

[Application Services](#) / Core Printing

API Collection

Core Printing

Overview

Core Printing is a C API that Mac apps and command line tools can use to perform printing tasks that don't display a user interface. Core Printing defines a set of opaque types and a rich set of operations on instances of these types. The Core Printing opaque types include:

- `PMPrintSession` for general information about a print job
- `PMPrintSettings` for print job parameters
- `PMPageFormat` for the page format of a printed document
- `PMPaper` for information about a type of paper
- `PMPrinter` for information about a printer

In Carbon applications, Core Printing is used together with Carbon Printing to implement printing features. For more information about Carbon Printing, see [Carbon Printing Reference](#).

In Cocoa applications, Core Printing can be used to extend the functionality in the Cocoa printing classes. The [`NSPrintInfo`](#) class provides direct access to some Core Printing objects.

Note

Core Printing is available to 64-bit applications, except for functions, data types, and constants that have been deprecated.

Topics

Releasing and Retaining Printing Objects

```
func PMRelease(PMObject?) -> OSStatus
```

Releases a printing object by decrementing its reference count.

```
func PMRetain(PMObject?) -> OSStatus
```

Retains a printing object by incrementing its reference count.

Creating and Using Page Format Objects

```
func PMCreatePageFormat(UnsafeMutablePointer<PMPageFormat?>) -> OSStatus
```

Creates a new page format object.

```
func PMCreatePageFormatWithPMPaper(UnsafeMutablePointer<PMPageFormat?>, PMPaper) -> OSStatus
```

Creates a page format object with a specified paper.

```
func PMCopyPageFormat(PMPageFormat, PMPageFormat) -> OSStatus
```

Copies the settings from one page format object into another.

```
func PMSessionDefaultPageFormat(PMPrintSession, PMPageFormat) -> OSStatus
```

Assigns default parameter values to a page format object used in the specified printing session.

```
func PMSessionValidatePageFormat(PMPrintSession, PMPageFormat, UnsafeMutablePointer<DarwinBoolean?>) -> OSStatus
```

Updates the values in a page format object and validates them against the current formatting printer.

```
func PMSessionCreatePageFormatList(PMPrintSession, PMPrinter?, UnsafeMutablePointer<Unmanaged<CFArray>?>) -> OSStatus
```

Obtains a list of page format objects, each of which describes a paper size available on the specified printer.

```
func PMPageFormatCreateDataRepresentation(PMPageFormat, UnsafeMutablePointer<Unmanaged<CFData>?>, PMDataFormat) -> OSStatus
```

Creates a data representation of a page format object.

```
func PMPageFormatCreateWithDataRepresentation(CFData, UnsafeMutablePointer<PMPageFormat?>) -> OSStatus
```

Creates a page format object from a data representation.

Accessing Data in Page Format Objects

```
func PMGetPageFormatExtendedData(PMPageFormat, OSType, UnsafeMutablePointer<UInt32?>, UnsafeMutableRawPointer?) -> OSStatus
```

Obtains extended page format data previously stored by your application.

```
func PMSetPageFormatExtendedData(PMPageFormat, OSType, UInt32, UnsafeMutableRawPointer) -> OSStatus
```

Stores your application-specific data in a page format object.

```
func PMGetPageFormatPaper(PMPageFormat, UnsafeMutablePointer<PMPaper?>) -> OSStatus
```

Obtains the paper associated with a page format object.

```
func PMPageFormatGetPrinterID(PMPageFormat, UnsafeMutablePointer<Unmanaged<CFString?>?) -> OSStatus
```

Obtains the identifier of the formatting printer for a page format object.

```
func PMGetOrientation(PMPageFormat, UnsafeMutablePointer<PMOrientation>) -> OSStatus
```

Obtains the current setting for page orientation.

```
func PMSetOrientation(PMPageFormat, PMOrientation, Bool) -> OSStatus
```

Sets the page orientation for printing.

```
func PMGetScale(PMPageFormat, UnsafeMutablePointer<Double>) -> OSStatus
```

Obtains the scaling factor currently applied to the page and paper rectangles.

```
func PMSetScale(PMPageFormat, Double) -> OSStatus
```

Sets the scaling factor for the page and paper rectangles.

```
func PMGetAdjustedPageRect(PMPageFormat, UnsafeMutablePointer<PMRect>) -> OSStatus
```

Obtains the imageable area or page rectangle, taking into account orientation, application drawing resolution, and scaling settings.

```
func PMGetAdjustedPaperRect(PMPageFormat, UnsafeMutablePointer<PMRect>) -> OSStatus
```

Obtains the rectangle defining the paper size, taking into account orientation, application drawing resolution, and scaling settings.

```
func PMGetUnadjustedPageRect(PMPageFormat, UnsafeMutable  
Pointer<PMRect>) -> OSStatus
```

Obtains the imageable area or page rectangle, unaffected by orientation, resolution, or scaling.

```
func PMGetUnadjustedPaperRect(PMPageFormat, UnsafeMutable  
Pointer<PMRect>) -> OSStatus
```

Obtains the paper rectangle, unaffected by rotation, resolution, or scaling.

Creating and Using Print Settings Objects

```
func PMCreatePrintSettings(UnsafeMutablePointer<PMPrintSettings?>) ->  
OSStatus
```

Creates a new print settings object.

```
func PMSessionDefaultPrintSettings(PMPrintSession, PMPrintSettings) ->  
OSStatus
```

Assigns default parameter values to a print settings object for the specified printing session.

```
func PMSessionValidatePrintSettings(PMPrintSession, PMPrintSettings,  
UnsafeMutablePointer<DarwinBoolean?>) -> OSStatus
```

Validates a print settings object within the context of the specified printing session.

```
func PMPrintSettingsCreateDataRepresentation(PMPrintSettings, Unsafe  
MutablePointer<Unmanaged<CFData>?>, PMDataFormat) -> OSStatus
```

Creates a data representation of a print settings object.

```
func PMPrintSettingsCreateWithDataRepresentation(CFData, UnsafeMutable  
Pointer<PMPrintSettings?>) -> OSStatus
```

Creates a print settings object from a data representation.

```
func PMCopyPrintSettings(PMPrintSettings, PMPrintSettings) -> OSStatus
```

Copies the settings from one print settings object into another.

```
func PMPrintSettingsToOptions(PMPrintSettings, UnsafeMutable  
Pointer<UnsafeMutablePointer<CChar>?>) -> OSStatus
```

Converts print settings into a CUPS options string.

```
func PMPrintSettingsToOptionsWithPrinterAndPageFormat(PMPrintSettings,
PMPrinter, PMPageFormat?, UnsafeMutablePointer<UnsafeMutable
Pointer<CChar>?>) -> OSStatus
```

Converts print settings and page format data into a CUPS options string for a specified printer.

Accessing Data in Print Settings Objects

```
func PMGetFirstPage(PMPrintSettings, UnsafeMutablePointer<UInt32>) ->
OSStatus
```

Obtains the number of the first page to be printed.

```
func PMSetFirstPage(PMPrintSettings, UInt32, Bool) -> OSStatus
```

Sets the default page number of the first page to be printed.

```
func PMGetLastPage(PMPrintSettings, UnsafeMutablePointer<UInt32>) ->
OSStatus
```

Obtains the number of the last page to be printed.

```
func PMSetLastPage(PMPrintSettings, UInt32, Bool) -> OSStatus
```

Sets the page number of the last page to be printed.

```
func PMGetPageRange(PMPrintSettings, UnsafeMutablePointer<UInt32>,
UnsafeMutablePointer<UInt32>) -> OSStatus
```

Obtains the valid range of pages that can be printed.

```
func PMSetPageRange(PMPrintSettings, UInt32, UInt32) -> OSStatus
```

Sets the valid range of pages that can be printed.

```
func PMPrintSettingsGetJobName(PMPrintSettings, UnsafeMutable
Pointer<Unmanaged<CFString>?>) -> OSStatus
```

Obtains the name of a print job.

```
func PMPrintSettingsSetJobName(PMPrintSettings, CFString) -> OSStatus
```

Specifies the name of a print job.

```
func PMGetCopies(PMPrintSettings, UnsafeMutablePointer<UInt32>) ->
OSStatus
```

Obtains the number of copies that the user requests to be printed.

```
func PMSetCopies(PMPrintSettings, UInt32, Bool) -> OSStatus
```

Sets the initial value for the number of copies to be printed.

```
func PMGetCollate(PMPrintSettings, UnsafeMutablePointer<DarwinBoolean>)
-> OSStatus
```

Obtains a Boolean value that indicates whether the job collate option is selected.

```
func PMSetCollate(PMPrintSettings, Bool) -> OSStatus
```

Specifies whether the job collate option is selected.

```
func PMGetDuplex(PMPrintSettings, UnsafeMutablePointer<PMDuplexMode>) -
> OSStatus
```

Obtains the selected duplex mode.

```
func PMSetDuplex(PMPrintSettings, PMDuplexMode) -> OSStatus
```

Sets the duplex mode.

```
func PMPrintSettingsGetValue(PMPrintSettings, CFString, UnsafeMutable
Pointer<Unmanaged<CTypeRef>?>) -> OSStatus
```

Obtains the value of a setting in a print settings object.

```
func PMPrintSettingsSetValue(PMPrintSettings, CFString, CTypeRef?,
Bool) -> OSStatus
```

Stores the value of a setting in a print settings object.

```
func PMPrintSettingsCopyAsDictionary(PMPrintSettings, UnsafeMutable
Pointer<Unmanaged<CFDictionary>?>) -> OSStatus
```

Creates a dictionary that contains the settings in a print settings object.

```
func PMPrintSettingsCopyKeys(PMPrintSettings, UnsafeMutable
Pointer<Unmanaged<CFArray>?>) -> OSStatus
```

Obtains the keys for items in a print settings object.

Creating Printing Session Objects

```
func PMCreateSession(UnsafeMutablePointer<PMPrintSession?>) -> OSStatus
```

Creates and initializes a printing session object and creates a context for printing operations.

Accessing Data in Printing Session Objects

```
func PMSessionGetDataFromSession(PMPrintSession, CFString, Unsafe
MutablePointer<Unmanaged<CTypeRef>?>) -> OSStatus
```

Obtains application-specific data previously stored in a printing session object.

```
func PMSessionSetDataInSession(PMPrintSession, CFString, CTypeRef) -> OSStatus
```

Stores your application-specific data in a printing session object.

```
func PMSessionGetCurrentPrinter(PMPrintSession, UnsafeMutablePointer<PMPrinter?>) -> OSStatus
```

Obtains the current printer associated with a printing session.

```
func PMSessionSetCurrentPMPrinter(PMPrintSession, PMPrinter) -> OSStatus
```

Changes the current printer for a printing session.

```
func PMSessionGetCGGraphicsContext(PMPrintSession, UnsafeMutablePointer<Unmanaged<CGContext>?>) -> OSStatus
```

Obtains the Quartz graphics context for the current page in a printing session.

```
func PMSessionError(PMPrintSession) -> OSStatus
```

Obtains the result code for any error returned by the printing session.

```
func PMSessionSetError(PMPrintSession, OSStatus) -> OSStatus
```

Sets the value of the current result code for the specified printing session.

Using Printer Presets

```
func PMPresetCopyName(PMPreset, UnsafeMutablePointer<Unmanaged<CFString>?>) -> OSStatus
```

Obtains the localized name for a preset.

```
func PMPresetCreatePrintSettings(PMPreset, PMPrintSession, UnsafeMutablePointer<PMPrintSettings?>) -> OSStatus
```

Creates a print settings object with settings that correspond to a preset.

```
func PMPresetGetAttributes(PMPreset, UnsafeMutablePointer<Unmanaged<CFDictionary>?>) -> OSStatus
```

Obtains the attributes of a preset.

Creating and Using Paper Objects

```
func PMPaperCreateCustom(PMPrinter?, CFString?, CFString?, Double, Double, UnsafePointer<PMPaperMargins>, UnsafeMutablePointer<PMPaper?>) -> OSStatus
```

Creates a custom paper object.

```
func PMPaperIsCustom(PMPaper) -> Bool
```

Returns a Boolean value indicating whether a specified paper is a custom paper.

Accessing Data in Paper Objects

```
func PMPaperGetID(PMPaper, UnsafeMutablePointer<Unmanaged<CFString>?>)
-> OSStatus
```

Obtains the identifier of a paper object.

```
func PMPaperGetWidth(PMPaper, UnsafeMutablePointer<Double>) -> OSStatus
```

Obtains the width of the sheet of paper represented by a paper object.

```
func PMPaperGetHeight(PMPaper, UnsafeMutablePointer<Double>) ->
OSStatus
```

Obtains the height of the sheet of paper represented by a paper object.

```
func PMPaperGetMargins(PMPaper, UnsafeMutablePointer<PMPaperMargins>) -
> OSStatus
```

Obtains the margins describing the unprintable area of the sheet represented by a paper object.

```
func PMPaperCreateLocalizedName(PMPaper, PMPrinter, UnsafeMutable
Pointer<Unmanaged<CFString>?>) -> OSStatus
```

Obtains the localized name for a given paper.

```
func PMPaperGetPrinterID(PMPaper, UnsafeMutable
Pointer<Unmanaged<CFString>?>) -> OSStatus
```

Obtains the printer ID of the printer to which a given paper corresponds.

```
func PMPaperGetPPDPaperName(PMPaper, UnsafeMutable
Pointer<Unmanaged<CFString>?>) -> OSStatus
```

Obtains the PPD paper name for a given paper.

Print Loop Functions

```
func PMSessionBeginCGDocumentNoDialog(PMPrintSession, PMPrintSettings,
PMPageFormat) -> OSStatus
```

Begins a print job that draws into a Quartz graphics context and suppresses the printing status dialog.

```
func PMSessionEndDocumentNoDialog(PMPrintSession) -> OSStatus
```

Ends a print job started by calling the function PMSessionBeginCGDocumentNoDialog(: : :) or PMSessionBeginDocumentNoDialog.

```
func PMSessionBeginPageNoDialog(PMPrintSession, PMPageFormat?, UnsafePointer<PMRect>?) -> OSStatus
```

Starts a new page for printing in the specified printing session and suppresses the printing status dialog.

```
func PMSessionEndPageNoDialog(PMPrintSession) -> OSStatus
```

Indicates the end of drawing the current page for the specified printing session.

Accessing the Print Job Destination

```
func PMSessionSetDestination(PMPrintSession, PMPrintSettings, PMDestinationType, CFString?, CFURL?) -> OSStatus
```

Sets the destination location, format, and type for a print job.

```
func PMSessionGetDestinationType(PMPrintSession, PMPrintSettings, UnsafeMutablePointer<PMDestinationType>) -> OSStatus
```

Obtains the output destination for a print job.

```
func PMSessionCopyDestinationFormat(PMPrintSession, PMPrintSettings, UnsafeMutablePointer<Unmanaged<CFString>?>) -> OSStatus
```

Obtains the destination format for a print job.

```
func PMSessionCopyDestinationLocation(PMPrintSession, PMPrintSettings, UnsafeMutablePointer<Unmanaged<CFURL>?>) -> OSStatus
```

Obtains a destination location for a print job.

```
func PMSessionCopyOutputFormatList(PMPrintSession, PMDestinationType, UnsafeMutablePointer<Unmanaged<CFArray>?>) -> OSStatus
```

Obtains an array of destination formats supported by the current print destination.

Creating Printer Objects

```
func PMServerLaunchPrinterBrowser(PMServer?, CFDictionary?) -> OSStatus
```

Launches the printer browser to browse the printers available for a print server.

```
func PMServerCreatePrinterList(PMServer?, UnsafeMutablePointer<Unmanaged<CFArray>?>) -> OSStatus
```

Creates a list of printers available to a print server.

```
func PMSessionCreatePrinterList(PMPrintSession, UnsafeMutablePointer<Unmanaged<CFArray>?>, UnsafeMutablePointer<CFIndex>?, UnsafeMutablePointer<PMPrinter?>?) -> OSStatus
```

Creates a list of printers available in the specified printing session.

```
func PMPrinterCreateFromPrinterID(CFString) -> PMPrinter?
```

Creates a printer object from a print queue identifier.

```
func PMCreateGenericPrinter(UnsafeMutablePointer<PMPrinter?>) -> OSStatus
```

Creates a generic printer object.

Accessing Information About a Printer

```
func PMPrinterCopyDescriptionURL(PMPrinter, CFString, UnsafeMutablePointer<Unmanaged<CFURL>?>) -> OSStatus
```

Obtains the URL of the description file for a given printer.

```
func PMPrinterCopyDeviceURI(PMPrinter, UnsafeMutablePointer<Unmanaged<CFURL>?>) -> OSStatus
```

Obtains the device URI of a given printer.

```
func PMPrinterCopyHostName(PMPrinter, UnsafeMutablePointer<Unmanaged<CFString>?>) -> OSStatus
```

Obtains the name of the server hosting the print queue for a given printer.

```
func PMPrinterCopyPresets(PMPrinter, UnsafeMutablePointer<Unmanaged<CFArray>?>) -> OSStatus
```

Obtains a list of print settings presets for a printer.

```
func PMPrinterGetCommInfo(PMPrinter, UnsafeMutablePointer<DarwinBoolean>?, UnsafeMutablePointer<DarwinBoolean>?) -> OSStatus
```

Obtains information about the communication channel for a printer.

```
func PMPrinterGetDriverCreator(PMPrinter, UnsafeMutablePointer<OSType>) -> OSStatus
```

Obtains the creator of the driver associated with the specified printer.

```
func PMPrinterGetID(PMPrinter) -> Unmanaged<CFString>?
```

Returns the unique identifier of a printer.

```
func PMPrinterGetLocation(PMPrinter) -> Unmanaged<CFString>?
```

Returns the location of a printer.

```
func PMPrinterGetMakeAndModelName(PMPrinter, UnsafeMutable  
Pointer<Unmanaged<CFString>?>) -> OSStatus
```

Obtains the manufacturer and model name of the specified printer.

```
func PMPrinterGetMimeTypes(PMPrinter, PMPrintSettings?, UnsafeMutable  
Pointer<Unmanaged<CFArray>?>) -> OSStatus
```

Obtains a list of MIME content types supported by a printer using the specified print settings.

```
func PMPrinterGetName(PMPrinter) -> Unmanaged<CFString>?
```

Returns the human-readable name of a printer.

```
func PMPrinterGetOutputResolution(PMPrinter, PMPrintSettings, Unsafe  
MutablePointer<PMResolution>) -> OSStatus
```

Obtains the printer hardware output resolution for the specified print settings.

```
func PMPrinterSetOutputResolution(PMPrinter, PMPrintSettings, Unsafe  
Pointer<PMResolution>) -> OSStatus
```

Sets the print settings to reflect the specified printer hardware output resolution.

```
func PMPrinterGetPaperList(PMPrinter, UnsafeMutable  
Pointer<Unmanaged<CFArray>?>) -> OSStatus
```

Obtains the list of papers available for a printer.

```
func PMPrinterGetPrinterResolutionCount(PMPrinter, UnsafeMutable  
Pointer<UInt32>) -> OSStatus
```

Obtains the number of resolution settings supported by the specified printer.

```
func PMPrinterGetIndexedPrinterResolution(PMPrinter, UInt32, Unsafe  
MutablePointer<PMResolution>) -> OSStatus
```

Obtains a resolution setting based on an index into the range of settings supported by the specified printer.

```
func PMPrinterGetState(PMPrinter, UnsafeMutablePointer<PMPrinterState>)  
-> OSStatus
```

Obtains the current state of the print queue for a printer.

```
func PMPrinterSetDefault(PMPrinter) -> OSStatus
```

Sets the default printer for the current user.

```
func PMPrinterIsDefault(PMPrinter) -> Bool
```

Returns a Boolean value indicating whether a printer is the default printer for the current user.

```
func PMPrinterIsFavorite(PMPrinter) -> Bool
```

Returns a Boolean value indicating whether a printer is in the user's list of favorite printers.

```
func PMPrinterIsPostScriptCapable(PMPrinter) -> Bool
```

Returns a Boolean value indicating whether a printer is PostScript capable.

```
func PMPrinterIsPostScriptPrinter(PMPrinter, UnsafeMutable  
Pointer<DarwinBoolean>) -> OSStatus
```

Determines whether a printer is a PostScript printer.

```
func PMPrinterIsRemote(PMPrinter, UnsafeMutablePointer<DarwinBoolean>)  
-> OSStatus
```

Indicates whether a printer is hosted by a remote print server.

Submitting a Print Job to a Printer

```
func PMPrinterPrintWithFile(PMPrinter, PMPrintSettings, PMPageFormat?,  
CFString?, CFURL) -> OSStatus
```

Submits a print job to a specified printer using a file that contains print data.

```
func PMPrinterPrintWithProvider(PMPrinter, PMPrintSettings, PMPage  
Format?, CFString, CGDataProvider) -> OSStatus
```

Submits a print job to a specified printer using a Quartz data provider to obtain the print data.

Accessing PostScript Printer Description Files

```
func PMCopyAvailablePPDs(PMPPDDomain, UnsafeMutable  
Pointer<Unmanaged<CFArray>?>) -> OSStatus
```

Obtains the list of PostScript printer description (PPD) files in a PPD domain.

```
func PMCopyLocalizedPPD(CFURL, UnsafeMutablePointer<Unmanaged<CFURL>?>)  
-> OSStatus
```

Obtains a localized PostScript printer description (PPD) file.

```
func PMCopyPPDData(CFURL, UnsafeMutablePointer<Unmanaged<CFData>?>) ->  
OSStatus
```

Obtains the uncompressed PPD data for a PostScript printer description (PPD) file.

Printing with PostScript Data

```
func PMCGImageCreateWithEPSDataProvider(CGDataProvider?, CGImage) ->
Unmanaged<CGImage>?
```

Creates an image that references both the PostScript contents of EPS data and a preview (proxy) image for the data.

```
func PMPrinterWritePostScriptToURL(PMPrinter, PMPrintSettings, PMPage
Format?, CFString?, CFURL, CFURL) -> OSStatus
```

Converts an input file of the specified MIME type to printer-ready PostScript for a destination printer.

Using PDF Workflow Items

```
func PMWorkflowCopyItems(UnsafeMutablePointer<Unmanaged<CFArray>?>) ->
OSStatus
```

Obtains an array of the available PDF workflow items.

```
func PMWorkflowSubmitPDFWithOptions(CFURL, CFString?, Unsafe
Pointer<CChar>?, CFURL) -> OSStatus
```

Submits a PDF file for workflow processing using the specified CUPS options string.

```
func PMWorkflowSubmitPDFWithSettings(CFURL, PMPrintSettings, CFURL) ->
OSStatus
```

Submits a PDF file for workflow processing using the specified print settings.

Data Types

```
typealias PMObject
```

The base type for all the opaque types used in Core Printing.

```
typealias PMPageFormat
```

An opaque type that stores the settings in the Page Setup dialog.

```
typealias PMPaper
```

An opaque type that stores information about the paper used in a print job.

```
typealias PMPaperMargins
```

A data structure that specifies the unprintable area of a paper object.

`typedef` `PMPreset`

An opaque type that stores information about a named preset available for a print job.

`typedef` `PMPrinter`

An opaque type that represents a printer.

`typedef` `PMPrintSession`

An opaque type that stores information about a print job.

`typedef` `PMPrintSettings`

An opaque type that stores the settings in the Print dialog.

`typedef` `PMServer`

An opaque type that identifies a local or remote print server.

Constants

`struct` `PMDataFormat`

Constants that specify the format of the data representation created with the functions `PMPageFormatCreateDataRepresentation(_ : : :)` and `PMPrintSettingsCreateDataRepresentation(_ : : :)`.

`typedef` `PMDestinationType`

Constants that specify a destination for a print job.

`typedef` `PMDuplexMode`

Constants that specify duplex mode settings.

`typedef` `PMOrientation`

Constants that specify page orientation.

⌵ PDF Workflow Dictionary Keys

Constants that specify the keys in a PDF workflow dictionary.

`typedef` `PMPPDDomain`

Constants that specify the domains for PostScript printer description (PPD) files.

⌵ Print All Pages Constant

A constant that specifies that all pages of a document should be printed.

`typedef` `PMQualityMode`

Constants that specify standard options for print quality.

`typealias PMPrinterState`

Constants that specify the current state of a print queue.

☰ Printer Description Types

Constants that specify printer description types.

☰ User Cancellation Constant

A constant that specifies an error value that indicates the user canceled a printing operation.

Result Codes

This table lists the result codes defined for Core Printing.

`var kPMGeneralError: Int`

An unspecified error occurred.

`var kPMOutOfScope: Int`

Your application called this function out of sequence with other printing functions.

`var kPMNoDefaultPrinter: Int`

The user has not specified a default printer.

`var kPMNotImplemented: Int`

The function is not implemented.

`var kPMNoSuchEntry: Int`

There is no entry to match your application's request.

`var kPMInvalidPrintSettings: Int`

Your application passed an invalid print settings object.

`var kPMInvalidPageFormat: Int`

Your application passed an invalid page format object.

`var kPMValueOutOfRange: Int`

Your application passed an out-of-range value.

`var kPMInvalidPrintSession: Int`

Your application passed an invalid printing session object.

`var kPMInvalidPrinter: Int`

Your application passed an invalid printer object.

`var kPMObjectInUse: Int`

The specified object is in use.

`var kPMInvalidIndex: Int`

An array index is invalid.

`var kPMStringConversionFailure: Int`

An internal error occurred while converting a string.

`var kPMXMLParseError: Int`

An error occurred while parsing XML data.

`var kPMInvalidJobTemplate: Int`

An internal error occurred while creating a job template.

`var kPMInvalidPrinterInfo: Int`

The printer information is invalid.

`var kPMInvalidConnection: Int`

The printer connection type is invalid.

`var kPMInvalidKey: Int`

The key in a ticket, job template, or dictionary is invalid.

`var kPMInvalidValue: Int`

The value in a ticket, job template, or dictionary is missing.

`var kPMInvalidAllocator: Int`

The specified memory allocator is invalid.

`var kPMInvalidTicket: Int`

The job ticket is invalid.

`var kPMInvalidItem: Int`

The item being added to a ticket is invalid.

`var kPMInvalidType: Int`

The data type in a ticket, job template, or dictionary is not the expected type.

`var kPMInvalidReply: Int`

A remote server or client sent an invalid reply.

```
var kPMInvalidFileType: Int
```

The file type is invalid.

```
var kPMInvalidObject: Int
```

The object is invalid.

```
var kPMInvalidPaper: Int
```

Your application passed an invalid paper object.

```
var kPMInvalidCalibrationTarget: Int
```

The dictionary specifying a printer calibration target is invalid.

```
var kPMInvalidPreset: Int
```

Your application passed an invalid preset object.