallback (
    IOMContext *ioModuleContextPtr
);
```

PMIOModuleWriteProcPtr

```
typedef OSStatus(* PMIOModuleWriteProcPtr) (
    IOMContext ioModuleContext,
    Ptr buffer,
    UInt32 *size,
    Boolean eoj
);
```

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

```
OSStatus MyCallback (
    IOMContext ioModuleContext,
    Ptr buffer,
    UInt32 *size,
    Boolean eoj
);
```

PMIOOpenProcPtr

```
typedef OSStatus(* PMIOOpenProcPtr) (  
    const void *jobContext  
);
```

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

```
OSStatus MyPMIOOpenCallback (  
    const void *jobContext  
);
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

`PMPrinterModuleDeprecated.h`

PMIOReadProcPtr

```
typedef OSStatus(* PMIOReadProcPtr) (  
    const void *jobContext,  
    Ptr buffer,  
    UInt32 *size,  
    Boolean *eoj  
);
```

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

```
OSStatus MyPMIOReadCallback (  
    const void *jobContext,  
    Ptr buffer,  
    UInt32 *size,  
    Boolean *eoj  
);
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

`PMPrinterModuleDeprecated.h`

PMIOSetAttributeProcPtr

```
typedef OSStatus(* PMIOSetAttributeProcPtr) (
    const void *jobContext,
    CFStringRef attribute,
    CTypeRef data
);
```

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

```
OSStatus MyPMIOSetAttributeCallback (
    const void *jobContext,
    CFStringRef attribute,
    CTypeRef data
);
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

`PMPrinterModuleDeprecated.h`

PMIOStatusProcPtr

```
typedef OSStatus(* PMIOStatusProcPtr) (
    const void *jobContext,
    CFStringRef *status
);
```

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

```
OSStatus MyPMIOStatusCallback (
    const void *jobContext,
    CFStringRef *status
);
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

`PMPrinterModuleDeprecated.h`

PMIOWriteProcPtr

```
typedef OSStatus(* PMIOWriteProcPtr) (
    const void *jobContext,
    Ptr buffer,
    UInt32 *size,
    Boolean eof
);
```

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

```
OSStatus MyPMIOWriteCallback (
    const void *jobContext,
    Ptr buffer,
    UInt32 *size,
    Boolean eof
);
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

`PMPrinterModuleDeprecated.h`

PMJobStreamGetNextBandProcPtr

```
typedef OSStatus(* PMJobStreamGetNextBandProcPtr) (
    const void *jobContext,
    PMRasterBand *pmRasterBand
);
```

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

```
OSStatus MyPMJobStreamGetNextBandCallback (
    const void *jobContext,
    PMRasterBand *pmRasterBand
);
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

`PMPrinterModuleDeprecated.h`

PMJobStreamGetPosProcPtr

```
typedef OSStatus(* PMJobStreamGetPosProcPtr) (
    const void *jobContext,
    UInt32 *markerPos
);
```

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

```
OSStatus MyPMJobStreamGetPosCallback (
    const void *jobContext,
    UInt32 *markerPos
);
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

`PMPrinterModuleDeprecated.h`

PMJobStreamOpenProcPtr

```
typedef OSStatus(* PMJobStreamOpenProcPtr) (
    const void *jobContext
);
```

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

```
OSStatus MyPMJobStreamOpenCallback (
    const void *jobContext
);
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

`PMPrinterModuleDeprecated.h`

PMJobStreamReadWriteProcPtr

```
typedef OSStatus(* PMJobStreamReadWriteProcPtr) (
    const void *jobContext,
    void *buffPtr,
    UInt32 *size
);
```

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

```
OSStatus MyPMJobStreamReadWriteCallback (
```



```
    const void *jobContext,  
    void *buffPtr,  
    UInt32 *size  
);
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

PMPrinterModuleDeprecated.h

PMJobStreamSetPosProcPtr

```
typedef OSStatus(* PMJobStreamSetPosProcPtr) (  
    const void *jobContext,  
    SInt16 posMode,  
    UInt32 markerPos  
);
```

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

```
OSStatus MyPMJobStreamSetPosCallback (  
    const void *jobContext,  
    SInt16 posMode,  
    UInt32 markerPos  
);
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

PMPrinterModuleDeprecated.h

PMNotificationProcPtr

```
typedef OSStatus(* PMNotificationProcPtr) (  
    const void *jobContext,  
    CFDictionaryRef notificationDict,  
    CFDictionaryRef *notificationReplyDict  
);
```

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

```
OSStatus MyPMNotificationCallback (  
    const void *jobContext,  
    CFDictionaryRef notificationDict,  
    CFDictionaryRef *notificationReplyDict  
);
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

PMPrinterModuleDeprecated.h

PMPDECloseProcPtr

```
typedef OSStatus(* PMPDECloseProcPtr) (
    PMPDEContext context
);
```

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

```
OSStatus MyPMPDEClose (
    PMPDEContext context
);
```

Parameters

context

A pointer to a custom data structure that contains state information shared among the functions in a printing dialog extension. This is the same context defined by the prologue function—see [PMPDEPrologueProcPtr](#) (page 29).

Return Value

A result code.

Discussion

If your pane is visible, your close function is called when the user

- switches to another pane (including the Summary pane)
- changes printers
- dismisses the dialog by clicking Preview, Print, or Save

If your pane is visible and the user cancels the dialog, your close function is not called. The close function is not required to provide any services, and does not return any information to the caller.

PMPDEGetSummaryTextProcPtr

```
typedef OSStatus(* PMPDEGetSummaryTextProcPtr) (
    PMPDEContext context,
    CFArrayRef *titleArray,
    CFArrayRef *summaryArray
);
```

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

```
OSStatus MyPMPDEGetSummary (
    PMPDEContext context,
    CFArrayRef *titleArray,
    CFArrayRef *summaryArray
);
```

```
);
```

Parameters

context

A pointer to a custom data structure that contains state information shared among the functions in a printing dialog extension. This is the same context defined by the prologue function—see [PMPDEPrologueProcPtr](#) (page 29).

titleArray

A pointer to an array of strings that contain brief, localized descriptions of the settings in your user interface. The array and strings must be Core Foundation types.

On entry, the array is undefined. Before returning, you should assign an array of type `CFArrayRef`. The printing system assumes ownership of the array and is responsible for releasing it.

If your printing dialog extension does not supply summary text—or your user interface does not have any settings—you should assign `NULL` before returning.

summaryArray

A pointer to an array of strings that contain brief, localized descriptions of the current values of the settings in your user interface. The array and strings must be Core Foundation types.

On entry, the array is undefined. Before returning, you should assign an array of type `CFArrayRef`. The printing system assumes ownership of the array and is responsible for releasing it.

If your printing dialog extension does not supply summary text—or your user interface does not have any settings—you should assign `NULL` before returning.

Return Value

A result code.

Discussion

Whenever a user switches to the Summary pane in a printing dialog, the printing system calls the summary function of each active printing dialog extension.

The printing system expects *titleArray* and *summaryArray* to be in a one-to-one correspondence. Here's a formal description of what that means. Your user interface has associated with it *n* pairs of strings, each pair consisting of textual descriptions of a single setting and its current value. The two arrays are in a one-to-one correspondence if for all *k* from 0 to *n*-1, the strings *titleArray*[*k*] and *summaryArray*[*k*] form a valid pair.

PMPDEInitializeProcPtr

```
typedef OSStatus(* PMPDEInitializeProcPtr) (
    PMPDEContext context,
    PMPDEFlags *flags,
    PMPDERef ref,
    ControlRef embedderUserPane,
    PMPrintSession printSession
);
```

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

```
OSStatus MyPMPDEInitialize (
    PMPDEContext context,
    PMPDEFlags *flags,
    PMPDERef ref,
```

```
ControlRef embedderUserPane,
PMPrintSession printSession
);
```

Parameters

context

A pointer to a custom data structure that contains state information shared among the functions in a printing dialog extension. This is the same context defined by the prologue function—see [PMPDEPrologueProcPtr](#) (page 29).

flags

A pointer to an integer flag that provides information about the capabilities of your printing dialog extension. On entry, the flag is undefined. Before returning, your initialization function should assign a valid feature request flag. For information about the defined flags, see [“PDE Feature Flags”](#) (page 51).

ref

Reserved for future use.

embedderUserPane

The control provided by the printing system to host the user interface of your printing dialog extension. This parameter is a standard Carbon user pane—a root control into which your printing dialog extension embeds the various controls and other elements in your user interface.

Before returning, your initialization function should create static user interface elements—such as controls, icons, and images—and embed them in the user pane provided by the printing system. You should position the user interface elements with respect to the coordinate system of the dialog window, and make them visible. You should also set the initial values of your controls, based on either your default settings or previous settings saved by the user. Do not modify the user pane itself.

If any user interface elements require special handling, you can install Carbon event handlers for them. For example, the Duplex pane in the Print dialog uses an event handler to enable or disable the two binding selection buttons, based on the current checkbox setting.

Do not do any custom drawing at this time, since the user pane is still invisible. If the user decides to display your user interface, the printing system makes it visible and the Control Manager automatically draws your static elements.

printSession

A pointer to a print session object. This is the same print session created by the application prior to displaying the dialog. You can use this parameter to gain access to one of the specialized printing tickets, such as page format or print settings.

Return Value

A result code.

Discussion

If the printing system calls your initialization function, you can safely assume that your prologue function was called first.

If your initialization function returns a nonzero status code, the printing system does not call your summary function.

PMPDEOpenProcPtr

```
typedef OSStatus(* PMPDEOpenProcPtr) (
    PMPDEContext context
);
```

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

```
OSStatus MyPMPDEOpen (
    PMPDEContext context
);
```

Parameters

context

A pointer to a custom data structure that contains state information shared among the functions in a printing dialog extension. This is the same context defined by the prologue function—see [PMPDEPrologueProcPtr](#) (page 29).

Return Value

A result code.

Discussion

If the user selects your pane for display in the dialog, your open function is called immediately before the pane is made visible.

An open function is not required to provide any services, and does not return any information to the caller.

PMPDEPrologueProcPtr

```
typedef OSStatus(* PMPDEPrologueProcPtr) (
    PMPDEContext *context,
    OSType *creator,
    CFStringRef *userOptionKind,
    CFStringRef *title,
    UInt32 *maxH,
    UInt32 *maxV
);
```

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

```
OSStatus MyPMPDEPrologue (
    PMPDEContext *context,
    OSType *creator,
    CFStringRef *userOptionKind,
    CFStringRef *title,
    UInt32 *maxH,
    UInt32 *maxV
);
```

Parameters*context*

A pointer to a custom data structure that contains state information shared among the functions in a printing dialog extension. On entry, the caller's `PMPDEContext` variable is undefined. Before returning, your prologue function should assign a new context, or `NULL` to indicate that no context exists.

The printing system does not assume ownership of memory used for a context, so your printing dialog extension should release it when it is no longer needed. Typically this is done in your termination function.

creator

Reserved for future use.

userOptionKind

A pointer to a string that contains a unique identifier for the pane implemented by your printing dialog extension. The printing system uses this identifier to determine whether your printing dialog extension implements an Apple-defined pane, or a custom pane defined by you. On entry, the string is undefined. Before returning, your prologue function should assign the appropriate identifier.

If your printing dialog extension implements or overrides an Apple-defined pane, assign one of the identifiers listed in “[PDE Pane Kind Identifiers](#)” (page 53). If your printing dialog extension implements a custom pane, assign a Core Foundation string that contains your own custom identifier.

By convention, a custom identifier takes the form `<domain>.print.pde.<signature>`, where `<domain>` is your vendor-specific domain and `<signature>` is a short name or signature for the pane you are implementing.

The printing system does not assume ownership of this string, so your printing dialog extension should release it when it is no longer needed.

title

A pointer to a string that contains the localized title of the pane implemented by your printing dialog extension. The printing system displays this title in two places—the pane selection pop-up menu and the Summary pane.

On entry, the string is undefined. Before returning, your prologue function should assign a Core Foundation string containing the localized title.

If you are implementing an Apple-defined pane, the printing system may ignore your localized title and use an Apple-defined title instead.

Your printing dialog extension retains ownership of the string, and you should release it when it is no longer needed. Typically this is done in your terminate function.

maxH

A pointer to a value that represents the maximum number of horizontal pixels your user interface requires. When your user interface is made visible, the printing system might use this value to adjust the width of the dialog.

On entry, the value is undefined. Before returning, your prologue function should assign the maximum horizontal extent of your user interface in pixels.

maxV

A pointer to a value that represents the maximum number of vertical pixels your user interface requires. When your user interface is made visible, the printing system uses this value to adjust the height of the dialog.

On entry, the value is undefined. Before returning, your prologue function should assign the maximum vertical extent of your user interface in pixels.

Return Value

A result code.

Discussion

When the printing system displays a printing dialog, it calls the prologue function for each registered dialog extension.

If your prologue function returns a nonzero status code, the printing system does not include your pane in the dialog, and does not call your terminate function. In this circumstance, your prologue function should release any allocated memory or resources before exiting.

While the dialog is open, if some user action causes the printing system to scan for plug-ins again—choosing a different printer, for example—then your prologue function is called again.

PMPDESyncProcPtr

```
typedef OSStatus(* PMPDESyncProcPtr) (
    PMPDEContext context,
    PMPrintSession printSession,
    Boolean reinitializePlugIn
);
```

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

```
OSStatus MyPMPDESync (
    PMPDEContext context,
    PMPrintSession printSession,
    Boolean reinitializePlugIn
);
```

Parameters

context

A pointer to a custom data structure that contains state information shared among the functions in a printing dialog extension. This is the same context defined by the prologue function—see [PMPDEPrologueProcPtr](#) (page 29).

printSession

A pointer to a printing session object. Your synchronization function should use this parameter to gain access to the job ticket it uses to save user interface settings. These settings are stored as extended data in either the print settings ticket (for Print dialog extensions) or the page format ticket (for Page Setup dialog extensions).

reinitializePlugIn

A Boolean value that indicates which synchronization operation to perform. If the value is `true`, you should locate and retrieve the ticket settings and use them to update your user interface settings. If the value is `false`, you should use your current user interface settings to update the ticket settings.

Return Value

A result code.

Discussion

The printing system calls your synchronization function at certain times—in response to user actions—to update either the user interface or the ticket. Also, your printing dialog extension can call your synchronization function directly. For example, your initialization function may need to synchronize your user interface to its default settings.

Here are some calling conditions where the *reinitializePlugIn* parameter might be `true`:

- Your own initialization function wants to synchronize your user interface to its default settings.
- The printing system wants you to initialize your user interface from the print settings ticket associated with a different printer.

Here are some calling conditions where the `reinitializePlugin` parameter might be `false`:

- Your user interface is visible, and the user tries to switch to another pane.
- The user saves the current dialog settings.
- The user clicks the Preview button or the Print button.

You may want to check the validity of the current settings in your pane at this time, to avoid recording invalid or inconsistent settings in a job ticket. To prevent the user from switching to another pane until a problem is corrected, return the result code `kPMDontSwitchPDEError`.

PMPDETerminateProcPtr

```
typedef OSStatus(* PMPDETerminateProcPtr) (
    PMPDEContext context,
    OSStatus status
);
```

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

```
OSStatus MyPMPDETerminate (
    PMPDEContext context,
    OSStatus status
);
```

Parameters

context

A pointer to a custom data structure that contains state information shared among the functions in a printing dialog extension. This is the same context defined by the prologue function—see [PMPDEPrologueProcPtr](#) (page 29).

status

Reserved for future use to indicate the conditions under which the user dismissed the dialog.

Return Value

A result code.

Discussion

The printing system calls your termination function when the printing system decides to tear down or rebuild the dialog, or when your pane is no longer needed.

Unless your prologue function returns a nonzero status code, your termination function is always called when the user dismisses the dialog.

Your termination function is also called if your printing feature is no longer relevant—for example, if your feature applies only to postscript printers and the user switches to a raster printer from within the dialog.

Your termination function should release Core Foundation objects, deallocate your context block, and perform any other actions necessary before your user interface is reinitialized or unloaded.

PMPluginGetAPIVersionProcPtr

```
typedef OSStatus(* PMPluginGetAPIVersionProcPtr) (
    PMPluginHeaderInterface *obj,
    PMPluginAPIVersion *versionPtr
);
```

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

```
OSStatus MyPMPluginGetAPIVersion (
    PMPluginHeaderInterface *obj,
    PMPluginAPIVersion *versionPtr
);
```

Parameters

obj

A pointer to a generic instance of a plug-in interface. You may ignore this parameter, as it isn't useful here.

versionPtr

A pointer to a data structure supplied by the caller for version information about your printing plug-in. Before returning, you should provide the correct version information for your plug-in by assigning version constants to the fields in this structure.

Return Value

A result code.

PMPluginReleaseProcPtr

Defines a pointer to the retain function in a printing plug-in interface. Your custom retain function increments the reference count of an instance of one of these interfaces.

```
typedef OSStatus (*PMPluginReleaseProcPtr) (
    PMPluginHeaderInterface **objPtr
);
```

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

```
OSStatus MyPMPluginRelease (
    PMPluginHeaderInterface **objPtr
);
```

Parameters

objPtr

The address of a pointer to a generic instance of a plug-in interface. Before returning, your retain function should decrement the reference count of the instance, and assign `NULL` to the pointer whose address is provided by this parameter. If the reference count reaches zero, you should delete the instance and any related storage.

Return Value

A result code.

PMPluginRetainProcPtr

```
typedef OSStatus (*PMPluginRetainProcPtr) (
    PMPluginHeaderInterface *obj
);
```

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

```
OSStatus MyPMPluginRetain (
    PMPluginHeaderInterface *obj
);
```

Parameters

obj

A pointer to a generic instance of a plug-in interface. Before returning, your retain function should increment the reference count of the instance.

Return Value

A result code.

PMPrBrowserAPIVersionProcPtr

```
typedef UInt32(* PMPrBrowserAPIVersionProcPtr) ();
```

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

```
UInt32 MyPMPrBrowserAPIVersionCallback ();
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

`PMPrinterBrowsers.h`

PMPrBrowserGetLookupSpecProcPtr

```
typedef OSStatus(* PMPrBrowserGetLookupSpecProcPtr) (
    PMPrBrowserRef ref,
    UInt32 specIndex,
    CFDictionaryRef *lookupSpec
);
```

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

```
OSStatus MyPMPrBrowserGetLookupSpecCallback) (
    PMPrBrowserRef ref,
    UInt32 specIndex,
    CFDictionaryRef *lookupSpec
);
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

PMPrBrowserBrowsers.h

PMPrBrowserGetSelectedPrintersProcPtr

```
typedef OSStatus(* PMPrBrowserGetSelectedPrintersProcPtr) (  
    PMPrBrowserContext context,  
    CFArrayRef *printers  
);
```

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

```
OSStatus MyPMPrBrowserGetSelectedPrintersCallback (  
    PMPrBrowserContext context,  
    CFArrayRef *printers  
);
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

PMPrBrowserBrowsers.h

PMPrBrowserInitializeProcPtr

```
typedef OSStatus(* PMPrBrowserInitializeProcPtr) (  
    PMPrBrowserContext context,  
    PMPrBrowserRef ref,  
    PMPrBrowserCallbacks *callbacks,  
    ControlRef pbUserPaneCtlHdl,  
    UInt32 numLookupSpecs  
);
```

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

```
OSStatus MyPMPrBrowserInitializeCallback (  
    PMPrBrowserContext context,  
    PMPrBrowserRef ref,  
    PMPrBrowserCallbacks *callbacks,  
    ControlRef pbUserPaneCtlHdl,  
    UInt32 numLookupSpecs  
);
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

PMPrinterBrowsers.h

PMPrBrowserPrologueProcPtr

```
typedef OSStatus(* PMPrBrowserPrologueProcPtr) (
    PMPrBrowserContext *context,
    PMPrBrowserFlags prologueFlags,
    CFStringRef *title,
    UInt32 *minH,
    UInt32 *minV,
    UInt32 *maxH,
    UInt32 *maxV
);
```

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

```
OSStatus MyPMPrBrowserPrologueCallback (
    PMPrBrowserContext *context,
    PMPrBrowserFlags prologueFlags,
    CFStringRef *title,
    UInt32 *minH,
    UInt32 *minV,
    UInt32 *maxH,
    UInt32 *maxV
);
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

PMPrinterBrowsers.h

PMPrBrowserResizeProcPtr

```
typedef OSStatus(* PMPrBrowserResizeProcPtr) (
    PMPrBrowserContext context,
    const Rect *frameRect
);
```

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

```
OSStatus MyPMPrBrowserResizeCallback (
    PMPrBrowserContext context,
    const Rect *frameRect
);
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

PMPrnterBrowsers.h

PMPrBrowserSelectionStatusProcPtr

```
typedef OSStatus(* PMPrBrowserSelectionStatusProcPtr) (  
    PMPrBrowserRef ref,  
    Boolean selected,  
    Boolean addNow  
);
```

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

```
OSStatus MyPMPrBrowserSelectionStatusCallback (  
    PMPrBrowserRef ref,  
    Boolean selected,  
    Boolean addNow  
);
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

PMPrnterBrowsers.h

PMPrBrowserSyncProcPtr

```
typedef OSStatus(* PMPrBrowserSyncProcPtr) (  
    PMPrBrowserContext context  
);
```

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

```
OSStatus MyPMPrBrowserSyncCallback (  
    PMPrBrowserContext context  
);
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

PMPrnterBrowsers.h

PMPPrBrowserSyncRequestProcPtr

```
typedef OSStatus(* PMPPrBrowserSyncRequestProcPtr) (
    PMPPrBrowserRef ref
);
```

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

```
OSStatus MyPMPPrBrowserSyncRequestCallback (
    PMPPrBrowserRef ref
);
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

`PMPPrinterBrowsers.h`

PMPPrBrowserTerminateProcPtr

```
typedef OSStatus(* PMPPrBrowserTerminateProcPtr) (
    PMPPrBrowserContext context,
    OSStatus status
);
```

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

```
OSStatus MyPMPPrBrowserTerminateCallback (
    PMPPrBrowserContext context,
    OSStatus status
);
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

`PMPPrinterBrowsers.h`

PMPPrBrowserWorksetPrintersProcPtr

```
typedef OSStatus(* PMPPrBrowserWorksetPrintersProcPtr) (
    PMPPrBrowserContext context,
    CFArrayRef printers
);
```

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

```
OSStatus MyPMPPrBrowserWorksetPrintersCallback (
    PMPPrBrowserContext context,
```

```
CFArrayRef printers  
);
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

PMPrinterBrowsers.h

PMPrintJobProcPtr

```
typedef OSStatus(* PMPrintJobProcPtr) (  
    PMContext printerModuleContext,  
    const void *jobContext,  
    PMTicketRef jobTicket,  
    const PMJobStreamProcs *inDataProcs  
);
```

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

```
OSStatus MyPMPrintJobCallback (  
    PMContext printerModuleContext,  
    const void *jobContext,  
    PMTicketRef jobTicket,  
    const PMJobStreamProcs *inDataProcs  
);
```

PMPrintPageProcPtr

```
typedef OSStatus(* PMPrintPageProcPtr) (  
    PMContext printerModuleContext,  
    const void *jobContext,  
    PMTicketRef jobTicket,  
    const PMJobStreamGetNextBandProcPtr pmJobStreamGetNextBandProc  
);
```

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

```
OSStatus MyPMPrintPageCallback (  
    PMContext printerModuleContext,  
    const void *jobContext,  
    PMTicketRef jobTicket,  
    const PMJobStreamGetNextBandProcPtr pmJobStreamGetNextBandProc  
);
```

PMTerminateProcPtr

```
typedef OSStatus(* PMSerializeProcPtr) (
    PMContext *printerModuleContext,
    const void *jobContext
);
```

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

```
OSStatus MyPMSerializeCallback (
    PMContext *printerModuleContext,
    const void *jobContext
);
```

Data Types

Printing Plug-in Data Types

PMPlugInHeader

Defines the table of callback functions in the printing plug-in interface.

```
struct PMPlugInHeader
{
    OSStatus (*Retain) (
        PMPlugInHeaderInterface *obj
    );

    OSStatus (*Release) (
        PMPlugInHeaderInterface **objPtr
    );

    OSStatus (*GetAPIVersion) (
        PMPlugInHeaderInterface *obj,
        PMPlugInAPIVersion *versionPtr
    );
};
typedef struct PMPlugInHeader PMPlugInHeader;
```

Fields**Retain**

A pointer to a function that satisfies the requirements for a printing plug-in retain function, as described in [PMPlugInRetainProcPtr](#) (page 34).

Release

A pointer to a function that satisfies the requirements for a printing plug-in release function, as described in [PMPlugInReleaseProcPtr](#) (page 33).

GetAPIVersion

A pointer to a function that satisfies the requirements for a printing plug-in version function, as described in [PMPlugInGetAPIVersionProcPtr](#) (page 33).

Availability

Available in Mac OS X v10.0 and later.

Declared In

PMPluginHeader.h

PMPluginHeaderInterface

Defines a generic instance of the `PMPluginHeader` interface—the printing system passes a parameter of this type when calling the three interface functions.

```
struct PMPluginHeaderInterface {
    const PMPluginHeader *vtable;
};
typedef struct PMPluginHeaderInterface PMPluginHeaderInterface;
```

Fields

`vtable`

A pointer to the function table for an implementation of the `PMPluginHeader` interface.

Discussion

When the printing system calls one of the three functions in the `PMPluginHeader` interface, a pointer to this generic data type is supplied as a parameter. It's really the address of the plug-in instance supplied by your query interface function (see *Core Foundation Plug-in Services*).

To gain access to your instance data—including the field that holds the reference count—you need to cast this pointer to the actual instance type defined in your plug-in.

Availability

Available in Mac OS X v10.0 and later.

Declared In

PMPluginHeader.h

PMPluginAPIVersion

Contains `PMPluginHeader` interface version information that printing plug-ins must provide to the printing system.

```
struct PMPluginAPIVersion {
    UInt32 buildVersionMajor;
    UInt32 buildVersionMinor;
    UInt32 baseVersionMajor;
    UInt32 baseVersionMinor;
};
```

Fields

`buildVersionMajor`

An integer value that specifies the major component of the API version with which the plug-in was compiled.

`buildVersionMinor`

An integer value that specifies the minor component of the API version with which the plug-in was compiled.

baseVersionMajor

An integer value that specifies the major component of the base API version with which this plug-in is forward compatible.

baseVersionMinor

An integer value that specifies the minor component of the base API version with which this plug-in is forward compatible.

Printing Dialog Extension Data Types

PlugInIntfVTable

Defines the table of callback functions in the printing dialog extension plug-in interface.

```
struct PlugInIntfVTable
{
    PMPlugInHeader plugInHeader;

    OSStatus (*Prologue) (
        PMPDEContext *context,
        OSType *creator,
        CFStringRef *userOptionKind,
        CFStringRef *title,
        UInt32 *maxH,
        UInt32 *maxV
    );

    OSStatus (*Initialize) (
        PMPDEContext context,
        PMPDEFlags *flags,
        PMPDERef ref,
        ControlRef embedderUserPane,
        PMPrintSession printSession
    );

    OSStatus (*Sync) (
        PMPDEContext context,
        PMPrintSession printSession,
        Boolean reinitializePlugIn
    );

    OSStatus (*GetSummaryText) (
        PMPDEContext context,
        CFArrayRef *titleArray,
        CFArrayRef *summaryArray
    );

    OSStatus (*Open) (
        PMPDEContext context
    );

    OSStatus (*Close) (
        PMPDEContext context
    );
};
```

```

    OSStatus (*Terminate) (
        PMPDEContext context,
        OSStatus status
    );
};
typedef struct PlugInIntfVTable PlugInIntfVTable;

```

Fields

plugInHeader

A table of pointers to the three functions in the printing plug-in interface.

Prologue

A pointer to a function that satisfies the requirements for the prologue function in a printing dialog extension, as described in [PMPDEPrologueProcPtr](#) (page 29).

Initialize

A pointer to a function that satisfies the requirements for the initialization function in a printing dialog extension, as described in [PMPDEInitializeProcPtr](#) (page 27).

Sync

A pointer to a function that satisfies the requirements for the synchronization function in a printing dialog extension, as described in [PMPDESyncProcPtr](#) (page 31).

GetSummaryText

A pointer to a function that satisfies the requirements for the summary function in a printing dialog extension, as described in [PMPDEGetSummaryTextProcPtr](#) (page 26).

Open

A pointer to a function that satisfies the requirements for the open function in a printing dialog extension, as described in [PMPDEOpenProcPtr](#) (page 29).

Close

A pointer to a function that satisfies the requirements for the close function in a printing dialog extension, as described in [PMPDECloseProcPtr](#) (page 26).

Terminate

A pointer to a function that satisfies the requirements for the termination function in a printing dialog extension, as described in [PMPDETerminateProcPtr](#) (page 32).

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

PMPrintingDialogExtensionsDeprecated.h

PlugInIntf

Defines a generic instance of the printing dialog extension plug-in interface.

```

struct PlugInIntf {
    PlugInIntfVTable *vtable;
};

```

Fields

vtable

A pointer to the table of callback functions in a printing dialog extension.

PMPDEContext

Defines a generic private context for a printing dialog extension—the printing system passes a parameter of this type when calling functions in the `PlugInIntfVTable` interface.

```
typedef struct OpaquePMPDEContext* PMPDEContext;
```

Discussion

A **context** is a pointer to a custom data structure shared by the functions in a printing dialog extension. In Mac OS X, Carbon applications can open the same printing dialog in several document windows concurrently. Printing dialog extensions must support this possibility, which means they must be reentrant.

To ensure reentrancy, your printing dialog extension uses dynamically allocated memory for state information specific to one dialog window. Your prologue function allocates this memory and supplies its address—called a context—to the printing system. The printing system makes sure that the context associated with the current session (the active dialog) is passed in to your other functions as a calling parameter.

The context you provide is a generic pointer of type `PMPDEContext`. You can gain access to your actual context data by casting this parameter to the actual context type defined in your printing dialog extension.

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

`PMPrintingDialogExtensionsDeprecated.h`

PMPDEFlags

Defines an integer flag that provides additional information about a printing dialog extension to the printing system. A value of this data type is passed back to the printing system by the initialize function.

```
typedef UInt32 PMPDEFlags;
enum {
    kPMPDENoFlags = 0,
    kPMPDENoSummary = 1,
    kPMPDEAllFlags = -1
};
```

Discussion

The printing system examines this integer flag to learn more about its capabilities. For a description of the initialize function, see [PMPDEInitializeProcPtr](#) (page 27).

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

`PMPrintingDialogExtensionsDeprecated.h`

PMPDERef

Defines a generic instance of the printing dialog extension plug-in interface.

```
typedef struct OpaquePMPDERef* PMPDERef;
```

Discussion

A value of this type is passed to the initialize function. For a description of the initialize function, see [PMPDEInitializeProcPtr](#) (page 27).

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

PMPrintingDialogExtensionsDeprecated.h

Printer Module Data Types

PMProcs

Defines the table of callback functions in the printer module plug-in interface.

```
struct PMProcs
{
    PMPlugInHeader                pluginHeader;
    PMCreatePBMInfoProcPtr        CreatePBMInfo;
    PMInitializeProcPtr           Initialize;
    PMCreatePDEPathsProcPtr       CreatePDEPaths;
    PMCreatePrinterTicketsProcPtr CreatePrinterTickets;
    PMBeginJobProcPtr             BeginJob;
    PMPrintJobProcPtr             PrintJob;
    PMPrintPageProcPtr            PrintPage;
    PMImageAccessProcPtr          ImageAccess;
    PMCancelJobProcPtr            CancelJob;
    PMEndJobProcPtr               EndJob;
    PMTerminateProcPtr            Terminate;
};
```

PMInterface

Defines a generic instance of the printer module plug-in interface.

```
struct PMInterface {
    const PMProcs *vtable;
};
typedef struct PMInterface PMInterface;
```

Fields

vtable

A pointer to the table of callback functions in a printer module.

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

PMPrinterModuleDeprecated.h

PMInterfaceRef

Defines a pointer to a generic instance of the printer module plug-in interface.

```
typedef PMInterface* PMInterfaceRef;
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

PMPrinterModuleDeprecated.h

PMIOProcsDefines the table of callback functions in the `PMIOProcs` interface.

```
struct PMIOProcs
{
    CFIndex                version;
    GetConnInfoProcPtr    GetConnInfoProc;
    PMIOOpenProcPtr       PMIOOpenProc;
    PMIOReadProcPtr       PMIOReadProc;
    PMIOWriteProcPtr      PMIOWriteProc;
    PMIOStatusProcPtr     PMIOStatusProc;
    PMIOGetAttributeProcPtr PMIOGetAttributeProc;
    PMIOSetAttributeProcPtr PMIOSetAttributeProc;
    PMIOCloseProcPtr      PMIOCloseProc;
};
```

PMJobStreamProcs

Defines the table of callback functions in the printer module data access interface.

```
struct PMJobStreamProcs
{
    CFIndex                version;
    PMJobStreamOpenProcPtr PMJobStreamOpenProc;
    PMJobStreamReadWriteProcPtr PMJobStreamReadProc;
    PMJobStreamReadWriteProcPtr PMJobStreamWriteProc;
    PMJobStreamGetPosProcPtr PMJobStreamGetPosProc;
    PMJobStreamSetPosProcPtr PMJobStreamSetPosProc;
    PMJobStreamGetPosProcPtr PMJobStreamGetEOFProc;
};
```

PMContext

Defines an opaque type for a private context in a printer module.

```
typedef struct OpaquePMContext* PMContext;
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

PMPrinterModuleDeprecated.h

PMDrawingCtx

Defines an opaque type for a drawing context in a printer module.

```
typedef struct OpaquePMDrawingCtx* PMDrawingCtx;
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

PMPrinterModuleDeprecated.h

PMImageRef

Defines an opaque type for an image in a printer module.

```
typedef struct OpaquePMImageRef* PMImageRef;
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

PMPrinterModuleDeprecated.h

Printer Browser Module Data Types

PMInterfacePrBrowser

Defines the table of callback functions in the printer browser module plug-in interface.

```

struct PMInterfacePrBrowser
{
    IUnknownVTbl u;
    PMPPrBrowserGetSelectedPrintersProcPtr getSelectedPrinters;
    PMPPrBrowserInitializeProcPtr initialize;
    PMPPrBrowserPrologueProcPtr prologue;
    PMPPrBrowserResizeProcPtr resize;
    PMPPrBrowserSyncProcPtr sync;
    PMPPrBrowserTerminateProcPtr terminate;
    PMPPrBrowserWorksetPrintersProcPtr worksetPrinters;
};

```

PMInterfacePrBrowserPtr

Defines a pointer to a function table for the `PMInterfacePrBrowser` interface.

```
typedef PMInterfacePrBrowser* PMInterfacePrBrowserPtr;
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

`PMPrinterBrowsers.h`

PMPPrBrowserCallbacks

Defines the table of Print Center callback routines for printer browser modules.

```

struct PMPPrBrowserCallbacks
{
    CFIndex version;
    PMPPrBrowserGetLookupSpecProcPtr getLookupSpec;
    PMPPrBrowserSyncRequestProcPtr syncRequest;
    PMPPrBrowserSelectionStatusProcPtr selStatus;
};

```

PMPPrBrowserCallbacksPtr

Defines a pointer to a table of Print Center callbacks for printer browser modules.

```
typedef PMPPrBrowserCallbacks* PMPPrBrowserCallbacksPtr;
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

`PMPrinterBrowsers.h`

PMPPrBrowserContext

Defines an opaque type for a private context in a printer browser module.

```
typedef struct OpaquePMPPrBrowserContext* PMPPrBrowserContext;
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

PMPPrinterBrowsers.h

PMPPrBrowserFlags

Defines an integer flag that provides additional information about a printer browser module to the printing system.

```
typedef UInt32 PMPPrBrowserFlags;
enum {
    kPMPPrBrowserPCNoFlags = 0,
    kPMPPrBrowserPCNoUI = 1,
    kPMPPrBrowserPCAllFlags = -1
};
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

PMPPrinterBrowsers.h

PMPPrBrowserRef

Defines an opaque type for an instance of the printer browser module plug-in interface.

```
typedef struct OpaquePMPPrBrowserRef* PMPPrBrowserRef;
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

PMPPrinterBrowsers.h

PMInterfaceAPIVersion

Defines the table of callback functions in the `PMInterfaceAPIVersion` interface.

```
struct PMInterfaceAPIVersion
{
    IUnknownVtbl          u;
    PMPPrBrowserAPIVersionProcPtr  apiVersion;
};
```

PMInterfaceAPIVersionPtr

Defines a pointer to a function table for the `PMInterfaceAPIVersion` interface.

```
typedef PMInterfaceAPIVersion* PMInterfaceAPIVersionPtr;
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In

`PMPrinterBrowsers.h`

I/O Module Data Types

IOMContext

Defines an opaque type for a private context in an I/O module.

```
typedef struct OpaqueIOMContext* IOMContext;
```

Availability

Available in Mac OS X v10.0 and later.

Declared In

`PMIOModule.h`

IOMInterface

Defines a generic instance of the I/O module plug-in interface.

```
struct IOMInterface {
    const IOMProcs *vtable;
};
```

Fields

`vtable`

A pointer to the table of callback functions in an I/O module.

IOMInterfaceRef

Defines a pointer to a generic instance of the I/O module plug-in interface.

```
typedef const IOMInterface* IOMInterfaceRef;
```

Availability

Available in Mac OS X v10.0 and later.

Declared In

PMIOModule.h

IOMProcs

Defines the table of callback functions in the I/O module plug-in interface.

```
struct IOMProcs
{
    PMPlugInHeader pluginHeader;

    PMIOModuleGetConnectionInfoProcPtr    GetConnectionInfo;
    PMIOModuleInitializeProcPtr           Initialize;
    PMIOModuleOpenProcPtr                  Open;
    PMIOModuleReadProcPtr                  Read;
    PMIOModuleWriteProcPtr                  Write;
    PMIOModuleStatusProcPtr                 Status;
    PMIOModuleGetAttributeProcPtr          GetAttribute;
    PMIOModuleSetAttributeProcPtr          SetAttribute;
    PMIOModuleCloseProcPtr                  Close;
    PMIOModuleTerminateProcPtr              Terminate;
};
typedef struct IOMProcs IOMProcs;
```

Availability

Available in Mac OS X v10.0 and later.

Declared In

PMIOModule.h

Constants

PDE Feature Flags

Specify the flags returned by a printing dialog extension in its initialization function.

```
typedef UInt32 PMPDEFlags;
enum {
    kPMPDENoFlags = 0,
    kPMPDENoSummary = 1,
    kPMPDEAllFlags = -1
};
```

Constants

kPMPDENoFlags

Specifies that a printing dialog extension does not have any special capabilities to report to the printing system.

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared in `PMPrintingDialogExtensionsDeprecated.h`.

kPMPDENoSummary

Specifies that a printing dialog extension does not provide summary information. Your initialization function can indicate this to the printing system by using the `flags` parameter to pass back this constant. For more information about the initialization function, see [PMPDEInitializeProcPtr](#) (page 27). For more information about providing summary information, see [PMPDEGetSummaryTextProcPtr](#) (page 26).

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared in `PMPrintingDialogExtensionsDeprecated.h`.

kPMPDEAllFlags

Not used.

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared in `PMPrintingDialogExtensionsDeprecated.h`.

PDE Interface Identifier

Specifies the unique identifier for the printing dialog extension plug-in interface.

```
#define kDialogExtensionIntfIDStr
CFSTR("A996FD7E-B738-11D3-8519-0050E4603277")
```

Constants

kDialogExtensionIntfIDStr

A UUID that a printing dialog extension should use in its query interface function to verify that the caller wants an instance of a printing dialog extension plug-in interface, and not some other plug-in interface.

Discussion

For more information about the query interface function, see *Core Foundation Plug-in Services*.

PDE Interface Version

Specify the major and minor components of the version numbers for the interface supported by your printing dialog extension.

```
#define kPDEBuildVersionMajor 1
#define kPDEBuildVersionMinor 0
#define kPDEBaseVersionMajor 1
#define kPDEBaseVersionMinor 0
```

Constants

kPDEBuildVersionMajor

Specifies that Apple has designated the current release of the printing dialog extension plug-in interface to be major version 1, as in 1.0.

kPDEBuildVersionMinor

Specifies that Apple has designated the current release of the printing dialog extension plug-in interface to be minor version 0, as in 1.0.

kPDEBaseVersionMajor

Specifies that Apple has designated the first release of the printing dialog extension plug-in interface to be major version 1, as in 1.0.

kPDEBaseVersionMinor

Specifies that Apple has designated the first release of the printing dialog extension plug-in interface to be minor version 0, as in 1.0.

PDE Pane Kind Identifiers

Identifiers associated with the Apple-defined panes in a printing dialog.

```
#define kPMColorPDEKindID CFSTR("com.apple.print.pde.ColorKind")
#define kPMCopiesAndPagesPDEKindID CFSTR("com.apple.print.pde.CopiesAndPagesKind")
#define kPMCoverPagePDEKindID CFSTR("com.apple.print.pde.CoverPageKind")
#define kPMDuplexPDEKindID CFSTR("com.apple.print.pde.DuplexKind")
#define kPMLayoutPDEKindID CFSTR("com.apple.print.pde.LayoutUserOptionKind")
#define kPMOutputOptionsPDEKindID CFSTR("com.apple.print.pde.OutputOptionsKind")
#define kMPMPageAttributesPDEKindID CFSTR("com.apple.print.pde.PageAttributesKind")
#define kMPMPaperFeedPDEKindID CFSTR("com.apple.print.pde.PaperFeedKind")
#define kMPMPriorityPDEKindID CFSTR("com.apple.print.pde.PriorityKind")
#define kPMRotationScalingPDEKindID CFSTR("com.apple.print.pde.RotationScalingKind")
```

Constants

kPMColorPDEKindID

Identifies the Print dialog pane named Color Option.

kPMCopiesAndPagesPDEKindID

Identifies the Print dialog pane named Copies & Pages.

kPMCoverPagePDEKindID

Identifies the Print dialog pane named Cover Page.

kPMDuplexPDEKindID

Identifies the Print dialog pane named Duplex.

- kPMLayoutPDEKindID
Identifies the Print dialog pane named Layout.
- kPMOutputOptionsPDEKindID
Identifies the Print dialog pane named Output Options.
- kPMPageAttributesKindID
Identifies the Page Setup dialog pane named Page Attributes.
- kPMPaperFeedPDEKindID
Identifies the Print dialog pane named Paper Feed.
- kPMPriorityPDEKindID
Identifies the Print dialog pane named Priority.
- kPMRotationScalingPDEKindID
Identifies the Print dialog pane named Rotation & Scaling.

PDE Ticket Identifiers

Specify the types of printing job tickets used by printing dialog extensions.

```
#define kPDE_PMPrintSettingsRef    CFSTR("PMPrintSettingsTicket")
#define kPDE_PMPageFormatRef      CFSTR("PMPageFormatTicket")
#define kPDE_PMJobTemplateRef     CFSTR("PMJobTemplateTicket")
#define kPDE_PMPrinterInfoRef     CFSTR("PMPrinterInfoTicket")
```

Constants

- kPDE_PMPrintSettingsRef
Specifies a print settings ticket.
- kPDE_PMPageFormatRef
Specifies a page format ticket.
- kPDE_PMJobTemplateRef
Specifies a job template ticket.
- kPDE_PMPrinterInfoRef
Specifies a printer info ticket.

PDE Type Identifiers

Specify the different types of printing dialog extensions.

```
#define kAppPageSetupDialogTypeIDStr
CFSTR("B9A0DA98-E57F-11D3-9E83-0050E4603277")
#define kAppPrintDialogTypeIDStr
CFSTR("BCB07250-E57F-11D3-8CA6-0050E4603277")
#define kPrinterModuleTypeIDStr
CFSTR("BDB091F4-E57F-11D3-B5CC-0050E4603277")
```

Constants

- kAppPageSetupDialogTypeIDStr
The identifier for a Page Setup dialog extension supplied with an application.
- kAppPrintDialogTypeIDStr
The identifier for a Print dialog extension supplied with an application.

`kPrinterModuleTypeIDStr`

The identifier for a Print dialog extension supplied with a printer module.

Discussion

Apple associates each type of printing dialog extension with a UUID, represented as a constant Core Foundation string.

One or more of these constants is used in the `CFPlugInTypes` property list entry of a printing dialog extension, to declare a supported type and associate the type with a factory function that implements it.

For more information about factory functions, see *Core Foundation Plug-in Services*.

I/O Module Interface Version

Specify the major and minor components of the version numbers for the interface supported by your I/O module.

```
enum {
    kIOBuildVersionMajor = 1,
    kIOBuildVersionMinor = 0,
    kIOBaseVersionMajor = 1,
    kIOBaseVersionMinor = 0
};
```

Printer Module Interface Version

Specify the major and minor components of the version numbers for the interface supported by your printer module.

```
enum {
    kPMBuildVersionMajor = 1,
    kPMBuildVersionMinor = 0,
    kPMBaseVersionMajor = 1,
    kPMBaseVersionMinor = 0
};
```

Printer Module Status Codes

Specify the status and error event codes and keys reported by the printer module (used in `PMNotificationProc`).

```
enum {
    kPMEventPrinterStatus = 4000,
    kPMEventErrorOccurred = 4001,
    kPMEventRecoverableErrorOccurred = 4002,
    kPMEventRecoverableErrorCleared = 4003
};
```

Other Printer Module Constants

Defined in `PMPrinterModule.h`.

```
enum {
    kPMBrowserInfoNumValues = 4
};
```

Print Center Feature Flags

Specify the flags returned by Print Center in its prologue function.

```
typedef UInt32 PMPPrBrowserFlags;
enum {
    kPMPPrBrowserPCNoFlags = 0,
    kPMPPrBrowserPCNoUI = 1,
    kPMPPrBrowserPCAllFlags = -1
};
```

Print Center Signatures

Specify the Print Center creator code and icon signatures.

```
enum {
    kPMPPrBrowserPCCreator = 'pctr',
    kPMPPrBrowserWorksetPrinterIconType = 'wspr',
    kPMPPrBrowserUnknownPrinterIconType = '?ptr'
};
```

Result Codes

The table below lists the result codes defined in the Carbon Printing Manager for functions in the printing plug-in interfaces.

Result Code	Value	Description
kPMCloseFailed	-9785	A file or connection could not be closed. Available in Mac OS X v10.0 and later.
kPMDontSwitchPDEError	-9531	Tells the printing system not to switch out of the current pane. Available in Mac OS X v10.0 and later.
kPMEditRequestFailed	-9544	Error handling request to update Edit menu. Available in Mac OS X v10.0 and later.
kPMInvalidLookupSpec	-9542	Error retrieving lookup specification. Available in Mac OS X v10.0 and later.
kPMInvalidPBMLRef	-9540	Invalid printer browser module. Available in Mac OS X v10.0 and later.

Result Code	Value	Description
kPMInvalidPDEContext	-9530	Invalid printing dialog extension context. Available in Mac OS X v10.0 and later.
kPMInvalidPrinterAddress	-9780	Invalid printer address. NetInfo printcap entry not found.Unable to open USB interface. Available in Mac OS X v10.0 and later.
kPMIOAttrNotAvailable	-9787	I/O attribute not available on current connection type. Available in Mac OS X v10.0 and later.
kPMNoSelectedPrinters	-9541	No selected printers or error getting selection. Available in Mac OS X v10.0 and later.
kPMOpenFailed	-9781	A file or connection could not be opened. Available in Mac OS X v10.0 and later.
kPMPrBrowserNoUI	-9545	User interface function call with no user interface present. Available in Mac OS X v10.0 and later.
kPMReadFailed	-9782	A file or connection read operation failed. Available in Mac OS X v10.0 and later.
kPMReadGotZeroData	-9788	A file or connection read operation returned no data. Available in Mac OS X v10.1 and later.
kPMStatusFailed	-9784	Connection status failed. Available in Mac OS X v10.0 and later.
kPMSyncRequestFailed	-9543	Error handling sync request. Available in Mac OS X v10.0 and later.
kPMUnsupportedConnection	-9786	Connection type not supported. Available in Mac OS X v10.0 and later.
kPMWriteFailed	-9783	A file or connection write operation failed. Available in Mac OS X v10.0 and later.

Document Revision History

This table describes the changes to *Printing Plug-in Interfaces Reference*.

Date	Notes
2002-06-01	First version of this document.

REVISION HISTORY

Document Revision History

Index

G

GetConnInfoProcPtr [callback 12](#)

I

I/O Module Interface Version [55](#)

IOMContext [data type 50](#)

IOMInterface [structure 50](#)

IOMInterfaceRef [data type 50](#)

IOMProcs [structure 51](#)

K

kAppPageSetupDialogTypeIDStr [constant 54](#)

kAppPrintDialogTypeIDStr [constant 54](#)

kDialogExtensionIntfIDStr [constant 52](#)

kPDEBaseVersionMajor [constant 53](#)

kPDEBaseVersionMinor [constant 53](#)

kPDEBuildVersionMajor [constant 53](#)

kPDEBuildVersionMinor [constant 53](#)

kPDE_PMJobTemplateRef [constant 54](#)

kPDE_PMPageFormatRef [constant 54](#)

kPDE_PMPrinterInfoRef [constant 54](#)

kPDE_PMPrintSettingsRef [constant 54](#)

kPMCloseFailed [constant 56](#)

kPMColorPDEKindID [constant 53](#)

kPMCopiesAndPagesPDEKindID [constant 53](#)

kPMCoverPagePDEKindID [constant 53](#)

kPMDontSwitchPDEError [constant 56](#)

kPMDuplexPDEKindID [constant 53](#)

kPMEditRequestFailed [constant 56](#)

kPMInvalidLookupSpec [constant 56](#)

kPMInvalidPBMRef [constant 56](#)

kPMInvalidPDEContext [constant 57](#)

kPMInvalidPrinterAddress [constant 57](#)

kPMIOAttrNotAvailable [constant 57](#)

kPMLayoutPDEKindID [constant 54](#)

kPMNoSelectedPrinters [constant 57](#)

kPMOpenFailed [constant 57](#)

kPMOutputOptionsPDEKindID [constant 54](#)

kPMPageAttributesKindID [constant 54](#)

kPMPaperFeedPDEKindID [constant 54](#)

kPMPDEAllFlags [constant 52](#)

kPMPDENoFlags [constant 52](#)

kPMPDENoSummary [constant 52](#)

kPMPrBrowserNoUI [constant 57](#)

kPMPriorityPDEKindID [constant 54](#)

kPMReadFailed [constant 57](#)

kPMReadGotZeroData [constant 57](#)

kPMRotationScalingPDEKindID [constant 54](#)

kPMStatusFailed [constant 57](#)

kPMSyncRequestFailed [constant 57](#)

kPMUnsupportedConnection [constant 57](#)

kPMWriteFailed [constant 57](#)

kPrinterModuleTypeIDStr [constant 55](#)

O

Other Printer Module Constants [55](#)

P

PDE Feature Flags [51](#)

PDE Interface Identifier [52](#)

PDE Interface Version [52](#)

PDE Pane Kind Identifiers [53](#)

PDE Ticket Identifiers [54](#)

PDE Type Identifiers [54](#)

PlugInIntf [structure 43](#)

PlugInIntfVTable [structure 42](#)

PMBeginJobProcPtr [callback 13](#)

PMCancelJobProcPtr [callback 13](#)

PMCOMAddRefProcPtr [callback 13](#)

PMCOMQueryInterfaceProcPtr [callback 14](#)

PMCOMReleaseProcPtr [callback 14](#)

PMContext [data type 46](#)

- PMCreateLocalizedPaperSizeCFString **function** 7
- PMCreatePaperSizeCFString **function** 8
- PMCreatePrinterBrowserModuleInfoProcPtr **callback** 14
- PMCreatePrinterTicketsProcPtr **callback** 15
- PMCreatePrintingDialogExtensionsPathsProcPtr **callback** 15
- PMDrawingCtx **data type** 47
- PMEndJobProcPtr **callback** 15
- PMImageAccessProcPtr **callback** 16
- PMImageRef **data type** 47
- PMInitializeProcPtr **callback** 16
- PMInterface **structure** 45
- PMInterfaceAPIVersion **structure** 49
- PMInterfaceAPIVersionPtr **data type** 50
- PMInterfacePrBrowser **structure** 47
- PMInterfacePrBrowserPtr **data type** 48
- PMInterfaceRef **data type** 46
- PMIOCloseProcPtr **callback** 16
- PMIOGetAttributeProcPtr **callback** 17
- PMIOModuleCloseProcPtr **callback** 17
- PMIOModuleGetAttributeProcPtr **callback** 18
- PMIOModuleGetConnectionInfoProcPtr **callback** 18
- PMIOModuleInitializeProcPtr **callback** 18
- PMIOModuleOpenProcPtr **callback** 19
- PMIOModuleReadProcPtr **callback** 19
- PMIOModuleSetAttributeProcPtr **callback** 19
- PMIOModuleStatusProcPtr **callback** 20
- PMIOModuleTerminateProcPtr **callback** 20
- PMIOModuleWriteProcPtr **callback** 20
- PMIOOpenProcPtr **callback** 21
- PMIOProcs **structure** 46
- PMIOReadProcPtr **callback** 21
- PMIOSetAttributeProcPtr **callback** 22
- PMIOStatusProcPtr **callback** 22
- PMIOWriteProcPtr **callback** 23
- PMJobStreamGetNextBandProcPtr **callback** 23
- PMJobStreamGetPosProcPtr **callback** 24
- PMJobStreamOpenProcPtr **callback** 24
- PMJobStreamProcs **structure** 46
- PMJobStreamReadWriteProcPtr **callback** 24
- PMJobStreamSetPosProcPtr **callback** 25
- PMNotificationProcPtr **callback** 25
- PMPDECloseProcPtr **callback** 26
- PMPDEContext **data type** 44
- PMPDEFlags **data type** 44
- PMPDEGetSummaryTextProcPtr **callback** 26
- PMPDEInitializeProcPtr **callback** 27
- PMPDEOpenProcPtr **callback** 29
- PMPDEPrologueProcPtr **callback** 29
- PMPDERef **data type** 44
- PMPDESyncProcPtr **callback** 31
- PMPDETerminateProcPtr **callback** 32
- PMPPluginAPIVersion **structure** 41
- PMPPluginGetAPIVersionProcPtr **callback** 33
- PMPPluginHeader **structure** 40
- PMPPluginHeaderInterface **structure** 41
- PMPPluginReleaseProcPtr **callback** 33
- PMPPluginRetainProcPtr **callback** 34
- PMPPrBrowserAPIVersionProcPtr **callback** 34
- PMPPrBrowserCallbacks **structure** 48
- PMPPrBrowserCallbacksPtr **data type** 48
- PMPPrBrowserContext **data type** 49
- PMPPrBrowserFlags **data type** 49
- PMPPrBrowserGetLookupSpecProcPtr **callback** 34
- PMPPrBrowserGetSelectedPrintersProcPtr **callback** 35
- PMPPrBrowserInitializeProcPtr **callback** 35
- PMPPrBrowserPrologueProcPtr **callback** 36
- PMPPrBrowserRef **data type** 49
- PMPPrBrowserResizeProcPtr **callback** 36
- PMPPrBrowserSelectionStatusProcPtr **callback** 37
- PMPPrBrowserSyncProcPtr **callback** 37
- PMPPrBrowserSyncRequestProcPtr **callback** 38
- PMPPrBrowserTerminateProcPtr **callback** 38
- PMPPrBrowserWorksetPrintersProcPtr **callback** 38
- PMPrintJobProcPtr **callback** 39
- PMPrintPageProcPtr **callback** 39
- PMProcs **structure** 45
- PMTerminateProcPtr **callback** 40
- Print Center Feature Flags 56
- Print Center Signatures 56
- Printer Module Interface Version 55
- Printer Module Status Codes 55