
NSPrinter

Inherits From:	NSObject
Conforms To:	NSCoding NSCopying NSObject (NSObject)
Declared In:	AppKit/NSPrinter.h

Class Description

An NSPrinter object describes a printer's capabilities, such as whether the printer can print in color and whether it provides a particular font. An NSPrinter object represents either a particular make or type of printer, or an actual printer available to the computer. You use NSPrinter to get information about printers, not modify printer attributes or control a printing job.

There are two ways to create an NSPrinter:

- To create an abstract object that provides information about a type of printer rather than an object that represents an actual printer device, use the **printerWithType:** class method, passing a printer type (an NSString) as the argument. The **printerTypes** class method provides a list of printer types, model names recognized by the computer. Printer types are described in files written in PostScript Printer Description (PPD) format. The location of these files is platform dependent.
- Use the **printerWithName:** class method to create or find an NSPrinter that corresponds to an actual printer device. Use the **printerNames** class method to get a list of recognized printer names.

Once you have an NSPrinter, there's only one thing you can do with it: retrieve information regarding the type of printer or regarding the actual printer the object represents.

When you create an NSPrinter object, the object reads the file that corresponds to the type of printer, a model name, you specified and stores the data it finds there in named tables. Printer types are described in files written in the PostScript Printer Description (PPD) format. Any piece of information in the PPD tables can be retrieved through the methods **stringForKey:inTable:** and **stringListForKey:inTable:**, as explained later. Commonly needed items, such as whether a printer supports color or the size of the page on which it prints, are available through more direct methods (methods such as **isColor** and **pageSizeForPaper:**).

Note: To understand what the NSPrinter tables contain, you need to be acquainted with the PPD file format. This is described in *PostScript Printer Description File Format Specification, version 4.0*, available from Adobe Systems Incorporated. The rest of this class description assumes a familiarity with the concepts and terminology presented in the Adobe manual. A brief summary of the PPD format is given below; PPD terms defined in the Adobe manual are shown in *italic*.

PPD Format

A PPD file statement, or *entry*, associates a value with a main keyword:

```
*mainKeyword: value
```

The asterisk is literal; it indicates the beginning of a new entry.

For example:

```
*modelName: "MMimeo Machine"  
*3dDevice: False
```

A main keyword can be qualified by an *option keyword*:

```
*mainKeyword optionKeyword: value
```

For example:

```
*PaperDensity Letter: "0.1"  
*PaperDensity Legal: "0.2"  
*PaperDensity A4: "0.3"  
*PaperDensity B5: "0.4"
```

In addition, any number of entries may have the same main keyword with no option keyword yet give different values:

```
*InkName: ProcessBlack/Process Black  
*InkName: CustomColor/Custom Color  
*InkName: ProcessCyan/Process Cyan  
*InkName: ProcessMagenta/Process Magenta  
*InkName: ProcessYellow/Process Yellow
```

Option keywords and values can sport *translation strings*. A translation string is a textual description, appropriate for display in a user interface, of the option or value. An option or value is separated from its translation string by a slash:

```
*Resolution 300dpi/300 dpi: " ... "  
*InkName: ProcessBlack/Process Black
```

In the first example, the **300dpi** option would be presented in a user interface as “300 dpi.” In the second example, the translation string for the **ProcessBlack** value is set to “Process Black”.

NSPrinter treats entries that have an ***OrderDependency** or ***UIConstraint** main keyword specially. Such entries take the following forms (the bracketed elements are optional):

```
*OrderDependency: real section mainKeyword [optionKeyword]
```

```
*UIConstraint: mainKeyword1 [optionKeyword1] mainKeyword2 [optionKeyword2]
```

There may be more than one UIConstraint entry with the same *mainKeyword1* or *mainKeyword1/optionKeyword1* value. Below are some examples of ***OrderDependency** and ***UIConstraint** entries:

```
*OrderDependency: 10 AnySetup *Resolution
*UIConstraint: *Option3 None *PageSize Legal
*UIConstraint: *Option3 None *PageRegion Legal
```

Explaining these entries is beyond the scope of this documentation; however, it's important to note their forms in order to understand how they're represented in the NSPrinter tables.

NSPrinter Tables

NSPrinter defines five key-value tables to store PPD information. The tables are identified by the names given below:

Name	Contents
PPD	General information about a printer type. This table contains the values for all entries in a PPD file except those with the *OrderDependency and *UIConstraint main keywords. The values in this table don't include the translation strings.
PPDOptionTranslation	Option keyword translation strings.
PPDArgumentTranslation	Value translation strings.
PPDOrderDependency	*OrderDependency values.
PPDUIConstraints	*UIConstraint values.

There are two principle methods for retrieving data from the NSPrinter tables:

- **stringForKey:inTable:** returns the value for the first occurrence of a given key in the given table.
- **stringListForKey:inTable:** returns an array of values, one for each occurrence of the key.

For both methods, the first argument is an NSString that names a key—which part of a PPD file entry the key corresponds to depends on the table (as explained in the following sections). The second argument names the table that you want to search. The values that are returned by these methods, whether singular or in an array, are always NSStrings, even if the value wasn't a quoted string in the PPD file.

The NSPrinter tables store data as ASCII text, thus the two methods described above are sufficient for retrieving any value from any table. NSPrinter provides a number of other methods, such as **booleanForKey:inTable:** and **intForKey:inTable:**, that retrieve single values and coerce them, if possible, into particular data types. The coercion doesn't affect the data that's stored in the table (it remains in ASCII format).

To check the integrity of a table, use the **isKey:forTable:** and **statusForTable:** methods. The former returns a boolean that indicates whether the given key is valid for the given table; the latter returns an error code that describes the general state of a table (in particular, whether it actually exists).

Retrieving Values from the PPD Table

Keys for the PPD table are strings that name a main keyword or main keyword/option keyword pairing (formatted as “*mainKeyword/optionKeyword*”). In both cases, you exclude the main keyword asterisk. The following example creates an NSPrinter and invokes **stringForKey:inTable:** to retrieve the value for an un-optioned main keyword:

```
/* Create an NSPrinter object for a printer type. */
NSPrinter *prType = [NSPrinter printerWithType:@"My_Mimeo_Machine"]

/* Sets sValue to FALSE. */
NSString *sValue = [prType stringForKey:@"3dDevice" inTable:@"PPD"];
```

To retrieve the value for a main keyword/option keyword pair, pass the keywords formatted as “*mainKeyword/optionKeyword*”:

```
/* Sets sValue to "0.3". */
NSString *sValue = [prType stringForKey:@"PaperDensity/A4" inTable:@"PPD"];
```

stringForKey:inTable: can determine if a main keyword has options. If you pass a main keyword (only) as the first argument to the method, and if that keyword has options in the PPD file, the method returns an empty string. If it doesn't have options, it returns the value of the first occurrence of the main keyword:

```
/* Sets sValue to an empty string. */
NSString *sValue = [prType stringForKey:@"PaperDensity" inTable:@"PPD"];

/* Sets sValue to "ProcessBlack". */
NSString *sValue = [prType stringForKey:@"InkName" inTable:@"PPD"];
```

To retrieve the values for all occurrences of a main keyword, use the **stringListForKey:inTable:** method giving the main keyword only:

```
/* Sets sList to an array containing "ProcessBlack", "CustomColor", etc. */
NSArray *sList = [prType stringListForKey:@"InkName" inTable:@"PPD"];
```

In addition, **stringListForKey:inTable:** can be used to retrieve all the options for a main keyword (given that the main keyword has options):

```
/* Sets sList to an array containing "Letter", "Legal", "A4", etc. */
NSArray *sList = [prType stringListForKey:@"PaperDensity" inTable:@"PPD"];
```

Retrieving Values from the Option and Argument Translation Tables

A key to a translation table is similar to a key to the PPD table: It's a main keyword or main/option keyword pair (again excluding the asterisk). However, the values that are returned from the translation tables are the translation strings for the option or argument (value) portions of the PPD file entry. For example:

```
/* Sets sValue to "300 dpi". */
NSString *sValue = [prType stringForKey:@"Resolution/300dpi"
                                inTable:@"PPDOptionTranslation"];
```

```

/* Sets sList to an array containing "Process Black", "Custom Color", etc. */
NSArray *sList = [prType stringListForKey:@"InkName"
                    inTable:@"PPDArgumentTranslation"];

```

As with the PPD table, use **stringListForKey:inTable:** to request an array of all occurrences of a main keyword.

Retrieving Values from the Order Dependency Table

As mentioned earlier, an order dependency entry takes this form:

```
*OrderDependency: real section mainKeyword [optionKeyword]
```

These entries are stored in the PPDOrderDependency table. To retrieve a value from this table, always use **stringListForKey:inTable:**. The value passed as the key is, again, a main keyword or main keyword/option keyword pair; however, these values correspond to the *mainKeyword* and *optionKeyword* parts of an order dependency entry's value. As with the other tables, the main keyword's asterisk is excluded. The method returns an NSArray of two NSStrings that correspond to the *real* and *section* values for the entry. For example:

```

/* Sets sList to an array containing "10" and "AnySetup". */
NSArray *sList = [prType stringListForKey:@"Resolution"
                    inTable:@"PPDOrderDependency"];

```

Retrieving Values from the UIConstraints Table

Retrieving a value from the PPDUIConstraints table is similar to retrieving a value from the PPDOrderDependency table: always use **stringListForKey:inTable:** and the key corresponds to elements in the entry's value. Given the following form (as described earlier), the key corresponds to *mainKeyword1/optionKeyword1*:

```
*UIConstraint: mainKeyword1 [optionKeyword1] mainKeyword2 [optionKeyword2]
```

The NSArray that's returned by **stringListForKey:inTable:** contains the *mainKeyword2* and *optionKeyword2* values (with the keywords stored as separate elements in the NSArray) for every ***UIConstraints** entry that has the given *mainKeyword1/optionKeyword1* value. For example:

```

/* Sets sList to an array containing:
   "PageSize", "Legal", "PageRegion", and "Legal" */
NSArray *sList = [prType stringListForKey:@"Option3/None"
                    inTable:@"PPDUIConstraints"];

```

Note that the main keywords that are returned in the NSArray don't have asterisks. Also, the NSArray that's returned always alternates main and option keywords. If a particular main keyword doesn't have an option associated with it, the string for the option will be empty (but the entry in the NSArray for the option *will* exist).

Adopted Protocols

NSCoding	– encodeWithCoder: – initWithCoder:
NSCopying	– copyWithZone:

Method Types

Creating an NSPrinter	+ printerWithName: + printerWithName:domain:includeUnavailable: + printerWithType:
Getting general printer information	+ printerNames + printerTypes
Getting attributes	– domain – host – name – note – type
Getting specific information	– acceptsBinary – imageRectForPaper: – pageSizeForPaper: – isColor – isFontAvailable: – isOutputStackInReverseOrder – languageLevel
Querying the tables	– isKey:inTable: – stringForKey:inTable: – stringListForKey:inTable: – booleanForKey:inTable: – floatForKey:inTable: – intForKey:inTable: – rectForKey:inTable: – sizeForKey:inTable: – statusForTable: – deviceDescription

Class Methods

printerNames

+ (NSArray *)printerNames

Returns an array of recognized printer names.

See also: +printerTypes, – name

printerTypes

+ (NSArray *)printerTypes

Returns an array of recognized model names.

See also: +printerNames, – type

printerWithName:

+ (NSPrinter *)printerWithName:(NSString *)name

Returns an NSPrinter that represents an actual printer with the given *name*. Returns **nil** if the specified printer is not available.

See also: + printerWithType:, + printerNames, – name



printerWithName:domain:includeUnavailable:

+ (NSPrinter *)printerWithName:(NSString *)name domain:(NSString *)domain
includeUnavailable:(BOOL)flag

This method is for Mach platforms only—it is not defined for other platforms. Returns an NSPrinter that represents an actual printer with the given *name* and *domain*. If *domain* is **nil**, the first printer (matching *name*) found on any host or domain is used. Returns **nil** if the specified printer is not available and *flag* is NO. If *flag* is YES, the availability of the printer is ignored.

See also: + printerWithName:, + printerWithType:, + printerNames, – domain, – name

printerWithType:

+ (NSPrinter *)printerWithType:(NSString *)type

Returns an NSPrinter with the given printer *type*.

See also: + printerWithName:, + printerTypes, – type

Instance Methods

acceptsBinary

– (BOOL)**acceptsBinary**

Returns YES if the receiver accepts binary PostScript, otherwise NO.

booleanForKey:inTable:

– (BOOL)**booleanForKey:(NSString *)key inTable:(NSString *)table**

Returns a boolean value associated with *key* in *table*. Will also return NO if *key* is not in *table*.

See also: – **isKey:inTable:**, – **stringForKey:inTable:**

copyWithZone:

@protocol NSCopying

– (id)**copyWithZone:(NSZone *)zone**

Doesn't return a copy of the receiver. Returns the receiver with its reference count incremented (sends **retain** to the receiver).

deviceDescription

– (NSDictionary *)**deviceDescription**

Returns a dictionary of keys and values describing the device. See NSGraphics.h for possible keys.



domain

– (NSString *)**domain**

This method is for Mach platforms only—it is not defined for other platforms. Returns the name of the domain in which the receiver's printer resides. Returns **nil** if the receiver doesn't represent an actual printer.

See also: + **printerWithName:domain:includeUnavailable:**

floatForKey:inTable:

– (float)**floatForKey:(NSString *)key inTable:(NSString *)table**

Returns a floating-point value associated with *key* in *table*. Returns 0.0 if *key* is not in *table*.

See also: – **isKey:inTable:**, – **stringForKey:inTable:**

host

– (NSString *)**host**

Returns the name of the receiver's host computer.

imageRectForPaper:

– (NSRect)**imageRectForPaper:(NSString *)paperName**

Returns the printing rectangle for the paper *paperName*. Possible values for *paperName* are contained in the printer's PPD file. Typical values are Letter and Legal.

See also: – **pageSizeForPaper:**

intForKey:inTable:

– (int)**intForKey:(NSString *)key inTable:(NSString *)table**

Returns an integer value associated with *key* in *table*. Returns 0 if *key* is not in *table*.

See also: – **isKey:inTable:**, – **stringForKey:inTable:**

isColor

– (BOOL)**isColor**

Returns YES if the receiver can print color, otherwise NO.

isFontAvailable:

– (BOOL)**isFontAvailable:(NSString *)faceName**

Returns YES if font *faceName* is available to the receiver, otherwise NO.

isKey:inTable:

– (BOOL)**isKey:**(NSString *)*key* **inTable:**(NSString *)*table*

Returns YES if *key* is in *table*, otherwise NO.

isOutputStackInReverseOrder

– (BOOL)**isOutputStackInReverseOrder**

Returns YES if the receiver outputs pages in reverse page order, otherwise NO.

languageLevel

– (int)**languageLevel**

Returns the PostScript Language Level recognized by the receiver.

name

– (NSString *)**name**

Returns the receiver's name.

See also: + **printerNames**, + **printerWithName:**

note

– (NSString *)**note**

Returns the note associated with the receiver.

pageSizeForPaper:

– (NSSize)**pageSizeForPaper:**(NSString *)*paperName*

Returns the size of the page for the paper type *paperName*. Possible values for *paperName* are contained in the printer's PPD file. Typical values are Letter and Legal.

See also: – **imageRectForPaper:**

rectForKey:inTable:

– (NSRect)**rectForKey:(NSString *)key inTable:(NSString *)table**

Returns the rectangle associated with *key* in *table*. Returns NSZeroRect if *key* is not in *table*.

See also: – **isKey:inTable:**, – **stringForKey:inTable:**

sizeForKey:inTable:

– (NSSize)**sizeForKey:(NSString *)key inTable:(NSString *)table**

Returns the size associated with *key* in *table*. The returned width and height is 0.0 if *key* is not in *table*.

See also: – **isKey:inTable:**, – **stringForKey:inTable:**

statusForTable:

– (NSPrinterTableStatus)**statusForTable:(NSString *)table**

Returns the status of *table*:

NSPrinterTableOK

NSPrinterTableNotFound

NSPrinterTableError

stringForKey:inTable:

– (NSString *)**stringForKey:(NSString *)key inTable:(NSString *)table**

Returns the first occurrence of a value associated with *key* in *table*. If *key* is a main keyword only, and if that keyword has options in the PPD file, this method returns an empty string. Use **stringListForKey:inTable:** to retrieve the values for all occurrences of a main keyword. Returns **nil** if *key* is not in *table*.

See also: – **isKey:inTable:**, – **booleanForKey:inTable:**, – **floatForKey:inTable:**, – **intForKey:inTable:**, – **rectForKey:inTable:**, – **sizeForKey:inTable:**

stringListForKey:inTable:

– (NSArray *)**stringListForKey:(NSString *)key inTable:(NSString *)table**

Returns an array of strings, one for each occurrence, associated with *key* in *table*. Returns **nil** if *key* is not in *table*.

See also: – **isKey:inTable:**, – **stringForKey:inTable:**

type

– (NSString *)**type**

Returns the name of the receiver's type.

See also: + **printerTypes**