Ticket Services Reference

Printing > Carbon

2002-10-23

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Simultaneously published in the United States and Canada.

Contents

Ticket Services Reference

Framework:	ApplicationServices/ApplicationServices.h, Carbon/Carbon.h
Declared in	PMDefinitions.h PMDefinitionsDeprecated.h PMTemplate.h PMTicket.h PMTicketDeprecated.h

Overview

Ticket Services provides the functions and data types used to communicate printing information (page format and print settings) among the various modules in the printing system. Developers who write printing dialog extensions and printer modules need to use the functions described in this reference and also need to consult the *Printing Plug-in Interfaces Reference*. Developers who write applications do not need this reference; see the *Carbon Printing Manager Reference* instead.

Functions by Task

Managing Tickets

`PMTicketCreate` (page 51)
Creates a new ticket.

`PMTicketValidate` (page 97)
Validates a ticket against the constraint values specified in a job template.

`PMTicketRetain` (page 76)
Increments the retention count of a ticket object.

`PMTicketRemoveTicket` (page 75)
Removes a subticket from a ticket.

`PMTicketRelease` (page 74)
Decrements the retention count of a ticket object.

`PMTicketReleaseAndClear` (page 74)
Decrements the retention count of a ticket and sets the ticket reference to `NULL`.

`PMTicketCopy` (page 49)
Copies a ticket.

`PMTicketCreateTemplate` (page 51)
Retrieves a template from a ticket.

`PMTicketCopyItem` (page 50)
Copies an item from one ticket to another ticket.

`PMTicketReleaseItem` (page 75)
Removes an item from a ticket.

`PMTicketDeleteItem` (page 52)
Makes an item in a ticket unavailable.

`PMTicketLockItem` (page 72)
Locks an item in a ticket.

`PMTicketUnlockItem` (page 96)
Unlocks an item in a ticket.

`PMTicketIsItemLocked` (page 72)
Checks to see if an item in a ticket is locked.

`PMTicketConfirmTicket` (page 48)
Checks whether a ticket appears to be valid.

`PMTicketContainsItem` (page 48)
Checks whether an item exists in a ticket.

`PMTicketContainsTicket` (page 49)
Checks whether a ticket is contained in another ticket.

`PMTicketFillFromArray` (page 53)
Adds items defined in an array of ticket item structures to a ticket.

Setting Ticket Items

`PMTicketSetBoolean` (page 76)
Writes an item of type `Boolean` to a ticket.

`PMTicketSetBytes` (page 77)
Writes an item that's an array of type `UInt8` to a ticket.

`PMTicketSetCFArray` (page 78)
Writes an item of type `CFArray` to a ticket.

`PMTicketSetCFBoolean` (page 78)
Writes an item of type `CFBoolean` to a ticket.

`PMTicketSetCFData` (page 79)
Writes an item of type `CFData` to a ticket.

`PMTicketSetCFDate` (page 80)
Writes an item of type `CFDate` to a ticket.

`PMTicketSetCFDictionary` (page 81)
Writes an item of type `CFDictionary` to a ticket.

`PMTicketSetCFNumber` (page 81)
Writes an item of type `CFNumber` to a ticket.

`PMTicketSetCFString` (page 82)
Writes an item of type `CFString` to a ticket.

`PMTicketSetCString` (page 83)
Writes an item that's a C-style string to a ticket.

`PMTicketSetCStringArray` (page 84)
Writes an item that's an array of C-style strings to a ticket.

`PMTicketSetDouble` (page 84)
Writes an item of type `double` to a ticket.

`PMTicketSetDoubleArray` (page 85)
Writes an item that's an array of values of type `double` to a ticket.

`PMTicketSetItem` (page 86)
Writes an item of type `CFType` to a ticket.

`PMTicketSetPMRect` (page 87)
Writes an item of type `PMRect` to a ticket.

`PMTicketSetPMRectArray` (page 88)
Writes an item that's an array of values of type `PMRect` to a ticket.

`PMTicketSetPMResolution` (page 89)
Writes an item of type `PMResolution` to a ticket.

`PMTicketSetPMResolutionArray` (page 90)
Writes an item that's an array of data of type `PMResolution` to a ticket.

`PMTicketSetPString` (page 91)
Writes an item that's a Pascal-style string to a ticket.

`PMTicketSetSInt32` (page 91)
Writes an item of type `SInt32` to a ticket.

`PMTicketSetSInt32Array` (page 92)
Writes an item that's an array of data of type `SInt32` to a ticket.

`PMTicketSetTemplate` (page 93)
Writes an item that's a job template to a ticket.

`PMTicketSetTicket` (page 93)
Writes a subticket to a ticket.

`PMTicketSetUInt32` (page 94)
Writes an item of type `UInt32` to a ticket.

`PMTicketSetUInt32Array` (page 95)
Writes an item that's an array of data of type `UInt32` to a ticket.

Getting Ticket Items

`PMTicketGetAllocator` (page 53)
Obtains the allocator object used for allocating memory for a ticket.

`PMTicketGetAPIVersion` (page 54)
Obtains the version of the application programming interface (API) used to created a ticket.

`PMTicketGetBoolean` (page 54)
Obtains data for a ticket item of type `Boolean`.

`PMTicketGetBytes` (page 55)
Obtains data for a ticket item that's an array of `UInt8` values.

`PMTicketGetCFArray` (page 56)
Obtains data for a ticket item that's a Core Foundation array.

PMTicketGetCFBoolean (page 57)
Obtains data for a ticket item that's an array of CFBoolean values.

PMTicketGetCFData (page 57)
Obtains data for a ticket item of type CFData.

PMTicketGetCFDate (page 58)
Obtains data for a ticket item of type CFDate.

PMTicketGetCFDictionary (page 59)
Obtains data for a ticket item of type CFDictionary.

PMTicketGetCFNumber (page 60)
Obtains data for a ticket item of type CFNumber.

PMTicketGetCFString (page 60)
Obtains data for a ticket item of type CFString.

PMTicketGetCString (page 61)
Obtains data for a ticket item of that's a C-style string.

PMTicketGetDouble (page 62)
Obtains data for a ticket item of type double.

PMTicketGetEnumType (page 62)
Obtains a ticket's ticket type (job, paper, converter, and so on).

PMTicketGetIndexPMResolution (page 63)
Obtains an indexed resolution for a ticket item that's an array of type PMResolution.

PMTicketGetItem (page 64)
Obtains data for a ticket item of type CFType.

PMTicketGetLockedState (page 64)
Obtains the lock state of a ticket.

PMTicketGetPMRect (page 66)
Obtains data for a ticket item of type PMRect.

PMTicketGetPMResolution (page 66)
Obtains data for a ticket item of type PMResolution.

PMTicketGetPPDDict (page 67)
Obtains data for a ticket item that's a PostScript printer description (PPD) dictionary.

PMTicketGetPString (page 68)
Obtains data for a ticket item that's a Pascal-style string.

PMTicketGetRetainCount (page 69)
Obtains the retention count for a ticket.

PMTicketGetSInt32 (page 69)
Obtains data for a ticket item of type SInt32.

PMTicketGetTicket (page 70)
Obtains a subticket from a ticket.

PMTicketGetType (page 71)
Obtains a value that specifies the ticket's type.

PMTicketGetUInt32 (page 71)
Obtains an item of type UInt32 from a ticket.

Managing Templates

`PMTemplateCreate` (page 16)
Creates a new job template.

`PMTemplateDelete` (page 17)
Deletes an existing job template.

`PMTemplateMakeEntry` (page 33)
Creates a new entry in a job template.

`PMTemplateMakeFullEntry` (page 33)
Creates a new entry in a job template and sets the default and constraint values for the entry.

`PMTemplateMergeTemplates` (page 34)
Merges the entries from one job template with entries from another job template.

`PMTemplateRemoveEntry` (page 35)
Removes an entry from a job template.

`PMTemplateValidateItem` (page 47)
Validates an item in a job template.

`PMTemplateIsLocked` (page 31)
Checks to see if a job template is locked.

Setting Template Items

`PMTemplateSetBooleanDefaultValue` (page 35)
Sets the default value for a job template entry of type `Boolean`.

`PMTemplateSetCFArrayConstraintValue` (page 36)
Sets the constraint values for a job template entry of type `CFArray`.

`PMTemplateSetCFDataDefaultValue` (page 36)
Sets the default value for a job template entry of type `CFDataRef`.

`PMTemplateSetCFDefaultValue` (page 37)
Sets the default value for a job template entry of type `CFTypeRef`.

`PMTemplateSetCFRangeConstraint` (page 37)
Sets a range of constraint values for a job template entry of type `CFTypeRef`.

`PMTemplateSetDoubleDefaultValue` (page 38)
Sets the default value for a job template entry of type `double`.

`PMTemplateSetDoubleListConstraint` (page 38)
Sets constraint values for a job template entry of type `double`.

`PMTemplateSetDoubleRangeDefaultValue` (page 40)
Sets the default range of values for a job template entry of type `double`.

`PMTemplateSetDoubleRangeConstraint` (page 39)
Sets a range of constraint values for a job template entry of type `double`.

`PMTemplateSetDoubleRangesConstraint` (page 40)
Sets a range of constraint values for a job template entry that is a range of type `double`.

`PMTemplateSetPMRectDefaultValue` (page 41)
Sets the default value for a job template entry of type `PMRect`.

PMTemplateSetPMRectListConstraint (page 42)
Sets constraint values for a job template entry of type PMRect.

PMTemplateSetPMTicketDefaultValue (page 43)
Sets the default value for a job template entry of type PMTicketRef.

PMTemplateSetPMTicketListConstraint (page 43)
Sets constraint values for a job template entry of type PMTicketRef.

PMTemplateSetSInt32DefaultValue (page 44)
Sets the default value for a job template entry of type SInt32.

PMTemplateSetSInt32ListConstraint (page 44)
Sets constraint values for a job template entry of type SInt32.

PMTemplateSetSInt32RangeDefaultValue (page 46)
Sets the default range of values for a job template entry of type SInt32.

PMTemplateSetSInt32RangeConstraint (page 45)
Sets a range of constraint values for a job template entry of type SInt32.

PMTemplateSetSInt32RangesConstraint (page 46)
Sets a range of constraint values for a job template entry that is a range of type SInt32.

Getting Template Items

PMTemplateGetValueType (page 31)
Obtains the data type of a job template entry.

PMTemplateGetConstraintType (page 20)
Obtains the constraint type for a job template entry.

PMTemplateGetBooleanDefaultValue (page 17)
Obtains the default value for a job template entry of type Boolean.

PMTemplateGetCFArrayConstraintValue (page 18)
Obtains an array of constraint values for a job template entry of type CFArrayRef.

PMTemplateGetCFDataDefaultValue (page 18)
Obtains the default value for a job template entry of type CFDataRef.

PMTemplateGetCFDefaultValue (page 19)
Obtains the default value for a job template entry of type CFTypeRef.

PMTemplateGetCFRangeConstraintValue (page 19)
Obtains the range of constraint values for a job template entry of type CFTypeRef.

PMTemplateGetDoubleDefaultValue (page 21)
Obtains the default value for a job template entry of type double.

PMTemplateGetDoubleRangeDefaultValue (page 23)
Obtains the default range values for a job template entry of type double.

PMTemplateGetDoubleListConstraintValue (page 21)
Obtains constraint values for a job template entry of type double.

PMTemplateGetDoubleRangeConstraintValue (page 22)
Obtains the default range of values for a job template entry of type double.

PMTemplateGetDoubleRangesConstraintValue (page 24)
Obtains a range for the minimum and maximum constraint values for a job template entry that is a range of type double.

PMTemplateGetPMRectDefaultValue (page 25)
Obtains the default value for a job template entry of type PMRect.

PMTemplateGetPMRectListConstraintValue (page 26)
Obtains constraint values for a job template entry of type PMRect.

PMTemplateGetPMTicketDefaultValue (page 26)
Obtains the default value for a job template entry of type PMTicket.

PMTemplateGetListTicketConstraintValue (page 24)
Obtains the constraint value for a job template entry that of type PMTicketRef.

PMTemplateGetSInt32DefaultValue (page 27)
Obtains the default value for a job template entry of type SInt32.

PMTemplateGetSInt32RangeDefaultValue (page 29)
Obtains the default range values for a job template entry of type SInt32.

PMTemplateGetSInt32ListConstraintValue (page 28)
Obtains constraint values for a job template entry of type SInt32.

PMTemplateGetSInt32RangeConstraintValue (page 28)
Obtains the default range of values for a job template entry of type SInt32.

PMTemplateGetSInt32RangesConstraintValue (page 30)
Obtains a range of constraint values for a job template entry that is a range of type SInt32.

Converting To and From XML

PMTemplateCreateXML (page 16)
Converts the data in a job template to data represented as XML.

PMTicketToXML (page 96)
Converts the data in a ticket to XML.

PMTicketWriteXML (page 97)
Converts a ticket to XML and then writes it to a file stream.

PMTicketWriteXMLToFile (page 98)
Converts a ticket to XML and then writes it to a file.

PMXMLToTicket (page 98)
Converts a ticket saved as XML to a ticket.

PMTemplateLoadFromXML (page 32)
Restores a job template that was previously converted to XML.

PMTicketReadXMLFromFile (page 73)
Restores a ticket previously converted to XML and saved as a file.

Deprecated Functions

PMTicketSetMetaItem (page 87)
Writes an item that does not need to be stored in an XML-representation of a ticket.

PMTicketGetMetaItem (page 65)
Obtains data for a item added with the function PMTicketSetMetaItem.

Functions

PMTemplateCreate

Creates a new job template.

```
OSStatus PMTemplateCreate (
   PMTemplateRef *newTemplate
);
```

Parameters

newTemplate

On return, points to a job template data structure allocated by the function.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTemplate.h`

PMTemplateCreateXML

Converts the data in a job template to data represented as XML.

```
OSStatus PMTemplateCreateXML (
   PMTemplateRef srcTemplate,
   CFDataRef *xmlData
);
```

Parameters

srcTemplate

A reference to a job template created by calling the function `PMTemplateCreate`.

xmlData

On return, points to a reference to an XML representation of the data in the job template specified by the `srcTemplate` parameter.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTemplate.h`

PMTemplateDelete

Deletes an existing job template.

```
OSStatus PMTemplateDelete (
   PMTemplateRef *oldTemplate
);
```

Parameters

oldTemplate

On input, a reference to a job template created by calling the function `PMTemplateCreate`. On return, points to `NULL`.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTemplate.h`

PMTemplateGetBooleanDefaultValue

Obtains the default value for a job template entry of type `Boolean`.

```
OSStatus PMTemplateGetBooleanDefaultValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   Boolean *defaultValue
);
```

Parameters

pmTemplate

A reference to a job template. You can call the function `PMSessionGetDataFromSession` with the ticket type set to `kPDE_PMJobTemplateRef` to get a value that you can then cast to a job template reference.

key

A reference to a `CFString` object that uniquely identifies the data you want to retrieve from the template.

defaultValue

On return, points to the default value for the template entry specified by the `key` parameter.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTemplate.h`

PMTemplateGetCFArrayConstraintValue

Obtains an array of constraint values for a job template entry of type `CFArrayRef`.

```
OSStatus PMTemplateGetCFArrayConstraintValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   CFArrayRef *constraintValue
);
```

Parameters

pmTemplate

A reference to a job template. You can call the function `PMSessionGetDataFromSession` with the ticket type set to `kPDE_PMJobTemplateRef` to get a value that you can then cast to a job template reference.

key

A reference to a `CFString` object that uniquely identifies the data you want to retrieve from the template.

constraintValue

On return, a poinrter to a Core Foundation array reference for the constraint values for the template entry specified by the `key` parameter.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTemplate.h`

PMTemplateGetCFDataDefaultValue

Obtains the default value for a job template entry of type `CFDataRef`.

```
OSStatus PMTemplateGetCFDataDefaultValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   CFDataRef *defaultValue
);
```

Parameters

pmTemplate

A reference to a job template. You can call the function `PMSessionGetDataFromSession` with the ticket type set to `kPDE_PMJobTemplateRef` to get a value that you can then cast to a job template reference.

key

A reference to a `CFString` object that uniquely identifies the data you want to retrieve from the template.

defaultValue

On return, points to a Core Foundation data reference that specfies the default data for the template entry specified by the `key` parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateGetCFDefaultValue

Obtains the default value for a job template entry of type `CFTypeRef`.

```
OSStatus PMTemplateGetCFDefaultValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   CFTypeRef *defaultValue
);
```

Parameters

pmTemplate
A reference to a job template. You can call the function `PMSessionGetDataFromSession` with the ticket type set to `kPDE_PMJobTemplateRef` to get a value that you can then cast to a job template reference.

key
A reference to a `CFString` object that uniquely identifies the data you want to retrieve from the template.

defaultValue
On return, points to a Core Foundation type reference that specifies the default data for the template entry specified by the `key` parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateGetCFRangeConstraintValue

Obtains the range of constraint values for a job template entry of type `CFTypeRef`.

```
OSStatus PMTemplateGetCFRangeConstraintValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   CFTypeRef *min,
   CFTypeRef *max
);
```

Parameters

pmTemplate

A reference to a job template. You can call the function `PMSessionGetDataFromSession` with the ticket type set to `kPDE_PMJobTemplateRef` to get a value that you can then cast to a job template reference.

key

A reference to a `CFString` object that uniquely identifies the data you want to retrieve from the template.

min

On return, points to a Core Foundation type reference that specifies the minimum value to which the template entry specified by the `key` parameter is constrained.

max

On return, points to a Core Foundation type reference that specifies the maximum value to which the template entry specified by the `key` parameter is constrained.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTemplate.h`

PMTemplateGetConstraintType

Obtains the constraint type for a job template entry.

```
OSStatus PMTemplateGetConstraintType (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   PMConstraintType *constraintType
);
```

Parameters

pmTemplate

A reference to a job template. You can call the function `PMSessionGetDataFromSession` with the ticket type set to `kPDE_PMJobTemplateRef` to get a value that you can then cast to a job template reference.

key

A reference to a `CFString` object that uniquely identifies the data whose constraint type you want to retrieve from the template.

constraintType

On return, points to the constraint type for the data specified by the `key` parameter. See "Constraint Types" (page 104) for a list of the constraint types that can be returned.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateGetDoubleDefaultValue

Obtains the default value for a job template entry of type `double`.

```
OSStatus PMTemplateGetDoubleDefaultValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   double *defaultValue
);
```

Parameters

pmTemplate

A reference to a job template. You can call the function `PMSessionGetDataFromSession` with the ticket type set to `kPDE_PMJobTemplateRef` to get a value that you can then cast to a job template reference.

key

A reference to a `CFString` object that uniquely identifies the data you want to retrieve from the template.

defaultValue

On return, points to the default `double` value for the template entry specified by the `key` parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateGetDoubleListConstraintValue

Obtains constraint values for a job template entry of type `double`.

```
OSStatus PMTemplateGetDoubleListConstraintValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   int *listSize,
   double *doubleList
);
```

Parameters

pmTemplate

A reference to a job template. You can call the function `PMSessionGetDataFromSession` with the ticket type set to `kPDE_PMJobTemplateRef` to get a value that you can then cast to a job template reference.

key

A reference to a `CFString` object that uniquely identifies the data you want to retrieve from the template.

listSize

A pointer to the number of items in the array specified by the parameter `doubleList`.

doubleList

On return, points to an array of `double` values to which the template entry specified by the `key` parameter is constrained.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Discussion

You need to call the function `PMTemplateGetDoubleListConstraintValue` twice. First, call the function to get the number of items in the array specified by the parameter `doubleList`. You should call the function again after you allocate space for the array.

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTemplate.h`

PMTemplateGetDoubleRangeConstraintValue

Obtains the default range of values for a job template entry of type `double`.

```
OSStatus PMTemplateGetDoubleRangeConstraintValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   double *min,
   double *max
);
```

Parameters

pmTemplate

A reference to a job template. You can call the function `PMSessionGetDataFromSession` with the ticket type set to `kPDE_PMJobTemplateRef` to get a value that you can then cast to a job template reference.

key

A reference to a `CFString` object that uniquely identifies the data you want to retrieve from the template.

min

On return, points to the minimum value to which the template entry specified by the `key` parameter is constrained.

max

On return, points to the maximum value to which the template entry specified by the `key` parameter is constrained.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateGetDoubleRangeDefaultValue

Obtains the default range values for a job template entry of type `double`.

```
OSStatus PMTemplateGetDoubleRangeDefaultValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   double *min,
   double *max
);
```

Parameters

pmTemplate

A reference to a job template. You can call the function `PMSessionGetDataFromSession` with the ticket type set to `kPDE_PMJobTemplateRef` to get a value that you can then cast to a job template reference.

key

A reference to a `CFString` object that uniquely identifies the data you want to retrieve from the template.

min

On return, points to the minimum value to which the range is constrained for the template entry specified by the `key` parameter.

max

On return, points to the maximum value to which the range is constrained for the template entry specified by the `key` parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
PMTemplate.h

PMTemplateGetDoubleRangesConstraintValue

Obtains a range for the minimum and maximum constraint values for a job template entry that is a range of type double.

```
OSStatus PMTemplateGetDoubleRangesConstraintValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   double *minForMin,
   double *maxForMin,
   double *minForMax,
   double *maxForMax
);
```

Parameters

pmTemplate

A reference to a job template. You can call the function PMSessionGetDataFromSession with the ticket type set to kPDE_PMJobTemplateRef to get a value that you can then cast to a job template reference.

key

A reference to a CFString object that uniquely identifies the data you want to retrieve from the template.

minForMin

On return, points to the minimum value to which the minimum of the range is constrained for the template entry specified by the key parameter.

maxForMin

On return, points to the maximum value to which the minimum of the range is constrained for the template entry specified by the key parameter.

minForMax

On return, points to the minimum value to which the maximum of the range is constrained for the template entry specified by the key parameter.

maxForMax

On return, points to the maximum value to which the maximum of the range is constrained for the template entry specified by the key parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
PMTemplate.h

PMTemplateGetListTicketConstraintValue

Obtains the constraint value for a job template entry that of type PMTicketRef.

```
OSStatus PMTemplateGetListTicketConstraintValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   PMTicketRef *listTicket
);
```

Parameters

pmTemplate

A reference to a job template. You can call the function `PMSessionGetDataFromSession` with the ticket type set to `kPDE_PMJobTemplateRef` to get a value that you can then cast to a job template reference.

key

A reference to a `CFString` object that uniquely identifies the data you want to retrieve from the template.

listTicket

On return, points to a reference to the list ticket to which the template entry specified by the `key` parameter is constrained.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateGetPMRectDefaultValue

Obtains the default value for a job template entry of type `PMRect`.

```
OSStatus PMTemplateGetPMRectDefaultValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   PMRect *defaultValue
);
```

Parameters

pmTemplate

A reference to a job template. You can call the function `PMSessionGetDataFromSession` with the ticket type set to `kPDE_PMJobTemplateRef` to get a value that you can then cast to a job template reference.

key

A reference to a `CFString` object that uniquely identifies the data you want to retrieve from the template.

defaultValue

On return, points to the default `PMRect` data for the template entry specified by the `key` parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateGetPMRectListConstraintValue

Obtains constraint values for a job template entry of type `PMRect`.

```
OSStatus PMTemplateGetPMRectListConstraintValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   int *listSize,
   PMRect *rectList
);
```

Parameters

pmTemplate

A reference to a job template. You can call the function `PMSessionGetDataFromSession` with the ticket type set to `kPDE_PMJobTemplateRef` to get a value that you can then cast to a job template reference.

key

A reference to a `CFString` object that uniquely identifies the data you want to retrieve from the template.

listSize

A pointer to the number of items in the array specified by the parameter `rectList`.

rectList

On return, points to an array of `PMRect` values to which the template entry specified by the `key` parameter is constrained.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Discussion
You need to call the function `PMTemplateGetPMRectListConstraintValue` twice. First, call the function to get the number of items in the array specified by the parameter `rectList`. You should call the function again after you allocate space for the array.

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateGetPMTicketDefaultValue

Obtains the default value for a job template entry of type `PMTicket`.

```
OSStatus PMTemplateGetPMTicketDefaultValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   PMTicketRef *defaultValue
);
```

Parameters

pmTemplate

A reference to a job template. You can call the function `PMSessionGetDataFromSession` with the ticket type set to `kPDE_PMJobTemplateRef` to get a value that you can then cast to a job template reference.

key

A reference to a `CFString` object that uniquely identifies the data you want to retrieve from the template.

defaultValue

On return, points to a reference to the default ticket for the template entry specified by the `key` parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateGetSInt32DefaultValue

Obtains the default value for a job template entry of type `SInt32`.

```
OSStatus PMTemplateGetSInt32DefaultValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   SInt32 *defaultValue
);
```

Parameters

pmTemplate

A reference to a job template. You can call the function `PMSessionGetDataFromSession` with the ticket type set to `kPDE_PMJobTemplateRef` to get a value that you can then cast to a job template reference.

key

A reference to a `CFString` object that uniquely identifies the data you want to retrieve from the template.

defaultValue

On return, points to the default value for the template entry specified by the `key` parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateGetSInt32ListConstraintValue

Obtains constraint values for a job template entry of type `SInt32`.

```
OSStatus PMTemplateGetSInt32ListConstraintValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   int *listSize,
   SInt32 *sint32List
);
```

Parameters

pmTemplate

A reference to a job template. You can call the function `PMSessionGetDataFromSession` with the ticket type set to `kPDE_PMJobTemplateRef` to get a value that you can then cast to a job template reference.

key

A reference to a `CFString` object that uniquely identifies the data you want to retrieve from the template.

listSize

On return, points to the number of items in the array specified by the parameter `sint32List`.

sint32List

On return, points to an array of `SInt32` values to which the template entry specified by the `key` parameter is constrained.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Discussion
You need to call the function `PMTemplateGetSInt32ListConstraintValue` twice. First, call the function to get the number of items in the array specified by the parameter `sint32List`. You should call the function again after you allocate space for the array.

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateGetSInt32RangeConstraintValue

Obtains the default range of values for a job template entry of type `SInt32`.

```
OSStatus PMTemplateGetSInt32RangeConstraintValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   SInt32 *min,
   SInt32 *max
);
```

Parameters

pmTemplate

A reference to a job template. You can call the function `PMSessionGetDataFromSession` with the ticket type set to `kPDE_PMJobTemplateRef` to get a value that you can then cast to a job template reference.

key

A reference to a `CFString` object that uniquely identifies the data you want to retrieve from the template.

min

On return, points to the minimum value to which the template entry specified by the `key` parameter is constrained.

max

On return, points to the maximum value to which the template entry specified by the `key` parameter is constrained.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateGetSInt32RangeDefaultValue

Obtains the default range values for a job template entry of type `SInt32`.

```
OSStatus PMTemplateGetSInt32RangeDefaultValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   SInt32 *min,
   SInt32 *max
);
```

Parameters

pmTemplate

A reference to a job template. You can call the function `PMSessionGetDataFromSession` with the ticket type set to `kPDE_PMJobTemplateRef` to get a value that you can then cast to a job template reference.

key

A reference to a `CFString` object that uniquely identifies the data you want to retrieve from the template.

min

On return, points to the minimum value to which the range is constrained for the template entry specified by the `key` parameter.

max

On return, points to the maximum value to which the range is constrained for the template entry specified by the `key` parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateGetSInt32RangesConstraintValue

Obtains a range of constraint values for a job template entry that is a range of type `SInt32`.

```
OSStatus PMTemplateGetSInt32RangesConstraintValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   SInt32 *minForMin,
   SInt32 *maxForMin,
   SInt32 *minForMax,
   SInt32 *maxForMax
);
```

Parameters

pmTemplate

A reference to a job template. You can call the function `PMSessionGetDataFromSession` with the ticket type set to `kPDE_PMJobTemplateRef` to get a value that you can then cast to a job template reference.

key

A reference to a `CFString` object that uniquely identifies the data you want to retrieve from the template.

minForMin

On return, points to the minimum value to which the minimum of the range is constrained for the template entry specified by the `key` parameter.

maxForMin

On return, points to the maximum value to which the minimum of the range is constrained for the template entry specified by the `key` parameter.

minForMax

On return, points to the minimum value to which the maximum of the range is constrained for the template entry specified by the `key` parameter.

maxForMax

On return, points to the maximum value to which the maximum of the range is constrained for the template entry specified by the `key` parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateGetValueType

Obtains the data type of a job template entry.

```
OSStatus PMTemplateGetValueType (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   PMValueType *valueType
);
```

Parameters

pmTemplate

A reference to a job template. You can call the function `PMSessionGetDataFromSession` with the ticket type set to `kPDE_PMJobTemplateRef` to get a value that you can then cast to a job template reference.

key

A reference to a `CFString` object that uniquely identifies the data you want to retrieve from the template.

valueType

On return, points to the value type for the template entry specified by the `key` parameter. See "PMTicketItemType" (page 102) for a list of values that can be returned.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateIsLocked

Checks to see if a job template is locked.

```
OSStatus PMTemplateIsLocked (
   PMTemplateRef srcTemplate,
   Boolean *locked
);
```

Parameters

srcTemplate

A reference to a job template created by calling the function `PMTemplateCreate`.

locked

On return, points to `true` if the job template is locked and `false` if the template is not locked.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Discussion
A job template cannot be modified unless it is unlocked.

Availability
Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateLoadFromXML

Restores a job template that was previously converted to XML.

```
OSStatus PMTemplateLoadFromXML (
   CFDataRef srcData,
   PMTemplateRef *destTemplate
);
```

Parameters

srcData

A reference to `CFData` that represents job template data. You obtain a `CFDataRef` by calling the function `PMTemplateCreateXML`.

destTemplate

A reference to a job template created by calling the function `PMTemplateCreate`. On return, the job template contains the entries specified by the XML data.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateMakeEntry

Creates a new entry in a job template.

```
OSStatus PMTemplateMakeEntry (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   PMValueType valueType,
   PMConstraintType constraintType
);
```

Parameters

pmTemplate

A reference to a job template created by calling the function `PMTemplateCreate`.

key

A reference to a `CFString` object that uniquely identifies the entry. You use the key to set and obtain the data for this entry.

valueType

The data type of the value associated with the entry. See "PMValueType" (page 103) for a list of constants you can use to specify the data type.

constraintType

The type of constraint you want applied to the data associated with the entry. See "Constraint Types" (page 104) for a list of constants you can use to specify the constraint type.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Discussion

After you call the function `PMTemplateMakeEntry`, you should set default and constraint values for the entry by calling the appropriate `PMTemplateSet` functions for the data type of the entry.

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTemplate.h`

PMTemplateMakeFullEntry

Creates a new entry in a job template and sets the default and constraint values for the entry.

```
OSStatus PMTemplateMakeFullEntry (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   PMValueType valueType,
   PMConstraintType constraintType,
   CFTypeRef defaultValue,
   CFTypeRef constraintValue
);
```

Parameters

pmTemplate

A reference to a job template created by calling the function `PMTemplateCreate`.

key

A reference to a `CFString` object that uniquely identifies the entry. You use the key to set and obtain the data for this entry.

valueType

The data type of the value associated with the entry. See "PMValueType" (page 103) for a list of constants you can use to specify the data type.

constraintType

The type of constraint you want applied to the data associated with the entry. See "Constraint Types" (page 104) for a list of constants you can use to specify the constraint type.

defaultValue

A reference to the default value for this entry.

constraintValue

A reference to the values used to constrain the data associated with this entry.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateMergeTemplates

Merges the entries from one job template with entries from another job template.

```
OSStatus PMTemplateMergeTemplates (
   PMTemplateRef sourceTemplate,
   PMTemplateRef destTemplate
);
```

Parameters

sourceTemplate

A reference to the job template whose entries you want to merge with the destination template.

destTemplate

A reference to the job template into which you want to merge entries from the source template. Any entry in the destination template that has the same key as an entry in the source template is overwritten by the data from the source template.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateRemoveEntry

Removes an entry from a job template.

```
OSStatus PMTemplateRemoveEntry (
   PMTemplateRef pmTemplate,
   CFStringRef key
);
```

Parameters

pmTemplate

A reference to a job template created by calling the function `PMTemplateCreate`.

key

A reference to a `CFString` object that uniquely identifies the entry you want to remove.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTemplate.h`

PMTemplateSetBooleanDefaultValue

Sets the default value for a job template entry of type `Boolean`.

```
OSStatus PMTemplateSetBooleanDefaultValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   Boolean defaultValue
);
```

Parameters

pmTemplate

A reference to a job template created by calling the function `PMTemplateCreate`.

key

A reference to a `CFString` object that uniquely identifies the data whose default value you want to set.

defaultValue

A `Boolean` value that specifies the default value for template entry specified by the `key` parameter.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTemplate.h`

PMTemplateSetCFArrayConstraintValue

Sets the constraint values for a job template entry of type `CFArray`.

```
OSStatus PMTemplateSetCFArrayConstraintValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   CFArrayRef constraintValue
);
```

Parameters

pmTemplate

A reference to a job template created by calling the function `PMTemplateCreate`.

key

A reference to a `CFString` object that uniquely identifies the data whose constraint values you want to set.

constraintValue

A reference to a Core Foundation array that contains the values to which the template entry specified by the `key` parameter is constrained.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTemplate.h`

PMTemplateSetCFDataDefaultValue

Sets the default value for a job template entry of type `CFDataRef`.

```
OSStatus PMTemplateSetCFDataDefaultValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   CFDataRef defaultValue
);
```

Parameters

pmTemplate

A reference to a job template created by calling the function `PMTemplateCreate`.

key

A reference to a `CFString` object that uniquely identifies the data whose default value you want to set.

defaultValue

A reference to the default data for template entry specified by the `key` parameter.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In
PMTemplate.h

PMTemplateSetCFDefaultValue

Sets the default value for a job template entry of type CFTypeRef.

```
OSStatus PMTemplateSetCFDefaultValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   CFTypeRef defaultValue
);
```

Parameters

pmTemplate
A reference to a job template created by calling the function PMTemplateCreate.

key
A reference to a CFString object that uniquely identifies the data whose default value you want to set.

defaultValue
A reference to the default data for template entry specified by the key parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
PMTemplate.h

PMTemplateSetCFRangeConstraint

Sets a range of constraint values for a job template entry of type CFTypeRef.

```
OSStatus PMTemplateSetCFRangeConstraint (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   CFTypeRef min,
   CFTypeRef max
);
```

Parameters

pmTemplate
A reference to a job template created by calling the function PMTemplateCreate.

key
A reference to a CFString object that uniquely identifies the data whose range of constraint values you want to set.

min

A reference to the minimum value to which the template entry specified by the `key` parameter is constrained.

max

A reference to the maximum value to which the template entry specified by the `key` parameter is constrained.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateSetDoubleDefaultValue

Sets the default value for a job template entry of type `double`.

```
OSStatus PMTemplateSetDoubleDefaultValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   double defaultValue
);
```

Parameters

pmTemplate

A reference to a job template created by calling the function `PMTemplateCreate`.

key

A reference to a `CFString` object that uniquely identifies the data whose default value you want to set.

defaultValue

A `double` value that specifies the default value for template entry specified by the `key` parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateSetDoubleListConstraint

Sets constraint values for a job template entry of type `double`.

```
OSStatus PMTemplateSetDoubleListConstraint (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   int listSize,
   double *doubleList
);
```

Parameters

pmTemplate

A reference to a job template created by calling the function `PMTemplateCreate`.

key

A reference to a `CFString` object that uniquely identifies the data whose constraint values you want to set.

listSize

The size of the array specified by the parameter `doubleList`.

doubleList

A pointer to an array of `double` values to which the template entry specified by the `key` parameter is constrained.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTemplate.h`

PMTemplateSetDoubleRangeConstraint

Sets a range of constraint values for a job template entry of type `double`.

```
OSStatus PMTemplateSetDoubleRangeConstraint (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   double min,
   double max
);
```

Parameters

pmTemplate

A reference to a job template created by calling the function `PMTemplateCreate`.

key

A reference to a `CFString` object that uniquely identifies the data whose constraint values you want to set.

min

A `double` value that specifies the minimum value to which the template entry specified by the `key` parameter is constrained.

max

A `double` value that specifies the maximum value to which the template entry specified by the `key` parameter is constrained.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateSetDoubleRangeDefaultValue

Sets the default range of values for a job template entry of type `double`.

```
OSStatus PMTemplateSetDoubleRangeDefaultValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   double min,
   double max
);
```

Parameters

pmTemplate

A reference to a job template created by calling the function `PMTemplateCreate`.

key

A reference to a `CFString` object that uniquely identifies the data whose default values you want to set.

min

A `double` value that specifies the default minimum value for the template entry specified by the `key` parameter.

max

A `double` value that specifies the default maximum value for template entry specified by the `key` parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateSetDoubleRangesConstraint

Sets a range of constraint values for a job template entry that is a range of type `double`.

```
OSStatus PMTemplateSetDoubleRangesConstraint (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   double minForMin,
   double maxForMin,
   double minForMax,
   double maxForMax
);
```

Parameters

pmTemplate

A reference to a job template created by calling the function `PMTemplateCreate`.

key

A reference to a `CFString` object that uniquely identifies the data whose range of constraint values you want to set.

minForMin

A `double` value that specifies the minimum value to which the minimum of the range is constrained for the template entry specified by the `key` parameter.

maxForMin

A `double` value that specifies the maximum value to which the minimum of the range is constrained for the template entry specified by the `key` parameter.

minForMax

A `double` value that specifies the minimum value to which the maximum of the range is constrained for the template entry specified by the `key` parameter.

maxForMax

A `double` value that specifies the maximum value to which the maximum of the range is constrained for the template entry specified by the `key` parameter.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTemplate.h`

PMTemplateSetPMRectDefaultValue

Sets the default value for a job template entry of type `PMRect`.

```
OSStatus PMTemplateSetPMRectDefaultValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   PMRect *defaultValue
);
```

Parameters

pmTemplate

A reference to a job template created by calling the function `PMTemplateCreate`.

key

A reference to a `CFString` object that uniquely identifies the data whose default value you want to set.

defaultValue

A pointer to a `PMRect` that specifies the default rectangle for the template entry specified by the `key` parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateSetPMRectListConstraint

Sets constraint values for a job template entry of type `PMRect`.

```
OSStatus PMTemplateSetPMRectListConstraint (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   int listSize,
   PMRect *rectList
);
```

Parameters

pmTemplate

A reference to a job template created by calling the function `PMTemplateCreate`.

key

A reference to a `CFString` object that uniquely identifies the data whose constraint values you want to set.

listSize

The size of the array specified by the parameter `rectList`.

rectList

A pointer to an array of `PMRect` values to which the template entry specified by the `key` parameter is constrained.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateSetPMTicketDefaultValue

Sets the default value for a job template entry of type `PMTicketRef`.

```
OSStatus PMTemplateSetPMTicketDefaultValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   PMTicketRef defaultValue
);
```

Parameters

pmTemplate

A reference to a job template created by calling the function `PMTemplateCreate`.

key

A reference to a `CFString` object that uniquely identifies the data whose default value you want to set.

defaultValue

A reference to a ticket that specifies the default ticket for the template entry specified by the `key` parameter.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTemplate.h`

PMTemplateSetPMTicketListConstraint

Sets constraint values for a job template entry of type `PMTicketRef`.

```
OSStatus PMTemplateSetPMTicketListConstraint (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   PMTicketRef listTicket
);
```

Parameters

pmTemplate

A reference to a job template created by calling the function `PMTemplateCreate`.

key

A reference to a `CFString` object that uniquely identifies the data whose constraint values you want to set.

listTicket

A reference to a ticket that specifies the default list ticket for the template entry specified by the `key` parameter.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateSetSInt32DefaultValue

Sets the default value for a job template entry of type `SInt32`.

```
OSStatus PMTemplateSetSInt32DefaultValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   SInt32 defaultValue
);
```

Parameters

`pmTemplate`
A reference to a job template created by calling the function `PMTemplateCreate`.

`key`
A reference to a `CFString` object that uniquely identifies the data whose default value you want to set.

`defaultValue`
An `SInt32` value that specifies the default value for the template entry specified by the `key` parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateSetSInt32ListConstraint

Sets constraint values for a job template entry of type `SInt32`.

```
OSStatus PMTemplateSetSInt32ListConstraint (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   int listSize,
   SInt32 *sint32List
);
```

Parameters

`pmTemplate`
A reference to a job template created by calling the function `PMTemplateCreate`.

key

A reference to a `CFString` object that uniquely identifies the data whose constraint values you want to set.

listSize

The size of the array specified by the parameter `sint32List`.

sint32List

A pointer to an array of `SInt32` values to which the template entry specified by the `key` parameter is constrained.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateSetSInt32RangeConstraint

Sets a range of constraint values for a job template entry of type `SInt32`.

```
OSStatus PMTemplateSetSInt32RangeConstraint (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   SInt32 min,
   SInt32 max
);
```

Parameters

pmTemplate

A reference to a job template created by calling the function `PMTemplateCreate`.

key

A reference to a `CFString` object that uniquely identifies the data whose constraint values you want to set.

min

An `SInt32` value that specifies the minimum value to which the template entry specified by the `key` parameter is constrained.

max

An `SInt32` value that specifies the maximum value to which the template entry specified by the `key` parameter is constrained.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTemplate.h`

PMTemplateSetSInt32RangeDefaultValue

Sets the default range of values for a job template entry of type `SInt32`.

```
OSStatus PMTemplateSetSInt32RangeDefaultValue (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   SInt32 min,
   SInt32 max
);
```

Parameters

pmTemplate

A reference to a job template created by calling the function `PMTemplateCreate`.

key

A reference to a `CFString` object that uniquely identifies the data whose default values you want to set.

min

An `SInt32` value that specifies the default minimum value for the template entry specified by the `key` parameter.

max

An `SInt32` value that specifies the default maximum value for the template entry specified by the `key` parameter.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTemplate.h`

PMTemplateSetSInt32RangesConstraint

Sets a range of constraint values for a job template entry that is a range of type `SInt32`.

```
OSStatus PMTemplateSetSInt32RangesConstraint (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   SInt32 minForMin,
   SInt32 maxForMin,
   SInt32 minForMax,
   SInt32 maxForMax
);
```

Parameters

pmTemplate

A reference to a job template created by calling the function `PMTemplateCreate`.

key

A reference to a `CFString` object that uniquely identifies the data whose range of constraint values you want to set.

minForMin

An SInt32 value that specifies the minimum value to which the minimum of the range is constrained for the template entry specified by the key parameter.

maxForMin

An SInt32 value that specifies the maximum value to which the minimum of the range is constrained for the template entry specified by the key parameter.

minForMax

An SInt32 value that specifies the minimum value to which the maximum of the range is constrained for the template entry specified by the key parameter.

maxForMax

An SInt32 value that specifies the maximum value to which the maximum of the range is constrained for the template entry specified by the key parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
PMTemplate.h

PMTemplateValidateItem

Validates an item in a job template.

```
OSStatus PMTemplateValidateItem (
   PMTemplateRef pmTemplate,
   CFStringRef key,
   CFTypeRef item,
   Boolean *validationResults
);
```

Parameters

pmTemplate

A reference to a job template created by calling the function PMTemplateCreate.

key

A reference to a CFString object that uniquely identifies the job template entry you want to validate.

item

A reference to the data you want to validate.

validationResults

On return, points to true is the item is validated and false if it is not validated.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
PMTemplate.h

PMTicketConfirmTicket

Checks whether a ticket appears to be valid.

```
OSStatus PMTicketConfirmTicket (
   PMTicketRef ticket
);
```

Parameters

ticket
A reference to a ticket created by calling the function PMTicketCreate.

Return Value
A result code. Returns noErr if the ticket appears to be valid. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
PMTicket.h

PMTicketContainsItem

Checks whether an item exists in a ticket.

```
Boolean PMTicketContainsItem (
   PMTicketRef ticket,
   UInt32 nodeIndex1,
   UInt32 nodeIndex2,
   CFStringRef key
);
```

Parameters

ticket
A reference to a ticket created by calling the function PMTicketCreate.

nodeIndex1
Reserved for future use. Currently, you must pass the constant kPMTopLevel.

nodeIndex2
Reserved for future use. Currently, you must pass the constant kPMTopLevel.

key
A reference to a CFString object that uniquely identifies the ticket item you want to check.

Return Value
Returns true if the item exists.

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
PMTicket.h

PMTicketContainsTicket

Checks whether a ticket is contained in another ticket.

```
OSStatus PMTicketContainsTicket (
   PMTicketRef ticket,
   CFStringRef requestedType,
   UInt32 index,
   Boolean *exists
);
```

Parameters

ticket

A reference to a ticket created by calling the function PMTicketCreate.

requestedType

A reference to a CFString object that specfies the ticket type of the ticket for which you want to check. See "Ticket Type Strings" (page 133) for a list of constants you can pass.

index

Reserved for future use. Currently, you must pass the constant kPMTopLevel.

exists

On return, points to true if the ticket contains a subticket of the ticket type specified by the requestedType parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
PMTicket.h

PMTicketCopy

Copies a ticket.

```
OSStatus PMTicketCopy (
   CFAllocatorRef allocator,
   PMTicketRef sourceTicket,
   PMTicketRef *destinationTicket
);
```

Parameters

allocator

A reference to the allocator object to be used for allocating memory. Pass a reference to a valid allocator or kCFAllocatorDefault to request the default allocator.

sourceTicket

A reference to a ticket created by calling the function `PMTicketCreate`. This is the ticket you want to copy.

destinationTicket

A pointer to a ticket reference. On return, points to a copy of the source ticket.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketCopyItem

Copies an item from one ticket to another ticket.

```
OSStatus PMTicketCopyItem (
   PMTicketRef sourceTicket,
   PMTicketRef destTicket,
   CFStringRef clientID,
   CFStringRef key,
   Boolean locked
);
```

Parameters

sourceTicket

A reference to a ticket created by calling the function `PMTicketCreate`. This is the ticket from which you want to copy an item.

destTicket

A reference to a ticket created by calling the function `PMTicketCreate`. This is the ticket to which the item is copied.

clientID

A reference to a `CFString` object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as `CFSTR("com.myvendorname.myprintingcode")`.

key

A reference to a `CFString` object that uniquely identifies the ticket item you want to copy.

locked

A `Boolean` value. Pass `true` to lock the copied item or `false` to set the item's lock state to unlock. Locking the item prevents any subsequent modification of this item.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Discussion
If an item must be copied from one ticket to another, call the function `PMTicketCopyItem` to make the simple transfer. This updates the modification date and client ID for the item. The *locked* field determines if subsequent updates can be made.

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketCreate

Creates a new ticket.

```
OSStatus PMTicketCreate (
   CFAllocatorRef allocator,
   CFStringRef ticketType,
   PMTicketRef *newTicket
);
```

Parameters

allocator

A reference to the allocator object to be used for allocating memory. Pass a reference to a valid allocator or `kCFAllocatorDefault` to request the default allocator.

ticketType

A reference to a `CFString` object that specfies the ticket type of the ticket you want to create. See "Ticket Type Strings" (page 133) for a list of constants you can pass.

newTicket

On return, a reference to a ticket.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketCreateTemplate

Retrieves a template from a ticket.

```
OSStatus PMTicketCreateTemplate (
   PMTicketRef ticket,
   UInt32 nodeIndex1,
   UInt32 nodeIndex2,
   CFStringRef key,
   PMTemplateRef *item
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

nodeIndex1

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

nodeIndex2

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

key

A reference to a `CFString` object that uniquely identifies the template you want to retrieve.

item

On return, points to the template reference specified by the `key` parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Discussion
The function `PMTicketCreateTemplate` retrieves the template data associated with the specified key. The template is stored in the ticket as flattened data. The function creates a template object from the flattened data.

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketDeleteItem

Makes an item in a ticket unavailable.

```
OSStatus PMTicketDeleteItem (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

clientID

A reference to a `CFString` object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as `CFSTR("com.myvendorname.myprintingcode")`.

key

A reference to a `CFString` object that uniquely identifies the ticket item you want to make unavailable.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Discussion
After an item is deleted by calling the function `PMTicketDeleteItem`, the result code `kPMInvalidItem` is returned when anyone tries to access the item. The function `PMTicketDeleteItem` actually makes the item unavailable rather than deleting the item. The function adds information to the item's dictionary to record the history of the item and to indicate the item is unavailable. You should call the function `PMTicketReleaseItem` if you want to completely remove the item from the ticket.

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketFillFromArray

Adds items defined in an array of ticket item structures to a ticket.

```
OSStatus PMTicketFillFromArray (
   PMTicketRef ticket,
   CFStringRef clientID,
   const PMTicketItemStruct *items,
   UInt32 itemCount
);
```

Parameters
ticket
A reference to a ticket created by calling the function `PMTicketCreate`.
clientID
A reference to a `CFString` object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as `CFSTR("com.myvendorname.myprintingcode")`.
items
A pointer to an array of ticket item structures.
itemCount
The number of items in the `items` array.

Return Value
A result code. If the result code is anything other than `noErr`, it's possible that not all of the items were added successfully. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketGetAllocator

Obtains the allocator object used for allocating memory for a ticket.

```
OSStatus PMTicketGetAllocator (
   PMTicketRef ticket,
   CFAllocatorRef *allocator
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

allocator

On return, a reference to the `CFAllocator` object associated with the ticket.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTicket.h`

PMTicketGetAPIVersion

Obtains the version of the application programming interface (API) used to created a ticket.

```
OSStatus PMTicketGetAPIVersion (
   PMTicketRef ticket,
   CFStringRef *apiVersion
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

apiVersion

On return, a reference to a `CFString` object that specifies the version of the API used to create the ticket.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTicket.h`

PMTicketGetBoolean

Obtains data for a ticket item of type `Boolean`.

```
OSStatus PMTicketGetBoolean (
   PMTicketRef ticket,
   UInt32 nodeIndex1,
   UInt32 nodeIndex2,
   CFStringRef key,
   Boolean *value
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

nodeIndex1

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

nodeIndex2

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

key

A reference to a `CFString` object that uniquely identifies the ticket item whose value you want to obtain. See the Base Services documentation for a description of the `CFStringRef` data type.

value

On return, points to the `Boolean` value for the item specified by the `key` parameter.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTicket.h`

PMTicketGetBytes

Obtains data for a ticket item that's an array of `UInt8` values.

```
OSStatus PMTicketGetBytes (
   PMTicketRef ticket,
   UInt32 nodeIndex1,
   UInt32 nodeIndex2,
   CFStringRef key,
   UInt8 *data,
   UInt32 *size
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

nodeIndex1

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

nodeIndex2

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

key

A reference to a `CFString` object that uniquely identifies the ticket item whose value you want to obtain.

data

On return, points to an array of data of type `UInt8` for the item specified by the `key` parameter.

size

On input, pass the size of the buffer pointed to by the `data` parameter. On return, points to the number of bytes in the array.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Discussion
If you don't know the size of the data buffer, you need to call the function `PMTicketGetBytes` twice. First, call the function to get the number of items in the array specified by the parameter `data`. You should call the function again after you allocate space for the array.

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketGetCFArray

Obtains data for a ticket item that's a Core Foundation array.

```
OSStatus PMTicketGetCFArray (
   PMTicketRef ticket,
   UInt32 nodeIndex1,
   UInt32 nodeIndex2,
   CFStringRef key,
   CFArrayRef *item
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

nodeIndex1

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

nodeIndex2

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

key

A reference to a `CFString` object that uniquely identifies the ticket item whose value you want to obtain.

item

On return, points to a Core Foundation array reference for the item specified by the `key` parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
PMTicket.h

PMTicketGetCFBoolean

Obtains data for a ticket item that's an array of `CFBoolean` values.

```
OSStatus PMTicketGetCFBoolean (
   PMTicketRef ticket,
   UInt32 nodeIndex1,
   UInt32 nodeIndex2,
   CFStringRef key,
   CFBooleanRef *item
);
```

Parameters

ticket
A reference to a ticket created by calling the function `PMTicketCreate`.

nodeIndex1
Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

nodeIndex2
Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

key
A reference to a `CFString` object that uniquely identifies the ticket item whose value you want to obtain.

item
On return, points to a Core Foundation Boolean reference for the item specified by the `key` parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
PMTicket.h

PMTicketGetCFData

Obtains data for a ticket item of type `CFData`.

```
OSStatus PMTicketGetCFData (
   PMTicketRef ticket,
   UInt32 nodeIndex1,
   UInt32 nodeIndex2,
   CFStringRef key,
   CFDataRef *item
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

nodeIndex1

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

nodeIndex2

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

key

A reference to a `CFString` object that uniquely identifies the ticket item whose value you want to obtain.

item

On return, points to a Core Foundation data reference for the item specified by the `key` parameter.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTicket.h`

PMTicketGetCFDate

Obtains data for a ticket item of type `CFDate`.

```
OSStatus PMTicketGetCFDate (
   PMTicketRef ticket,
   UInt32 nodeIndex1,
   UInt32 nodeIndex2,
   CFStringRef key,
   CFDateRef *item
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

nodeIndex1

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

nodeIndex2

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

key

A reference to a `CFString` object that uniquely identifies the ticket item whose value you want to obtain.

item

On return, points to a Core Foundation date reference for the item specified by the `key` parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketGetCFDictionary

Obtains data for a ticket item of type `CFDictionary`.

```
OSStatus PMTicketGetCFDictionary (
   PMTicketRef ticket,
   UInt32 nodeIndex1,
   UInt32 nodeIndex2,
   CFStringRef key,
   CFDictionaryRef *item
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

nodeIndex1

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

nodeIndex2

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

key

A reference to a `CFString` object that uniquely identifies the ticket item whose value you want to obtain.

item

On return, points to a Core Foundation dictionary reference for the item specified by the `key` parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketGetCFNumber

Obtains data for a ticket item of type `CFNumber`.

```
OSStatus PMTicketGetCFNumber (
   PMTicketRef ticket,
   UInt32 nodeIndex1,
   UInt32 nodeIndex2,
   CFStringRef key,
   CFNumberRef *item
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

nodeIndex1

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

nodeIndex2

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

key

A reference to a `CFString` object that uniquely identifies the ticket item whose value you want to obtain.

item

On return, points to a Core Foundation number reference for the item specified by the `key` parameter.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTicket.h`

PMTicketGetCFString

Obtains data for a ticket item of type `CFString`.

```
OSStatus PMTicketGetCFString (
   PMTicketRef ticket,
   UInt32 nodeIndex1,
   UInt32 nodeIndex2,
   CFStringRef key,
   CFStringRef *item
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

nodeIndex1

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

nodeIndex2

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

key

A reference to a `CFString` object that uniquely identifies the ticket item whose value you want to obtain.

item

On return, points to a Core Foundation string reference for the item specified by the `key` parameter.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTicket.h`

PMTicketGetCString

Obtains data for a ticket item of that's a C-style string.

```
OSStatus PMTicketGetCString (
   PMTicketRef ticket,
   UInt32 nodeIndex1,
   UInt32 nodeIndex2,
   CFStringRef key,
   UInt32 bufferSize,
   CFStringEncoding encoding,
   char *value
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

nodeIndex1

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

nodeIndex2

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

key

A reference to a `CFString` object that uniquely identifies the ticket item whose value you want to obtain.

bufferSize

The size of the string specified by the `value` parameter.

encoding

The encoding of the string. You should supply one of the `CFStringBuiltInEncodings` constants defined in Core Foundation String Services.

value

On return, points to the C-style string for the item specified by the `key` parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketGetDouble

Obtains data for a ticket item of type `double`.

```
OSStatus PMTicketGetDouble (
   PMTicketRef ticket,
   UInt32 nodeIndex1,
   UInt32 nodeIndex2,
   CFStringRef key,
   double *value
);
```

Parameters

ticket
A reference to a ticket created by calling the function `PMTicketCreate`.

nodeIndex1
Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

nodeIndex2
Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

key
A reference to a `CFString` object that uniquely identifies the ticket item whose value you want to obtain.

value
On return, points to the `double` value for the item specified by the `key` parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketGetEnumType

Obtains a ticket's ticket type (job, paper, converter, and so on).

```
OSStatus PMTicketGetEnumType (
   PMTicketRef ticket,
   PMTicketType *ticketType
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

ticketType

On return, points to a value of type `PMTicketType` that specifies a ticket's type.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTicket.h`

PMTicketGetIndexPMResolution

Obtains an indexed resolution for a ticket item that's an array of type `PMResolution`.

```
OSStatus PMTicketGetIndexPMResolution (
   PMTicketRef ticket,
   UInt32 nodeIndex1,
   UInt32 nodeIndex2,
   CFStringRef key,
   UInt32 index,
   PMResolution *res
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

nodeIndex1

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

nodeIndex2

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

key

A reference to a `CFString` object that uniquely identifies the ticket item whose value you want to obtain.

index

The index of the resolution you want to obtain.

res

On return, points to the resolution (`PMResolution`) specified by the `index` parameter.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketGetItem

Obtains data for a ticket item of type `CFType`.

```
OSStatus PMTicketGetItem (
   PMTicketRef ticket,
   UInt32 nodeIndex1,
   UInt32 nodeIndex2,
   CFStringRef key,
   CFTypeRef *item
);
```

Parameters

`ticket`
A reference to a ticket created by calling the function `PMTicketCreate`.

`nodeIndex1`
Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

`nodeIndex2`
Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

`key`
A reference to a `CFString` object that uniquely identifies the ticket item whose value you want to obtain.

`item`
On return, point to a Core Foundation type reference for the item specified by the `key` parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketGetLockedState

Obtains the lock state of a ticket.

```
OSStatus PMTicketGetLockedState (
   PMTicketRef ticket,
   Boolean *lockedState
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

lockedState

On return, points to `true` if the ticket is locked and `false` if the ticket is unlocked.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Discussion

Tickets can be locked by the printing system. If you try to modify a locked ticket the result code `kPMTicketLocked` is returned. You can call the function `PMTicketGetLockedState` before you call any other function that modifies the ticket.

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Not available to 64-bit applications.

Declared In

`PMTicketDeprecated.h`

PMTicketGetMetaItem

Obtains data for a item added with the function `PMTicketSetMetaItem`.

Not recommended.

```
OSStatus PMTicketGetMetaItem (
   PMTicketRef ticket,
   CFStringRef key,
   CFTypeRef *item
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

key

A reference to a `CFString` object that uniquely identifies the ticket item whose value you want to obtain.

item

On return, a reference to the data for the item specified by the `key` parameter.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Not available to 64-bit applications.

Declared In
PMTicketDeprecated.h

PMTicketGetPMRect

Obtains data for a ticket item of type PMRect.

```
OSStatus PMTicketGetPMRect (
   PMTicketRef ticket,
   UInt32 nodeIndex1,
   UInt32 nodeIndex2,
   CFStringRef key,
   PMRect *value
);
```

Parameters

ticket
A reference to a ticket created by calling the function PMTicketCreate.

nodeIndex1
Reserved for future use. Currently, you must pass the constant kPMTopLevel.

nodeIndex2
Reserved for future use. Currently, you must pass the constant kPMTopLevel.

key
A reference to a CFString object that uniquely identifies the ticket item whose value you want to obtain.

value
On return, points to the PMRect data for the item specified by the key parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
PMTicket.h

PMTicketGetPMResolution

Obtains data for a ticket item of type PMResolution.

```
OSStatus PMTicketGetPMResolution (
   PMTicketRef ticket,
   UInt32 nodeIndex1,
   UInt32 nodeIndex2,
   CFStringRef key,
   PMResolution *res
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

nodeIndex1

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

nodeIndex2

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

key

A reference to a `CFString` object that uniquely identifies the ticket item whose value you want to obtain.

res

On return, points to the `PMResolution` data for the item specified by the `key` parameter.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTicket.h`

PMTicketGetPPDDict

Obtains data for a ticket item that's a PostScript printer description (PPD) dictionary.

```
OSStatus PMTicketGetPPDDict (
   PMTicketRef ticket,
   UInt32 nodeIndex1,
   UInt32 nodeIndex2,
   CFMutableDictionaryRef *dict
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

nodeIndex1

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

nodeIndex2

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

dict

On return, points to a reference to the PostScript printer description (PPD) dictionary associated with the specified ticket. The dictionary holds pairs of PPD main and option keywords. The main keywords are specified as keys in the dictionary and the options keywords specify the value.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.
Not available to 64-bit applications.

Declared In
`PMTicketDeprecated.h`

PMTicketGetPString

Obtains data for a ticket item that's a Pascal-style string.

```
OSStatus PMTicketGetPString (
   PMTicketRef ticket,
   UInt32 nodeIndex1,
   UInt32 nodeIndex2,
   CFStringRef key,
   UInt32 bufferSize,
   CFStringEncoding encoding,
   StringPtr value
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

nodeIndex1

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

nodeIndex2

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

key

A reference to a `CFString` object that uniquely identifies the ticket item whose value you want to obtain.

bufferSize

The size of the `value` parameter.

encoding

The encoding of the string. You should supply one of the `CFStringBuiltInEncodings` constants defined in Core Foundation String Services.

value

On return, points to the Pascal-style string value for the ticket item specified by the `key` parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketGetRetainCount

Obtains the retention count for a ticket.

```
OSStatus PMTicketGetRetainCount (
   PMTicketRef ticket,
   CFIndex *retainCount
);
```

Parameters

`ticket`
A reference to a ticket created by calling the function `PMTicketCreate`.

`retainCount`
On return, points to the retention count for the ticket.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Discussion
The function `PMTicketGetRetainCount` behaves similarly to the Core Foundation function `CFGetRetainCount`.

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketGetSInt32

Obtains data for a ticket item of type `SInt32`.

```
OSStatus PMTicketGetSInt32 (
   PMTicketRef ticket,
   UInt32 nodeIndex1,
   UInt32 nodeIndex2,
   CFStringRef key,
   SInt32 *value
);
```

Parameters

`ticket`
A reference to a ticket created by calling the function `PMTicketCreate`.

`nodeIndex1`
Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

nodeIndex2

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

key

A reference to a `CFString` object that uniquely identifies the ticket item whose value you want to obtain.

value

On return, points to the `SInt32` value for the ticket item specified by the `key` parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketGetTicket

Obtains a subticket from a ticket.

```
OSStatus PMTicketGetTicket (
   PMTicketRef ticket,
   CFStringRef requestedType,
   UInt32 index,
   PMTicketRef *retrievedTicket
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

requestedType

A string that specifies the ticket type of the ticket you want to obtain. See "Ticket Type Strings" (page 133) for a list of strings you can pass.

index

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

retrievedTicket

On return, points to the ticket reference whose type you specified.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketGetType

Obtains a value that specifies the ticket's type.

```
OSStatus PMTicketGetType (
   PMTicketRef ticket,
   CFStringRef *ticketType
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

ticketType

On return, points to a string value that specifies the ticket's type. See "PMTicketType" (page 103) for a list of constants that can be returned

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTicket.h`

PMTicketGetUInt32

Obtains an item of type `UInt32` from a ticket.

```
OSStatus PMTicketGetUInt32 (
   PMTicketRef ticket,
   UInt32 nodeIndex1,
   UInt32 nodeIndex2,
   CFStringRef key,
   UInt32 *value
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

nodeIndex1

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

nodeIndex2

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

key

A reference to a `CFString` object that uniquely identifies the ticket item whose value you want to obtain.

value

On return, points to `UInt32` value for the ticket item specified by the `key` parameter.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketIsItemLocked

Checks to see if an item in a ticket is locked.

```
OSStatus PMTicketIsItemLocked (
   PMTicketRef ticket,
   CFStringRef key,
   Boolean *locked
);
```

Parameters

`ticket`

A reference to a ticket created by calling the function `PMTicketCreate`.

`key`

A reference to a `CFString` object that uniquely identifies the ticket item you want to check.

`locked`

On return, points to `true` if the item is locked or `false` if the item is not locked.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Discussion
The function `PMTicketIsItemLocked` checks only those items stored in the top-level of the ticket. It does not check items in subtickets.

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.
Not available to 64-bit applications.

Declared In
`PMTicketDeprecated.h`

PMTicketLockItem

Locks an item in a ticket.

```
OSStatus PMTicketLockItem (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

clientID

A reference to a `CFString` object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as `CFSTR("com.myvendorname.myprintingcode")`.

key

A reference to a `CFString` object that uniquely identifies the ticket item you want to lock.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Discussion

The function `PMTicketLockItem` locks only those items stored in the top-level of the ticket. It does not lock items in sub-tickets.

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Not available to 64-bit applications.

Declared In

`PMTicketDeprecated.h`

PMTicketReadXMLFromFile

Restores a ticket previously converted to XML and saved as a file.

```
OSStatus PMTicketReadXMLFromFile (
   CFAllocatorRef allocator,
   const char *path,
   PMTicketRef *ticket,
   CFStringRef *errorString
);
```

Parameters

allocator

A reference to the allocator object to be used for allocating memory. Pass a reference to a valid allocator or `kCFAllocatorDefault` to request the default allocator.

path

A string that specifies the path name of the XML file you want to read.

ticket

A reference to a ticket created by calling the function `PMTicketCreate`. On return, the ticket contains the entries specified by the XML file.

errorString
On return, a reference to a `CFString` object that contains an error message if an error occurred.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketRelease

Decrements the retention count of a ticket object.

```
OSStatus PMTicketRelease (
   PMTicketRef ticket
);
```

Parameters
ticket
A reference to a ticket created by calling the function `PMTicketCreate`.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketReleaseAndClear

Decrements the retention count of a ticket and sets the ticket reference to `NULL`.

```
OSStatus PMTicketReleaseAndClear (
   PMTicketRef *ticket
);
```

Parameters
ticket
A reference to a ticket created by calling the function `PMTicketCreate`.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
PMTicket.h

PMTicketReleaseItem

Removes an item from a ticket.

```
OSStatus PMTicketReleaseItem (
   PMTicketRef ticket,
   CFStringRef key
);
```

Parameters

ticket

A reference to a ticket created by calling the function PMTicketCreate.

key

A reference to a CFString object that uniquely identifies the ticket item you want to remove.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Discussion
An item can only be released if it is not locked. The function PMTicketReleaseItem works differently than the function PMTicketDeleteItem. The function PMTicketDeleteItem makes the item unavailable, but keeps information about the item in the ticket. The function PMTicketReleaseItem removes the item from the ticket.

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
PMTicket.h

PMTicketRemoveTicket

Removes a subticket from a ticket.

```
OSStatus PMTicketRemoveTicket (
   PMTicketRef ticket,
   CFStringRef typeToRemove,
   UInt32 index
);
```

Parameters

ticket

A reference to a ticket created by calling the function PMTicketCreate.

typeToRemove

A reference to a CFString object that specfies the ticket type of the ticket you want to create. See "Ticket Type Strings" (page 133) for a list of constants you can pass.

index

Reserved for future use. Currently, you must pass the constant kPMTopLevel.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketRetain

Increments the retention count of a ticket object.

```
OSStatus PMTicketRetain (
   PMTicketRef ticket
);
```

Parameters

ticket
A reference to a ticket created by calling the function `PMTicketCreate`.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketSetBoolean

Writes an item of type `Boolean` to a ticket.

```
OSStatus PMTicketSetBoolean (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key,
   Boolean value,
   Boolean locked
);
```

Parameters

ticket
A reference to a ticket created by calling the function `PMTicketCreate`.

clientID
A reference to a `CFString` object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as `CFSTR("com.myvendorname.myprintingcode")`.

key

A reference to a `CFString` object that uniquely identifies the ticket item you want to set.

value

The `Boolean` value to which you want to set the ticket item entry specified by the `key` parameter.

locked

Pass `true` to set the item to locked; `false` to set it to unlocked.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketSetBytes

Writes an item that's an array of type `UInt8` to a ticket.

```
OSStatus PMTicketSetBytes (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key,
   const UInt8 *data,
   UInt32 size,
   Boolean locked
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

clientID

A reference to a `CFString` object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as `CFSTR("com.myvendorname.myprintingcode")`.

key

A reference to a `CFString` object that uniquely identifies the ticket item you want to set.

data

A pointer to the data to which you want to set the ticket item entry specified by the `key` parameter.

size

The size of the data, in bytes.

locked

Pass `true` to set the item to locked; `false` to set it to unlocked.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In
PMTicket.h

PMTicketSetCFArray

Writes an item of type CFArray to a ticket.

```
OSStatus PMTicketSetCFArray (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key,
   CFArrayRef item,
   Boolean locked
);
```

Parameters

ticket
A reference to a ticket created by calling the function PMTicketCreate.

clientID
A reference to a CFString object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as CFSTR("com.myvendorname.myprintingcode").

key
A reference to a CFString object that uniquely identifies the ticket item you want to set.

item
A reference to the Core Foundation array to which you want to set the ticket item entry specified by the key parameter.

locked
Pass true to set the item to locked; false to set it to unlocked.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
PMTicket.h

PMTicketSetCFBoolean

Writes an item of type CFBoolean to a ticket.

```
OSStatus PMTicketSetCFBoolean (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key,
   CFBooleanRef item,
   Boolean locked
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

clientID

A reference to a `CFString` object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as `CFSTR("com.myvendorname.myprintingcode")`.

key

A reference to a `CFString` object that uniquely identifies the ticket item you want to set.

item

A reference to the `CFBoolean` value to which you want to set the ticket item entry specified by the `key` parameter.

locked

Pass `true` to set the item to locked; `false` to set it to unlocked.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTicket.h`

PMTicketSetCFData

Writes an item of type `CFData` to a ticket.

```
OSStatus PMTicketSetCFData (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key,
   CFDataRef item,
   Boolean locked
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

clientID

A reference to a `CFString` object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as `CFSTR("com.myvendorname.myprintingcode")`.

key

A reference to a `CFString` object that uniquely identifies the ticket item you want to set.

item

A reference to the `CFData` to which you want to set the ticket item entry specified by the `key` parameter.

locked

Pass `true` to set the item to locked; `false` to set it to unlocked.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketSetCFDate

Writes an item of type `CFDate` to a ticket.

```
OSStatus PMTicketSetCFDate (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key,
   CFDateRef item,
   Boolean locked
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

clientID

A reference to a `CFString` object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as `CFSTR("com.myvendorname.myprintingcode")`.

key

A reference to a `CFString` object that uniquely identifies the ticket item you want to set.

item

A reference to the `CFDate` value to which you want to set the ticket item entry specified by the `key` parameter.

locked

Pass `true` to set the item to locked; `false` to set it to unlocked.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
PMTicket.h

PMTicketSetCFDictionary

Writes an item of type CFDictionary to a ticket.

```
OSStatus PMTicketSetCFDictionary (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key,
   CFDictionaryRef item,
   Boolean locked
);
```

Parameters

ticket
A reference to a ticket created by calling the function PMTicketCreate.

clientID
A reference to a CFString object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as CFSTR("com.myvendorname.myprintingcode").

key
A reference to a CFString object that uniquely identifies the ticket item you want to set.

item
A reference to the Core Foundation dictionary to which you want to set the ticket item entry specified by the key parameter.

locked
Pass true to set the item to locked; false to set it to unlocked.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
PMTicket.h

PMTicketSetCFNumber

Writes an item of type CFNumber to a ticket.

```
OSStatus PMTicketSetCFNumber (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key,
   CFNumberRef item,
   Boolean locked
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

clientID

A reference to a `CFString` object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as `CFSTR("com.myvendorname.myprintingcode")`.

key

A reference to a `CFString` object that uniquely identifies the ticket item you want to set.

item

A reference to the `CFNumber` value to which you want to set the ticket item entry specified by the `key` parameter.

locked

Pass `true` to set the item to locked; `false` to set it to unlocked.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTicket.h`

PMTicketSetCFString

Writes an item of type `CFString` to a ticket.

```
OSStatus PMTicketSetCFString (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key,
   CFStringRef item,
   Boolean locked
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

clientID

A reference to a `CFString` object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as `CFSTR("com.myvendorname.myprintingcode")`.

key

A reference to a `CFString` object that uniquely identifies the ticket item you want to set.

item

A reference to the `CFString` object to which you want to set the ticket item entry specified by the `key` parameter.

locked

Pass `true` to set the item to locked; `false` to set it to unlocked.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketSetCString

Writes an item that's a C-style string to a ticket.

```
OSStatus PMTicketSetCString (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key,
   const char *value,
   Boolean locked
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

clientID

A reference to a `CFString` object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as `CFSTR("com.myvendorname.myprintingcode")`.

key

A reference to a `CFString` object that uniquely identifies the ticket item you want to set.

value

The C-style string to which you want to set the ticket item entry specified by the `key` parameter. The string must use UTF-8 encoding.

locked

Pass `true` to set the item to locked; `false` to set it to unlocked.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
PMTicket.h

PMTicketSetCStringArray

Writes an item that's an array of C-style strings to a ticket.

```
OSStatus PMTicketSetCStringArray (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key,
   const char **cStringArray,
   UInt32 count,
   Boolean locked
);
```

Parameters

ticket

A reference to a ticket created by calling the function PMTicketCreate.

clientID

A reference to a CFString object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as CFSTR("com.myvendorname.myprintingcode").

key

A reference to a CFString object that uniquely identifies the ticket item you want to set.

cStringArray

The array of C-style strings to which you want to set the ticket item entry specified by the key parameter. The strings must use UTF-8 encoding.

count

The number of characters in the C-style string array.

locked

Pass true to set the item to locked; false to set it to unlocked.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
PMTicket.h

PMTicketSetDouble

Writes an item of type double to a ticket.

```
OSStatus PMTicketSetDouble (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key,
   double value,
   Boolean locked
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

clientID

A reference to a `CFString` object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as `CFSTR("com.myvendorname.myprintingcode")`.

key

A reference to a `CFString` object that uniquely identifies the ticket item you want to set.

value

The `double` value to which you want to set the ticket item entry specified by the `key` parameter.

locked

Pass `true` to set the item to locked; `false` to set it to unlocked.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTicket.h`

PMTicketSetDoubleArray

Writes an item that's an array of values of type `double` to a ticket.

```
OSStatus PMTicketSetDoubleArray (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key,
   const double *array,
   UInt32 count,
   Boolean changeable
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

clientID

A reference to a `CFString` object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as `CFSTR("com.myvendorname.myprintingcode")`.

key
A reference to a `CFString` object that uniquely identifies the ticket item you want to set.

doubleArray
A pointer to an array of `double` values to which you want to set the ticket item entry specified by the `key` parameter.

count
The number of values in the array specified by the `doubleArray` parameter.

changeable
Pass `true` to set the item to locked; `false` to set it to unlocked.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketSetItem

Writes an item of type `CFType` to a ticket.

```
OSStatus PMTicketSetItem (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key,
   CFTypeRef item,
   Boolean locked
);
```

Parameters

ticket
A reference to a ticket created by calling the function `PMTicketCreate`.

clientID
A reference to a `CFString` object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as `CFSTR("com.myvendorname.myprintingcode")`.

key
A reference to a `CFString` object that uniquely identifies the ticket item you want to set.

item
A reference to the generic Core Foundation data to which you want to set the ticket item entry specified by the `key` parameter.

locked
Pass `true` to set the item to locked; `false` to set it to unlocked.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketSetMetaItem

Writes an item that does not need to be stored in an XML-representation of a ticket.

Not recommended.

```
OSStatus PMTicketSetMetaItem (
   PMTicketRef ticket,
   CFStringRef key,
   CFTypeRef item
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

key

A reference to a `CFString` object that uniquely identifies the ticket item you want to set.

item

A reference to the generic Core Foundation data to which you want to set the ticket item entry specified by the `key` parameter.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Discussion
You can use the function `PMTicketSetMetaItem` to add an item to a ticket when you don't want that item o be written to an XML-representation of the ticket. In other words, when you want to temporarily add an item to a ticket. Items added with this function cannot be locked.

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.
Not available to 64-bit applications.

Declared In
`PMTicketDeprecated.h`

PMTicketSetPMRect

Writes an item of type `PMRect` to a ticket.

```
OSStatus PMTicketSetPMRect (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key,
   PMRect *value,
   Boolean locked
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

clientID

A reference to a `CFString` object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as `CFSTR("com.myvendorname.myprintingcode")`.

key

A reference to a `CFString` object that uniquely identifies the ticket item you want to set.

value

A pointer to the `PMRect` value to which you want to set the ticket item entry specified by the `key` parameter. A `PMRect` data type is an array of four `double` values.

locked

Pass `true` to set the item to locked; `false` to set it to unlocked.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTicket.h`

PMTicketSetPMRectArray

Writes an item that's an array of values of type `PMRect` to a ticket.

```
OSStatus PMTicketSetPMRectArray (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key,
   PMRect *pmRectArray,
   UInt32 count,
   Boolean locked
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

clientID

A reference to a `CFString` object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as `CFSTR("com.myvendorname.myprintingcode")`.

key

A reference to a `CFString` object that uniquely identifies the ticket item you want to set.

pmRectArray

A pointer to an array of `PMRect` values to which you want to set the ticket item entry specified by the `key` parameter. A `PMRect` data type is an array of four `double` values.

count

The number of values in the array specified by the `pmRectArray` parameter.

locked

Pass `true` to set the item to locked; `false` to set it to unlocked.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTicket.h`

PMTicketSetPMResolution

Writes an item of type `PMResolution` to a ticket.

```
OSStatus PMTicketSetPMResolution (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key,
   PMResolution *value,
   Boolean locked
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

clientID

A reference to a `CFString` object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as `CFSTR("com.myvendorname.myprintingcode")`.

key

A reference to a `CFString` object that uniquely identifies the ticket item you want to set.

value

A pointer to the `PMResolution` value to which you want to set the ticket item entry specified by the `key` parameter. A `PMResolution` data type is an array of two `double` values.

locked

Pass `true` to set the item to locked; `false` to set it to unlocked.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketSetPMResolutionArray

Writes an item that's an array of data of type `PMResolution` to a ticket.

```
OSStatus PMTicketSetPMResolutionArray (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key,
   PMResolution *pmResolutionArray,
   UInt32 count,
   Boolean locked
);
```

Parameters

`ticket`
A reference to a ticket created by calling the function `PMTicketCreate`.

`clientID`
A reference to a `CFString` object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as `CFSTR("com.myvendorname.myprintingcode")`.

`key`
A reference to a `CFString` object that uniquely identifies the ticket item you want to set.

`pmResolutionArray`
A pointer to an array of the `PMResolution` values to which you want to set the ticket item entry specified by the `key` parameter. A `PMResolution` data type is an array of two `double` values.

`count`
The number of items in the array specified by the `pmResolutionArray` parameter.

`locked`
Pass `true` to set the item to locked; `false` to set it to unlocked.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketSetPString

Writes an item that's a Pascal-style string to a ticket.

```
OSStatus PMTicketSetPString (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key,
   ConstStringPtr value,
   Boolean locked
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

clientID

A reference to a `CFString` object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as `CFSTR("com.myvendorname.myprintingcode")`.

key

A reference to a `CFString` object that uniquely identifies the ticket item you want to set.

value

A pointer to the Pascal-style string to which you want to set the ticket item entry specified by the `key` parameter. The string must use MacRoman encoding.

locked

Pass `true` to set the item to locked; `false` to set it to unlocked.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTicket.h`

PMTicketSetSInt32

Writes an item of type `SInt32` to a ticket.

```
OSStatus PMTicketSetSInt32 (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key,
   SInt32 value,
   Boolean locked
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

clientID

A reference to a `CFString` object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as `CFSTR("com.myvendorname.myprintingcode")`.

key

A reference to a `CFString` object that uniquely identifies the ticket item you want to set.

value

The `SInt32` value to which you want to set the ticket item entry specified by the `key` parameter.

locked

Pass `true` to set the item to locked; `false` to set it to unlocked.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTicket.h`

PMTicketSetSInt32Array

Writes an item that's an array of data of type `SInt32` to a ticket.

```
OSStatus PMTicketSetSInt32Array (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key,
   const SInt32 *sInt32Array,
   UInt32 count,
   Boolean locked
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

clientID

A reference to a `CFString` object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as `CFSTR("com.myvendorname.myprintingcode")`.

key

A reference to a `CFString` object that uniquely identifies the ticket item you want to set.

sInt32Array

A pointer to an array of `SInt32` values to which you want to set the ticket item entry specified by the `key` parameter.

count

The number of values in the array specified by the `sInt32Array` parameter.

locked

Pass `true` to set the item to locked; `false` to set it to unlocked.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketSetTemplate

Writes an item that's a job template to a ticket.

```
OSStatus PMTicketSetTemplate (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key,
   PMTemplateRef item,
   Boolean locked
);
```

Parameters

ticket
A reference to a ticket created by calling the function `PMTicketCreate`.

clientID
A reference to a `CFString` object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as `CFSTR("com.myvendorname.myprintingcode")`.

key
A reference to a `CFString` object that uniquely identifies the ticket item you want to set.

item
A reference to a job template created by calling the function `PMTemplateCreate`.

locked
Pass `true` to set the item to locked; `false` to set it to unlocked.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketSetTicket

Writes a subticket to a ticket.

```
OSStatus PMTicketSetTicket (
   PMTicketRef ticket,
   PMTicketRef ticketToAdd,
   UInt32 index
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`. This is the ticket to which you want to add a subticket. Any ticket can contain another ticket.

ticketToAdd

A reference to a ticket created by calling the function `PMTicketCreate`. This is the ticket you want to be a subticket.

index

Reserved for future use. Currently, you must pass the constant `kPMTopLevel`.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTicket.h`

PMTicketSetUInt32

Writes an item of type `UInt32` to a ticket.

```
OSStatus PMTicketSetUInt32 (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key,
   UInt32 value,
   Boolean locked
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

clientID

A reference to a `CFString` object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as `CFSTR("com.myvendorname.myprintingcode")`.

key

A reference to a `CFString` object that uniquely identifies the ticket item you want to set.

value

The `UInt32` value to which you want to set the ticket item entry specified by the `key` parameter.

locked

Pass `true` to set the item to locked; `false` to set it to unlocked.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketSetUInt32Array

Writes an item that's an array of data of type `UInt32` to a ticket.

```
OSStatus PMTicketSetUInt32Array (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key,
   const UInt32 *uInt32Array,
   UInt32 count,
   Boolean locked
);
```

Parameters

ticket
A reference to a ticket created by calling the function `PMTicketCreate`.

clientID
A reference to a `CFString` object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as `CFSTR("com.myvendorname.myprintingcode")`.

key
A reference to a `CFString` object that uniquely identifies the ticket item you want to set.

uInt32Array
A pointer to the array of `UInt32` values to which you want to set the ticket item entry specified by the `key` parameter.

count
The number of values in the array specified by the `uInt32Array` parameter.

locked
Pass `true` to set the item to locked; `false` to set it to unlocked.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketToXML

Converts the data in a ticket to XML.

```
OSStatus PMTicketToXML (
   PMTicketRef ticket,
   CFDataRef *anXMLTicket
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

anXMLTicket

On return, points to a Core Foundation data reference that represents job template data. You are responsible for releasing the `CFData` reference.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Discussion

If you want to write the XML data is to a file, use the function `PMTicketWriteXMLToFile`.

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTicket.h`

PMTicketUnlockItem

Unlocks an item in a ticket.

```
OSStatus PMTicketUnlockItem (
   PMTicketRef ticket,
   CFStringRef clientID,
   CFStringRef key
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

clientID

A reference to a `CFString` object that uniquely identifies your application. The string should be in a format similar to a Java-style package name (think of it as a reverse URL), such as `CFSTR("com.myvendorname.myprintingcode")`.

key

A reference to a `CFString` object that uniquely identifies the ticket item you want to unlock.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Discussion
The function `PMTicketUnlockItem` unlocks only those items stored in the top-level of the ticket. It does not unlock items in subtickets.

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.
Not available to 64-bit applications.

Declared In
`PMTicketDeprecated.h`

PMTicketValidate

Validates a ticket against the constraint values specified in a job template.

```
OSStatus PMTicketValidate (
   PMTicketRef ticket,
   PMTemplateRef verifyingTemplate,
   CFArrayRef *invalidItems
);
```

Parameters

`ticket`
A reference to a ticket created by calling the function `PMTicketCreate`.

`verifyingTemplate`
A reference to the job template against which you want to validate the ticket.

`invalidItems`
On return, points to an array of invalid items, should there be any. If there are no invalid items, and the function returns `noErr`, then the value of `invalidItems` is undefined.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Discussion
Only those items in the ticket that have corresponding entries in the job template are checked. In other words, a ticket item's key must match a template item's key for the ticket item to be validated.

Availability
Not available in CarbonLib.
Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

PMTicketWriteXML

Converts a ticket to XML and then writes it to a file stream.

```
OSStatus PMTicketWriteXML (
   PMTicketRef ticket,
   FILE *file
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

xmlFile

On input, the file to which you want the XML data to be written.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTicket.h`

PMTicketWriteXMLToFile

Converts a ticket to XML and then writes it to a file.

```
OSStatus PMTicketWriteXMLToFile (
   PMTicketRef ticket,
   const char *path
);
```

Parameters

ticket

A reference to a ticket created by calling the function `PMTicketCreate`.

path

On input, the path of the file to which you want the XML data to be written. The function opens the file or creates one if one doesn't exist.

Return Value

A result code. See "Ticket Services Result Codes" (page 134).

Availability

Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In

`PMTicket.h`

PMXMLToTicket

Converts a ticket saved as XML to a ticket.

```
OSStatus PMXMLToTicket (
   CFAllocatorRef allocator,
   CFDataRef anXMLTicket,
   PMTicketRef *ticket,
   CFStringRef *conversionError
);
```

Parameters

allocator

A reference to the allocator object to be used for allocating memory. Pass a reference to a valid allocator or `kCFAllocatorDefault` to request the default allocator.

anXMLTicket

A reference to Core Foundation data that contains previously-converted job template data.

ticket

On return, a reference to a ticket that contains the data converted from the XML data specified by the `anXMLTicket` parameter.

conversionError

On return, a reference to a `CFString` object that specifies errors during the conversion process, if any. Pass `NULL` if you are not interested in getting the errors.

Return Value
A result code. See "Ticket Services Result Codes" (page 134).

Availability
Not available in CarbonLib.

Available in Mac OS X 10.0 and later.

Declared In
`PMTicket.h`

Data Types

ConstCStrList

Represents a static list of C-string pointers.

```
typedef CStrList ConstCStrList;
```

Discussion
For more information see "CStrList" (page 100).

Availability
Available in Mac OS X v10.0 and later.

Declared In
`PMTicket.h`

ConstPMRectList

Represents a static list of `PMRect` data structures.

```
typedef PMRectList ConstPMRectList;
```

Discussion
For more information see "PMRectList" (page 101).

Availability
Available in Mac OS X v10.0 and later.

Declared In
`PMTicket.h`

ConstSInt32List

Represents a static list of `SInt32List` data structures.

```
typedef SInt32List ConstSInt32List;
```

Discussion
Form more information see "SInt32List" (page 103).

Availability
Available in Mac OS X v10.0 and later.

Declared In
`PMTicket.h`

CStrList

Contains an array of `CString` pointers and the number of pointers in the array.

```
struct CStrList {
    SInt32 count;
    char **strArray;
};
typedef struct CStrList CStrList;
```

Fields

`count`
The number of `CString` pointers in the array.

`strArray`
A pointer to an array of `CString` pointers.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`PMTicket.h`

PMPrintingPhaseType

Represents printing phases.

```
typedef UInt16 PMPrintingPhaseType;
```

Discussion
For more information see "Printing Phase Types" (page 126).

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

Declared In
`PMTicket.h`

PMRectList

Contains a list of `PMRect` data structures.

```
struct PMRectList {
    SInt32 count;
    PMRect **pmRectArray;
};
typedef struct PMRectList PMRectList;
```

Fields

`count`
The number of `PMRect` pointers in the array.

`pmRectArray`
A pointer to a list of `PMRect` data structures.

Discussion
A `PMRect` data structure contains a set of four `double` values (top, left, bottom, and right). This structure is used to specify page and paper rectangles.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`PMTicket.h`

PMTemplateRef

Refers to a template object that contains private variables and functions necessary to represent a job template.

```
typedef struct OpaquePMTemplateRef* PMTemplateRef;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`PMTicket.h`

PMTicketErrors

Represents values that indicate error conditions.

```
typedef SInt16 PMTicketErrors;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`PMTicket.h`

PMTicketItemStruct

Contains information about a ticket item.

```
struct PMTicketItemStruct {
    char *key;
    PMTicketItemType itemType;
    Boolean locked;
    union {
        const   void *GenericData;
    const   char *cString;
      SInt32 sInt32;
      UInt32 boolean;
      ConstCStrList *cStrlist;
      PMRect *rect;
      ConstSInt32List *sInt32List;
      ConstPMRectList *pmRectList;
    } value;
};
```

Fields

`key`
A string that uniquely identifies the item.

`itemType`
The type of item defined in the union.

`locked`
The lock state of the item.

`value`
The data associated with the item.

Discussion
You can use this structure to define a static ticket item. An array of these structures can then be converted to a ticket by calling the function `PMTicketFillFromArray`.

PMTicketItemType

Represents a ticket item type.

```
typedef UInt16 PMTicketItemType;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`PMTicket.h`

PMTicketRef

Refers to a ticket object that contains private variables and functions necessary to represent a ticket.

```
typedef struct OpaquePMTicketRef* PMTicketRef;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`PMTicket.h`

PMTicketType

Represents a ticket type.

```
typedef SInt16 PMTicketType;
```

Discussion
For more information see "PMTicketType" (page 103).

Availability
Available in Mac OS X v10.0 and later.

Declared In
`PMTicket.h`

PMValueType

Represents a value type.

```
typedef SInt32 PMValueType;
```

Discussion
For more information see "Item Value Types" (page 110).

Availability
Available in Mac OS X v10.0 and later.

Declared In
`PMTemplate.h`

SInt32List

Contains an array of `SInt32` values.

```
struct SInt32List {
    SInt32 count;
    const SInt32 *sInt32Array;
};
typedef struct SInt32List SInt32List;
```

Fields

`count`

The number of signed 32-bit values in the array.

`sInt32Array`

A pointer to the array.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`PMTicket.h`

Constants

ColorSync Options

Defines strings and keys for items related to ColorSync options.

```
#define kPMColorDeviceIDStr kPMPrinterInfoPrelude "PMColorDeviceID"
#define kPMColorDeviceIDKey CFSTR(kPMColorDeviceIDStr)
#define kPMColorSyncProfilesStr kPMPrinterInfoPrelude "PMColorSyncProfiles"
#define kPMColorSyncProfilesKey CFSTR(kPMColorSyncProfilesStr)
```

Constants

`kPMColorDeviceIDStr`

The color device ID string.

`kPMColorDeviceIDKey`

The value of this key is a `CFString` representing a `CFUUID`; it must be unique per device.

`kPMColorSyncProfilesStr`

The ColorSync profiles sting.

`kPMColorSyncProfilesKey`

The value of this key is a `CFArray` of `CFDictionary` data structures, one `CFDictionary` data structure for each factory profile.

Constraint Types

Specify a constraint type for a job template entry.

```
typedef SInt32 PMConstraintType;
enum {
    kPMConstraintUndefined = 0,
    kPMConstraintRange = 1,
    kPMConstraintList = 2,
    kPMConstraintPrivate = 3
};
```

Constants

`kPMConstraintUndefined`
Values are not constrained.
Available in Mac OS X v10.0 and later.
Declared in `PMTemplate.h`.

`kPMConstraintRange`
Values are constrained by a range of two values, both of type `CFTypeRef`.
Available in Mac OS X v10.0 and later.
Declared in `PMTemplate.h`.

`kPMConstraintList`
Values are constrained by an array.
Available in Mac OS X v10.0 and later.
Declared in `PMTemplate.h`.

`kPMConstraintPrivate`
Values are constrained privately, and should not be checked by the system.
Available in Mac OS X v10.0 and later.
Declared in `PMTemplate.h`.

Converter Setup Ticket Keys

Defines strings and keys for items related to a converter setup ticket.

```
#define kPMConverterSetupPrelude "com.apple.print.ConverterSetup."
#define kPMBandingRequestedStr kPMConverterSetupPrelude "PMBandingRequested"
#define kPMBandingRequestedKey CFSTR(kPMBandingRequestedStr)
#define kPMRequiredBandHeightStr kPMConverterSetupPrelude "PMRequiredBandHeight"
#define kPMRequiredBandHeightKey CFSTR(kPMRequiredBandHeightStr)
#define kPMDepthSwitchingEnabledStr kPMConverterSetupPrelude "PMDepthSwitching"
#define kPMDepthSwitchingEnabledKey CFSTR(kPMDepthSwitchingEnabledStr)
#define kPMWhiteSkippingEnabledStr kPMConverterSetupPrelude "PMWhiteSpaceSkipping"
#define kPMWhiteSkippingEnabledKey CFSTR(kPMWhiteSkippingEnabledStr)
#define kPMConverterResHorizontalStr kPMConverterSetupPrelude
"PMConversionResHorizontal"
#define kPMConverterResHorizontalKey CFSTR(kPMConverterResHorizontalStr)
#define kPMConverterResVerticalStr kPMConverterSetupPrelude "PMConversionResVertcial"
#define kPMConverterResVerticalKey CFSTR(kPMConverterResVerticalStr)
#define kPMRequestedPixelFormatStr kPMConverterSetupPrelude "PMPixelFormat"
#define kPMRequestedPixelFormatKey CFSTR(kPMRequestedPixelFormatStr)
#define kPMRequestedPixelLayoutStr kPMConverterSetupPrelude "PMPixelLayout"
#define kPMRequestedPixelLayoutKey CFSTR(kPMRequestedPixelLayoutStr)
#define kPMCVColorSyncProfileIDKey CFSTR(kPMCVColorSyncProfileIDStr)
```

Constants

`kPMConverterSetupPrelude`
: The converter-setup prelude sting.

`kPMBandingRequestedStr`
: The banding-requested string.

`kPMBandingRequestedKey`
: The value of this key is a `CFBoolean`; turns banding on if it's available.

`kPMRequiredBandHeightStr`
: The banding-height string.

`kPMRequiredBandHeightKey`
: The value of this key is a `CFNumber` value; specifies the number of scan lines needed for each band. If it specifies the whole page, banding is disabled.

`kPMDepthSwitchingEnabledStr`
: The depth-switching-enabled string.

`kPMDepthSwitchingEnabledKey`
: The value of this key is a `CFBoolean` value; `true` specifies the printer module requests the converter to switch between black & white and color bands when possible.

`kPMWhiteSkippingEnabledStr`
: The white-skipping-enabled string.

`kPMWhiteSkippingEnabledKey`
: The value of this key is a `CFBoolean`; `true` specifies the printer module requests the converter to skip over white space if possible.

`kPMConverterResHorizontalStr`
: The horizontal rendering resolution string.

`kPMConverterResHorizontalKey`
: The value of this key is a `CFNumber` of type `CFNumberDoubleType`; specifies the final horizontal rendering resolution.

`kPMConverterResVerticalStr`
: The vertical resolution string.

`kPMConverterResVerticalKey`
: The value of this key is a `CFNumber` of type `CFNumberDoubleType`; specifies the final vertical rendering resolution.

`kPMRequestedPixelFormatStr`
: The pixel format string.

`kPMRequestedPixelFormatKey`
: The value of this key is a `CFNumber` of type `CFNumberLongType`; specifies the pixel format requested of the converter.

`kPMRequestedPixelLayoutStr`
: The pixel layout string.

`kPMRequestedPixelLayoutKey`
: The value of this key is a `CFNumber` of type `CFNumberLongType`; specifies the pixel layout requested of the converter.

`kPMCVColorSyncProfileIDKey`
: The value of this key is a `CFNumber` of type `kCFNumberSInt32Type`; specifies the profile ID for the ColorSync profile to be used with the print job.

Data Transmission Keys

Defines strings and keys for items related to data transmission.

```
#define kPMIsBinaryOKStr            kPMPrinterInfoPrelude "PMIsBinaryOK"
#define kPMIsBinaryOKKey            CFSTR(kPMIsBinaryOKStr)
#define kPM8BitCommStr            kPMPrinterInfoPrelude "PM8BitComm"
#define kPM8BitCommKey            CFSTR(kPM8BitCommStr)
#define kPMTransparentCommStr           kPMPrinterInfoPrelude "PMTransparentComm"
#define kPMTransparentCommKey            CFSTR(kPMTransparentCommStr)
```

Constants

`kPMIsBinaryOKStr`
: The binary is okay string.

`kPMIsBinaryOKKey`
: The value of this key is a `CFBoolean` representing the result of querying the PostScript printer about its ability to accept binary data. It is possible for the underlying communications to support binary data, both high bit characters and control characters, but for a spooler on the other end of the channel to not accept binary data. This value represents the spooler/printer's abilities.

`kPM8BitCommStr`
: The 8-bit communication string.

`kPM8BitCommKey`
: The value of this key is a `CFBoolean` indicating whether the communications channel can transmit characters in the range 0x80 -)xFF inclusive without them being damaged or interpreted as channel control characters.

`kPMTransparentCommStr`
: The transparent communication string.

`kPMTransparentCommKey`
: The value of this key is a `CFBoolean` indicating whether the communications channel can transmit characters in the range 0x00 - 0x1F inclusive without them being damaged or interpreted as channel control characters.

Document Ticket Keys

Defines strings and keys for items in a document ticket; currently not used.

```
#define kPMDocumentTicketPrelude "com.apple.print.DocumentTicket."
#define kPMSpoolFormatStr kPMDocumentTicketPrelude "PMSpoolFormat"
#define kPMSpoolFormatKey CFSTR(kPMSpoolFormatStr)
#define kPMPrinterModuleFormatStr kPMDocumentTicketPrelude "PMDocPMInputFormat"
#define kPMPrinterModuleFormatKey CFSTR(kPMPrinterModuleFormatStr)
```

Constants

`kPMDocumentTicketPrelude`
Currently not used.

`kPMSpoolFormatStr`
Currently not used.

`kPMSpoolFormatKey`
Currently not used.

`kPMPrinterModuleFormatStr`
Currently not used.

`kPMPrinterModuleFormatKey`
Currently not used.

Drawing Resolution Keys

Defines strings and keys for items in related to drawing resolution.

```
#define kPMPrinterSuggestedResStr kPMPrinterInfoPrelude
"PMPrinterSuggestedRes"
#define kPMPrinterSuggestedResKey CFSTR(kPMPrinterSuggestedResStr
        #define kPMPrinterMinResStr kPMPrinterInfoPrelude "PMPrinterMinRes"
#define kPMPrinterMinResKey CFSTR(kPMPrinterMinResStr)
#define kPMPrinterMaxResStr kPMPrinterInfoPrelude "PMPrinterMaxRes"
#define kPMPrinterMaxResKey CFSTR(kPMPrinterMaxResStr)
```

Constants

`kPMPrinterSuggestedResStr`
The suggested application drawing resolutions string.

`kPMPrinterSuggestedResKey`
The value of this key is the suggested application drawing resolutions key.

`kPMPrinterMinResStr`
The minimum drawing resolution string.

`kPMPrinterMinResKey`
The value of this key is the minimum range of resolutions for the printer; specified as a `CFArray` of two `CFNumber` values of type `kCFNumberDoubleType`.

`kPMPrinterMaxResStr`
The maximum printer resolution string.

`kPMPrinterMaxResKey`
The value of this key is the maximum range of resolutions for the printer; specified as a `CFArray` of two `CFNumber` values of type `kCFNumberDoubleType`.

Duplex Options

Specify values for the duplex options key (`kPMDuplexingKey`.)

```
enum {
    kPMDuplexNone = 1,
    kPMDuplexNoTumble = 2,
    kPMDuplexTumble = 3,
    kPMSimplexTumble = 4,
    kPMDuplexDefault = 1
};
```

Constants

`kPMDuplexNone`
Don't use duplex.
Available in Mac OS X v10.0 and later.
Declared in `PMDefinitions.h`.

`kPMDuplexNoTumble`
Print on both sides of the paper; flip pages from left to right.
Available in Mac OS X v10.0 and later.
Declared in `PMDefinitions.h`.

`kPMDuplexTumble`
Print on both sides of the paper; tumbling so pages flip top to bottom.
Available in Mac OS X v10.0 and later.
Declared in `PMDefinitions.h`.

`kPMSimplexTumble`
Print on only one side of the paper, but tumble the images while printing.
Available in Mac OS X v10.0 and later.
Declared in `PMDefinitions.h`.

`kPMDuplexDefault`
Don't use duplex; this option is the same as `kPMDuplexNone`.

Error Handling Options

Specify whether or not an error handler is available.

```
enum {
    kPSNoErrorHandler = 0,
    kPSErrorHandler = 1
};
```

Constants

`kPSNoErrorHandler`
Specifies that an error handler is not available.
Available in Mac OS X v10.0 and later.
Declared in `PMTicket.h`.

`kPSErrorHandler`
Specifies that an error handler is available.

Available in Mac OS X v10.0 and later.

Declared in `PMTicket.h`.

Fetch options

Specifies whether an item should be fetched.

```
#define kPMDontFetchItem NULL
```

Constants

`kPMDontFetchItem`
Don't fetch the item.

Installable Options

Defines a string and key for installable printer options.

```
#define kPMInstallableOptionStr kPMPrinterInfoPrelude
"PMInstallableOption"
#define kPMInstallableOptionKey CFSTR(kPMInstallableOptionStr)
```

Constants

`kPMInstallableOptionStr`
The installable options string.

`kPMInstallableOptionKey`
The value of this key is a packed array of Pascal strings that specifies the installable options in the PostScript printer description (PPD) file. The strings are as key-value pairs of PPD main and option keywords.

Item Value Types

Specify the data type of a ticket or template item.

```
typedef UInt16 PMTicketItemType;
enum {
    kPMItemInvalidType = 0,
    kPMItemCStringType = 1,
    kPMItemSInt32Type = 2,
    kPMItemBooleanType = 3,
    kPMItemCStrListType = 4,
    kPMItemPMRectType = 5,
    kPMItemSInt32ListType = 6,
    kPMItemPMRectListType = 7
};
```

Constants

`kPMItemInvalidType`
The type is not valid.
Available in Mac OS X v10.0 and later.
Declared in `PMTicket.h`.

`kPMItemCStringType`
A C-style string pointer.
Available in Mac OS X v10.0 and later.
Declared in `PMTicket.h`.

`kPMItemSInt32Type`
A signed 32-bit integer.
Available in Mac OS X v10.0 and later.
Declared in `PMTicket.h`.

`kPMItemBooleanType`
A `Boolean` value.
Available in Mac OS X v10.0 and later.
Declared in `PMTicket.h`.

`kPMItemCStrListType`
A list of C-style strings
Available in Mac OS X v10.0 and later.
Declared in `PMTicket.h`.

`kPMItemPMRectType`
A pointer to a `PMRect` data structure.
Available in Mac OS X v10.0 and later.
Declared in `PMTicket.h`.

`kPMItemSInt32ListType`
A pointer to an `SInt32List` data structure.
Available in Mac OS X v10.0 and later.
Declared in `PMTicket.h`.

`kPMItemPMRectListType`
A pointer to a `PMRectList` data structure.
Available in Mac OS X v10.0 and later.
Declared in `PMTicket.h`.

Job Ticket Keys

Defines strings and keys for items in a job ticket.

```
#define kPMJobTicketPrelude "com.apple.print.JobInfo."
#define kPMJobNameStr kPMJobTicketPrelude "PMJobName"
#define kPMJobNameKey CFSTR(kPMJobNameStr)
#define kPMApplicationNameStr kPMJobTicketPrelude "PMApplicationName"
#define kPMApplicationNameKey CFSTR(kPMApplicationNameStr)
#define kPMUserLanguageStr kPMJobTicketPrelude "PMUserLanguage"
#define kPMUserLanguageKey CFSTR(kPMUserLanguageStr)
#define kPMJobOwnerStr kPMJobTicketPrelude "PMJobOwner"
#define kPMJobOwnerKey CFSTR(kPMJobOwnerStr)
#define kPMJobTemplateStr kPMJobTicketPrelude "PMJobTemplate"
#define  CFSTR(kPMJobTemplateStr)
#define kPMPhaseStr kPMJobTicketPrelude "PMPrintingPhase"
#define kPMPhaseKey CFSTR(kPMPhaseStr)
#define kPMOutputTypeStr kPMJobTicketPrelude "PMOutputType"
#define kPMOutputTypeKey CFSTR(kPMOutputTypeStr)
```

Constants

`kPMJobTicketPrelude`
The job ticket prelude string.

`kPMJobNameStr`
The job name string.

`kPMJobNameKey`
The value of this key is a `CFString` that specifies the name of the job to be displayed in the print queue dialog.

`kPMApplicationNameStr`
The application name string.

`kPMApplicationNameKey`
The value of this key is a `CFString` that specifies the application's name.

`kPMUserLanguageStr`
The user language string.

`kPMUserLanguageKey`
The value of this key is a `CFNumber` of type `kCFNumberSInt32Type`.

`kPMJobOwnerStr`
The job owner string.

`kPMJobOwnerKey`
The value of this key is a `CFString` that specifies the name of the user who submitted the job.

`kPMJobTemplateStr`
The job template string.

`kPMJobTemplateKey`
The value of this key is a `CFDictionary` that is actually a `PMTemplateRef`.

`kPMPhaseStr`
The printing phase string.

`kPMPhaseKey`
The value of this key is a `CFNumber` of type `kCFNumberSInt32Type`; specifies an enumeration—Spooling, RIPing, and so forth.

`kPMOutputTypeStr`
The output type string.

`kPMOutputTypeKey`
The value of this key is a `CFString` that specifies a Mime type from the `kPMOutputTypeListKey` array generated by the printer module.

Default Copy/Collate Value

Specifies the default value for copy/collate.

```
enum {
    kPMCopyCollateDefault = 1
};
```

Constants

`kPMCopyCollateDefault`
Use the default value for copy/collate.

List Ticket Keys

Defines strings and keys for items in a list ticket.

```
#define kPMTicketListPrelude "com.apple.print.TicketList."
```

Constants

`kPMTicketListPrelude`
The list ticket prelude string.

Lock State

Specifies whether an item is locked or unlocked.

```
enum {
    kPMUnlocked = 0,
    kPMLocked = 1
};
```

Constants

`kPMUnlocked`
Indicates items is unlocked.

Available in Mac OS X v10.2 and later.

Declared in `PMDefinitions.h`.

`kPMLocked`
Indicates items is locked.

Available in Mac OS X v10.2 and later.

Not available to 64-bit applications.

Declared in `PMDefinitionsDeprecated.h`.

Memory Keys

Defines strings and keys for items related to printer memory.

```
#define kPMTotalMemInstalledStr kPMPrinterInfoPrelude
"PMTotalMemInstalled" #define kPMTotalMemInstalledKey
CFSTR(kPMTotalMemInstalledStr)
#define kPMTotalMemAvailableStr kPMPrinterInfoPrelude "PMTotalMemAvailable"
#define kPMTotalMemAvailableKey CFSTR(kPMTotalMemAvailableStr)
```

Constants

`kPMTotalMemInstalledStr`
: The memory installed string.

`kPMTotalMemInstalledKey`
: The value of this key is a `CFNumber` of type `kCFNumberSInt32Type`; specifies the memory installed in the printer.

`kPMTotalMemAvailableStr`
: The total memory available string.

`kPMTotalMemAvailableKey`
: The value of this key is a `CFNumber` of type `kCFNumberSInt32Type`; specifies the remaining memory available for use.

Page Format Ticket Keys

Defines strings and keys for items in a page format ticket.

```
#define kPMPageFormatPrelude "com.apple.print.PageFormat."
#define kPMAdjustedPaperRectStr kPMPageFormatPrelude "PMAdjustedPaperRect"
#define kPMAdjustedPaperRectKey CFSTR(kPMAdjustedPaperRectStr)
#define kPMAdjustedPageRectStr kPMPageFormatPrelude "PMAdjustedPageRect"
#define kPMAdjustedPageRectKey CFSTR(kPMAdjustedPageRectStr)
#define kPMDrawingResHorizontalStr kPMPageFormatPrelude "PMHorizontalRes"
#define kPMDrawingResHorizontalKey CFSTR(kPMDrawingResHorizontalStr)
#define kPMDrawingResVerticalStr kPMPageFormatPrelude "PMVerticalRes"
#define kPMDrawingResVerticalKey CFSTR(kPMDrawingResVerticalStr)
#define kPMPageScalingHorizontalStr kPMPageFormatPrelude "PMScaling"
#define kPMPageScalingHorizontalKey CFSTR(kPMPageScalingHorizontalStr)
#define kPMPageScalingVerticalStr kPMPageFormatPrelude "PMVerticalScaling"
#define kPMPageScalingVerticalKey CFSTR(kPMPageScalingVerticalStr)
#define kPMPageOrientationStr kPMPageFormatPrelude "PMOrientation"
#define kPMPageOrientationKey CFSTR(kPMPageOrientationStr)
#define kPMPageBackupRecordHdlStr kPMPageFormatPrelude "BackupPrintRecordHandle"
#define kPMPageBackupRecordHdlKey CFSTR(kPMPageBackupRecordHdlStr)
#define kPMPageBackupRecordDataStr kPMPageFormatPrelude "BackupPrintRecord"
#define kPMPageBackupRecordDataKey CFSTR kPMPageBackupRecordDataStr)
#define kPMPageCustomDialogHdlStr kPMPageFormatPrelude "CustomDialogRecord"
#define kPMPageCustomDialogHdlKey CFSTR(kPMPageCustomDialogHdlStr)
#define kPMFormattingPrinterStr kPMPageFormatPrelude "FormattingPrinter"
#define kPMFormattingPrinterKey CFSTR(kPMFormattingPrinterStr)
```

Constants

`kPMPageFormatPrelude`
: The page format ticket prelude string.

`kPMAdjustedPaperRectStr`
The adjusted paper rectangle string.

`kPMAdjustedPaperRectKey`
The value of this key is a `CFArray` of four `CFNumbers` of type `kCFNumberDoubleType` for the adjusted paper rectangle, in points.

`kPMAdjustedPageRectStr`
The adjusted page rectangle string.

`kPMAdjustedPageRectKey`
The value of this key is a `CFArray` of four `CFNumbers` of type `kCFNumberDoubleType` for the adjusted page rectangle, in points.

`kPMDrawingResHorizontalStr`
The horizontal drawing resolution string.

`kPMDrawingResHorizontalKey`
The value of this key is a `CFNumber` of type `kCFNumberDoubleType`; the drawing resolution in horizontal direction.

`kPMDrawingResVerticalStr`
The vertical drawing resolution string.

`kPMDrawingResVerticalKey`
The value of this key is a `CFNumber` of type `kCFNumberDoubleType`; the drawing resolution in vertical direction.

`kPMPageScalingHorizontalStr`
The horizontal page scaling string.

`PMPageScalingHorizontalKey`
The value of this key is a `CFNumber` of type `kCFNumberDoubleType`; the horizontal scaling factor applied to original page size.

`kPMPageScalingVerticalStr`
The vertical page scaling string.

`kPMPageScalingVerticalKey`
The value of this key is a `CFNumber` of type `kCFNumberDoubleType`; the vertical scaling factor applied to original page size.

`kPMPageOrientationStr`
The page orientation string.

`kPMPageOrientationKey`
The value of this key is a `CFNumber` of type `kCFNumberSInt32Type`; `PMOrientation`, 1 = portrait, 2 = landscape, 3 = reverse portrait, 4 = reverse landscape.

`kPMPageBackupRecordHdlStr`
The page backup record handle string.

`kPMPageBackupRecordHdlKey`
The value of this key is a print record handle (`CFData`); not used when data is flattened. (Mac OS 8 and 9 only)

`kPMPageBackupRecordDataStr`
The page backup record data string.

`kPMPageBackupRecordDataKey`
The value of this key is a print record (`CFData`) stored in complete form; used when flattening ticket with the print record. ((Mac OS 8 and 9 only)

`kPMPageCustomDialogHdlStr`
The page custom dialog handle string.

`kPMPageCustomDialogHdlKey`
: The value of this key is a handle (`CFData`) to the print record using for custom dialog calls; not stored when flattened. (Mac OS 8 and 9 only)

`kPMFormattingPrinterStr`
: The formatting printer string.

`kPMFormattingPrinterKey`
: The value of this key is a `CFString` that specifies name of the formatting printer.

Page Ticket Key

Defines strings and keys for items in a page ticket; currently not used.

```
#define kPMPageTicketPrelude "com.apple.print.PageTicket."
```

Constants

`kPMPageTicketPrelude`
: Currently not used.

Paper Info Ticket Keys

Defines strings and keys for items in a paper info ticket.

```
#define kPMPaperInfoPrelude "com.apple.print.PaperInfo."
#define kPMPaperNameStr kPMPaperInfoPrelude "PMPaperName"
#define kPMPaperNameKey CFSTR(kPMPaperNameStr)
#define kPMUnadjustedPaperRectStr kPMPaperInfoPrelude "PMUnadjustedPaperRect"
#define kPMUnadjustedPaperRectKey CFSTR(kPMUnadjustedPaperRectStr)
#define kPMUnadjustedPageRectStr kPMPaperInfoPrelude "PMUnadjustedPageRect"
#define kPMUnadjustedPageRectKey CFSTR(kPMUnadjustedPageRectStr)
#define kPMMatchPaperStr kPMPaperInfoPrelude "PMMatchPaper"
#define kPMMatchPaperKey CFSTR(kPMMatchPaperStr)
```

Constants

`kPMPaperInfoPrelude`
: The paper info ticket prelude string.

`kPMPaperNameStr`
: The paper name string.

`kPMPaperNameKey`
: The paper name key specifies the name of the paper displayed in the user interface.

`kPMUnadjustedPaperRectStr`
: The unadjusted paper rectangle string.

`kPMUnadjustedPaperRectKey`
: The unadjusted paper rectangle key specifies the unadjusted paper rectangle.

`kPMUnadjustedPageRectStr`
: The unadjusted page rectangle string.

`kPMUnadjustedPageRectKey`
: The unadjusted page rectangle key specifies the unadjusted page rectangle.

`kPMMatchPaperStr`
: The paper match string.

`kPMMatchPaperKey`
The paper match key specifies how closely the printing system must match the specified paper.

PostScript Language Level Targets

Specify language-level targets for PostScript printing.

```
enum {
    kPMPSTargetLanguageLevel2and3 = -3,
    kPMPSTargetLanguageLevel1and2 = -2,
    kPMPSTargetLanguageLevelUnknown = -1,
    kPMPSTargetLanguageLevel1 = 1,
    kPMPSTargetLanguageLevel2 = 2,
    kPMPSTargetLanguageLevel3 = 3,
    kPMPSTargetLanguageLevelDefault = -1
};
```

Constants

`kPMPSTargetLanguageLevel2and3`
Level 2 compatible, may take advantage of Level 3 features.

Available in Mac OS X v10.1 and later.

Declared in `PMTicket.h`.

`kPMPSTargetLanguageLevel1and2`
Level 1 compatible, may take advantage of Lever 2 and 3 features.

Available in Mac OS X v10.1 and later.

Declared in `PMTicket.h`.

`kPMPSTargetLanguageLevelUnknown`
Language level of target is unknown.

Available in Mac OS X v10.1 and later.

Declared in `PMTicket.h`.

`kPMPSTargetLanguageLevel1`
Level 1.

Available in Mac OS X v10.1 and later.

Declared in `PMTicket.h`.

`kPMPSTargetLanguageLevel2`
Level 2.

Available in Mac OS X v10.1 and later.

Declared in `PMTicket.h`.

`kPMPSTargetLanguageLevel3`
Level 3.

Available in Mac OS X v10.1 and later.

Declared in `PMTicket.h`.

`kPMPSTargetLanguageLevelDefault`
Same as `kPMPSTargetLanguageLevelUnknown`.

Available in Mac OS X v10.1 and later.

Declared in `PMTicket.h`.

PostScript Printer Description Tags

Defines strings and keys for items related to PostScript printer description files.

```
#define kPMDescriptionFileStr kPMPrinterInfoPrelude
"PMDescriptionFile"
#define kPMDescriptionFileKey CFSTR(kPMDescriptionFileStr)
#define kPMCompiledPPDStr kPMPrinterInfoPrelude "PMCompiledPPD"
#define kPMCompiledPPDKey CFSTR(kPMCompiledPPDStr)
```

Constants

`kPMDescriptionFileStr`
The printer description file string.

`kPMDescriptionFileKey`
The value of this key is a `CFString`; specifies the PostScript printer description file name or other description file.

`kPMCompiledPPDStr`
The compiled PostScript printer description string.

`kPMCompiledPPDKey`
The value of this key is a compiled PostScript printer description (`CFData`).

PostScript Printer Driver Keys

Defines a string and key for a PostScript printer driver.

```
#define kPMPrinterIsPostScriptDriverStr kPMPrinterInfoPrelude "PMIsPostScriptDriver"
#define kPMPrinterIsPostScriptDriverKey CFSTR(kPMPrinterIsPostScriptDriverStr)
```

Constants

`kPMPrinterIsPostScriptDriverStr`
The PostScript printer driver string.

`kPMPrinterIsPostScriptDriverKey`
The value of this key is a `CFBoolean` value.

Print Settings Ticket Keys

Defines strings and keys for items in a print settings ticket.

```
#define kPMBorderStr kPMPrintSettingsPrelude "PMBorder"
#define kPMBorderKey CFSTR(kPMBorderStr)
#define kPMBorderTypeStr kPMPrintSettingsPrelude "PMBorderType"
#define kPMBorderTypeKey CFSTR(kPMBorderTypeStr)
#define kPMLayoutNUpStr kPMPrintSettingsPrelude "PMLayoutNUp"
#define kPMLayoutNUpKey CFSTR(kPMLayoutNUpStr
#define kPMLayoutRowsStr kPMPrintSettingsPrelude "PMLayoutRows"
#define kPMLayoutRowsKey CFSTR(kPMLayoutRowsStr)
#define kPMLayoutColumnsStr kPMPrintSettingsPrelude "PMLayoutColumns"
#define kPMLayoutColumnsKey CFSTR(kPMLayoutColumnsStr)
#define kPMLayoutDirectionStr kPMPrintSettingsPrelude "PMLayoutDirection"
#define kPMLayoutDirectionKey CFSTR(kPMLayoutDirectionStr)
#define kPMLayoutTileOrientationStr kPMPrintSettingsPrelude "PMLayoutTileOrientation"
#define kPMLayoutTileOrientationKey CFSTR(kPMLayoutTileOrientationStr)
#define kPMQualityStr kPMPrintSettingsPrelude "PMQuality"
#define kPMQualityKey CFSTR(kPMQualityStr)
#define kPMPaperTypeStr kPMPrintSettingsPrelude "PMPaperType"
#define kPMPaperTypeKey CFSTR(kPMPaperTypeStr)
#define kPMJobStateStr kPMPrintSettingsPrelude "PMJobState"
#define kPMJobStateKey CFSTR(kPMJobStateStr)
#define kPMJobHoldUntilTimeStr kPMPrintSettingsPrelude "PMJobHoldUntilTime"
#define kPMJobHoldUntilTimeKey CFSTR(kPMJobHoldUntilTimeStr)
#define kPMJobPriorityStr kPMPrintSettingsPrelude "PMJobPriority"
#define kPMJobPriorityKey CFSTR(kPMJobPriorityStr)
#define kPMPaperSourceStr kPMPrintSettingsPrelude "PMPaperSource"
#define kPMPaperSourceKey CFSTR(kPMPaperSourceStr)
#define kPMDuplexingStr kPMPrintSettingsPrelude "PMDuplexing"
#define kPMDuplexingKey CFSTR(kPMDuplexingStr)
#define kPMColorModeStr kPMPrintSettingsPrelude "PMColorMode"
#define kPMColorModeKey CFSTR(kPMColorModeStr)
#define kPMColorSyncProfileIDStr kPMPrintSettingsPrelude "PMColorSyncProfileID"
#define kPMColorSyncProfileIDKey            CFSTR(kPMColorSyncProfileIDStr)
#define kPMColorSyncSystemProfilePathStr    kPMPrintSettingsPrelude
"PMColorSyncSystemProfilePath"
#define kPMColorSyncSystemProfilePathKey CFSTR(kPMColorSyncSystemProfilePathStr)
#define kPMPrintScalingHorizontalStr kPMPrintSettingsPrelude "PMScaling"
#define kPMPrintScalingHorizontalKey CFSTR(kPMPrintScalingHorizontalStr)
#define kPMPrintScalingVerticalStr kPMPrintSettingsPrelude "PMVerticalScaling"
#define kPMPrintScalingVerticalKey CFSTR(kPMPrintScalingVerticalStr)
#define kPMPrintScalingAlignmentStr kPMPrintSettingsPrelude "PMScalingAlignment"
#define kPMPrintScalingAlignmentKey CFSTR(kPMPrintScalingAlignmentStr)
#define kPMPrintOrientationStr kPMPrintSettingsPrelude "PMOrientation"
#define kPMPrintOrientationKey CFSTR(kPMPrintOrientationStr)
#define kPMPreviewStr kPMPrintSettingsPrelude "PMPreview"
#define kPMPreviewKey CFSTR(kPMPreviewStr
#define kPMPrintBackupRecordHdlStr kPMPrintSettingsPrelude "BackupPrintRecordHandle"
#define kPMPrintBackupRecordHdlKey CFSTR(kPMPrintBackupRecordHdlStr)
#define kPMPrintBackupRecordDataStr kPMPrintSettingsPrelude "BackupPrintRecord"
#define kPMPrintBackupRecordDataKey CFSTR(kPMPrintBackupRecordDataStr)
#define kPMPrintCustomDialogHdlStr kPMPrintSettingsPrelude "CustomDialogRecord"
#define kPMPrintCustomDialogHdlKey CFSTR(kPMPrintCustomDialogHdlStr)
#define kPMPrimaryPaperFeedStr                kPMPrintSettingsPrelude
"PMPrimaryPaperFeed"
#define kPMPrimaryPaperFeedKey                CFSTR(kPMPrimaryPaperFeedStr)
#define kPMSecondaryPaperFeedStr            kPMPrintSettingsPrelude
"PMSecondaryPaperFeed"
#define kPMSecondaryPaperFeedKey            CFSTR(kPMSecondaryPaperFeedStr)
#define kPMPSErrorHandlerStr                kPMPrintSettingsPrelude
```

```
"PMPSErrorHandler"
#define kPMPSErrorHandlerKey                CFSTR(kPMPSErrorHandlerStr)
#define kPMPSErrorOnScreenStr                kPMPrintSettingsPrelude
"PMPSErrorOnScreen"
#define kPMPSErrorOnScreenKey                CFSTR(kPMPSErrorOnScreenStr)
#define kPMPSTraySwitchStr                    kPMPrintSettingsPrelude
"PMPSTraySwitch"
#define kPMPSTraySwitchKey                    CFSTR(kPMPSTraySwitchStr)
#define kPMPPDDictStr     kPMPrintSettingsPrelude "kPMPPDDictStr"
#define kPMPPDDictKey CFSTR(kPMPPDDictStr)
```

Constants

`kPMPrintSettingsPrelude`
The print settings ticket prelude string.

`kPMDestinationTypeStr`
The destination type string.

`kPMDestinationTypeKey`
The value of this key is a `CFNumber` of type `kCFNumberSInt32Type`; can be `kPMDestinationPrinter`, `kPMDestinationFile`, or `kPMDestinationFax`.

`kPMOutputFilenameStr`
The output filename string.

`kPMOutputFilenameKey`
The value of this key is a `CFString`; a URL that specifies the output filename.

`kPMCopiesStr`
The copies string.

`kPMCopiesKey`
The value of this key is a `CFNumber` of type `kCFNumberSInt32Type` that specifies number of copies to print.

`kPMCopyCollateStr`
The copy/collate string.

`kPMCopyCollateKey`
The value of this key is a `CFBoolean` value; used to turn collating on.

`kPMReverseOrderStr`
The reverse order string.

`kPMReverseOrderKey`
The value of this key is a `CFBoolean` value; `true` specifies to print sheets back to front. All layout options are unaffected by reverse order.

`kPMPageRangeStr`
The page range string.

`kPMPageRangeKey`
The value of this key is a `CFArray` of type `kCFNumberSInt32Type`; indicates valid range of pages that the application is able to print.

`kPMFirstPageStr`
The first page string.

`kPMFirstPageKey`
The value of this key is a `CFNumber` of type `kCFNumberSInt32Type`; the first page selected by user to print.

`kPMLastPageStr`
The last page string.

`kPMLastPageKey`
The value of this key is a `CFNumber` of type `kCFNumberSInt32Type`; the last page selected by user to print.

`kPMBorderStr`
The border string.

`kPMBorderKey`
The value of this key is a `CFBoolean`; `true` specifies to use borders.

`kPMBorderTypeStr`
The border type string.

`kPMBorderTypeKey`
The value of this key is a `CFNumber` of type `kCFNumberSInt32Type`; specifies an enumeration (`PMBorderType`).

`kPMLayoutNUpStr`
The layout string.

`kPMLayoutNUpKey`
The value of this key is a `CFBoolean` value; turns N-Up layout on.

`kPMLayoutRowsStr`
The layout rows string.

`kPMLayoutRowsKey`
The value of this key is a `CFNumber` of type `kCFNumberSInt32Type`; indicates number of layout rows.

`kPMLayoutColumnsStr`
The layout columns string.

`kPMLayoutColumnsKey`
The value of this key is a `CFNumber` of type `kCFNumberSInt32Type`; indicates number of layout columns.

`kPMLayoutDirectionStr`
The layout direction string.

`kPMLayoutDirectionKey`
The value of this key is a `CFNumber` of type `kCFNumberSInt32Type`; specifies an enumeration (`PMLayoutDirection`).

`kPMLayoutTileOrientationStr`
The layout tile orientation string.

`kPMLayoutTileOrientationKey`
The value of this key is a `CFNumber` of type `kCFNumberSInt32Type`; `PMOrientation`, 1 = portrait, 2 = landscape, etc.

`kPMQualityStr`
The print quality string.

`kPMQualityKey`
The value of this key is a `CFNumber` of type `kCFNumberSInt32Type`; specfies an enumeration—draft, normal, best.

`kPMPaperTypeStr`
The paper type string.

`kPMPaperTypeKey`
The value of this key is a `CFNumber` of type `kCFNumberSInt32Type`.

`kPMJobStateStr`
The job state string.

`kPMJobStateKey`
The value of this key is a `CFNumber` of type `kCFNumberSInt32Type`; specifies an enumeration—active = 0, pending, hold until, hold indefinitely, aborted, finished.

`kPMJobHoldUntilTimeStr`
The job hold until string.

`kPMJobHoldUntilTimeKey`
The value of this key is a `CFDate`; specifies time to print the job.

`kPMJobPriorityStr`
The job priority string.

`kPMJobPriorityKey`
The value of this key is a `CFNumber` of type `kCFNumberSInt32Type`; specifies an enumeration—Low = 0, normal, urgent.

`kPMPaperSourceStr`
The paper source string.

`kPMPaperSourceKey`
The value of this key is a `CFNumber` of type `kCFNumberSInt32Type`; specifies an enumeration of paper sources.

`kPMDuplexingStr`
The duplexing string.

`kPMDuplexingKey`
The value of this key is a `CFNumber` of type `kCFNumberSInt32Type`; specifies an enumeration—`kPMDuplexNone`, `kPMDuplexNoTumble`, `kPMDuplexTumble`, `kPMSimplexTumble`.

`kPMColorModeStr`
The color mode string.

`kPMColorModeKey`
The value of this key is a `CFNumber` of type `kCFNumberSInt32Type`; specifies an enumeration—Black and White, Grayscale, Color, HiFi Color.

`kPMColorSyncProfileIDStr`
The ColorSync Profile ID string.

`kPMColorSyncProfileIDKey`
The value of this key is a `CFNumber` of type `kCFNumberSInt32Type`; specifies the ID of the ColorSync profile to use.

`kPMColorSyncSystemProfilePathStr`
The ColorSync system profile path string.

`kPMColorSyncSystemProfilePathKey`
The value of this key is a `CFString`; specifies the path of system profile.

`kPMPrintScalingHorizontalStr`
The horizontal print scaling string.

`kPMPrintScalingHorizontalKey`
The value of this key is a `CFNumber` of type `kCFNumberDoubleType`; specifies the horizontal scaling factor applied to original page size.

`kPMPrintScalingVerticalStr`
The vertical print scaling string.

`kPMPrintScalingVerticalKey`
The value of this key is a `CFNumber` of type `kCFNumberDoubleType`; specifies the vertical scaling factor applied to original page size.

`kPMPrintScalingAlignmentStr`
The print scaling alignment string.

`kPMPrintScalingAlignmentKey`
The value of this key is a `CFNumber` of type `kCFNumberSInt32Type`; specifies an enumeration (`PMScalingAlignment`).

`kPMPrintOrientationStr`
The print orientation string.

`kPMPrintOrientationKey`
The value of this key is a `CFNumber` of type `kCFNumberSInt32Type`; specifies and enumeration—`PMOrientation`, 1 = portrait, 2 = landscape, etc.

`kPMPreviewStr`
The print preview string.

`kPMPreviewKey`
The value of this key is a `CFString`; "YES" indicates the user clicked the Preview button.

`kPMPrintBackupRecordHdlStr`
The print backup record handle string.

`kPMPrintBackupRecordHdlKey`
The value of this key is a print record handle (`CFData`); not used when data is flattened. (Mac OS 8 and 9 only)

`kPMPrintBackupRecordDataStr`
The print backup record data string.

`kPMPrintBackupRecordDataKey`
The value of this key is a print record (`CFData`) stored in complete form; used when flattening a ticket with the print record. (Mac OS 8 and 9 only)

`kPMPrintCustomDialogHdlStr`
The print custom dialog handle string.

`kPMPrintCustomDialogHdlKey`
The value of this key is a handle (`CFData`) to the print record using for custom dialog calls; not stored when data is flattened. (Mac OS 8 and 9 only)

`kPMPrimaryPaperFeedStr`
The primary paper feed string.

`kPMPrimaryPaperFeedKey`
The value of this key is a `CFArray`; the main and option PPD keywords for input paper feed.

`kPMSecondaryPaperFeedStr`
The secondary paper feed string.

`kPMSecondaryPaperFeedKey`
The value of this key is a `CFArray`; the main and option PPD keywords for input paper feed.

`kPMPSErrorHandlerStr`
The PostScript error handle string.

`kPMPSErrorHandlerKey`
The value of this key is a `CFNumber` of type `kCFNumberSInt32Type`.

`kPMPSErrorOnScreenStr`
The PostScript error on screen string.

`kPMPSErrorOnScreenKey`
The value of this key is a `CFBoolean`; turns PostScript error onscreen notification on.

`kPMPSTraySwitchStr`
The PostScript tray switch string.

`kPMPSTraySwitchKey`
The value of this key is a `CFArray`; the main and option PostScript printer description file keywords for tray switching.

`kPMPPDDictStr`
The PostScript printer description file string.

`kPMPPDDictKey`
The value of this key is a `CFDictionary`; the main and option PostScript printer description file keywords for additional features.

Printer Driver Creator Code Key

Defines a string and key for the printer driver creator code.

```
#define kPMDriverCreatorStr                    kPMPrinterInfoPrelude
"PMDriverCreator        "
#define kPMDriverCreatorKey                    CFSTR(kPMDriverCreatorStr)
```

Constants
`kPMDriverCreatorStr`
The printer driver creator string.

`kPMDriverCreatorKey`
The value of this key is a `CFNumber` of type `kCFNumberSInt32Type` that specifies the creator code for the printer driver.

Printer Font Keys

Defines a string and key for the printer fonts.

```
#define kPMPrinterFontStr kPMPrinterInfoPrelude "Printer
Fonts    "
#define kPMPrinterFontKey CFSTR(kPMPrinterFontStr)
```

Constants
`kPMPrinterFontStr`
The printer font string.

`kPMPrinterFontKey`
The value of this key is `CFData` that specifies the printer resident fonts.

Printer Info Ticket Keys

Defines strings and keys for items in a printer info ticket.

```
#define kPMPrinterInfoPrelude "com.apple.print.PrinterInfo."
#define kPMPrinterLongNameStr kPMPrinterInfoPrelude "PMPrinterLongName"
#define kPMPrinterLongNameKey CFSTR(kPMPrinterLongNameStr)
#define kPMPrinterShortNameStr kPMPrinterInfoPrelude "PMPrinterShortName"
#define kPMPrinterShortNameKey CFSTR(kPMPrinterShortNameStr)
#define kPMMakeAndModelNameStr kPMPrinterInfoPrelude "PMMakeAndModelName"
#define kPMMakeAndModelNameKey CFSTR(kPMMakeAndModelNameStr)
#define kPMPrinterAddressStr kPMPrinterInfoPrelude "PMPrinterAddress"
#define kPMPrinterAddressKey CFSTR(kPMPrinterAddressStr)
#define kPMSupportsColorStr kPMPrinterInfoPrelude "PMSupportsColor"
#define kPMSupportsColorKey CFSTR(kPMSupportsColorStr)
#define kPMDoesCopiesStr kPMPrinterInfoPrelude "PMDoesCopies"
#define kPMDoesCopiesKey CFSTR(kPMDoesCopiesStr)
#define kPMDoesCopyCollateStr kPMPrinterInfoPrelude "PMDoesCopyCollate"
#define kPMDoesCopyCollateKey CFSTR(kPMDoesCopyCollateStr)
#define kPMDoesReverseOrderStr kPMPrinterInfoPrelude "PMDoesReverseOrderK"
#define kPMDoesReverseOrderKey CFSTR(kPMDoesReverseOrderStr)
#define kPMInputFileTypeListStr kPMPrinterInfoPrelude "PMInputFileTypeList"
#define kPMInputFileTypeListKey CFSTR(kPMInputFileTypeListStr)
#define kPMOutputTypeListStr     kPMPrinterInfoPrelude "PMOutputTypeList"
#define kPMOutputTypeListKey     CFSTR(kPMOutputTypeListStr)
```

Constants

`kPMPrinterInfoPrelude`
The printer info ticket prelude string.

`kPMPrinterLongNameStr`
The printer long name string.

`kPMPrinterLongNameKey`
The value of this key is a `CFString`; specifies the full name of the printer.

`kPMPrinterShortNameStr`
The printer short name string.

`kPMPrinterShortNameKey`
The value of this key is a `CFString`; specifies a shortened version of the printer name.

`kPMMakeAndModelNameStr`
The printer make and model string.

`kPMMakeAndModelNameKey`
The value of this key is a `CFString`; specifies the product name used for the printer

`kPMPrinterAddressStr`
The printer address string.

`kPMPrinterAddressKey`
The value of this key is the product address (`CFData`).

`kPMSupportsColorStr`
The color supported string.

`kPMSupportsColorKey`
The value of this key is a `CFBoolean`; `true` if the printer supports color printing.

`kPMDoesCopiesStr`
The copies string.

`kPMDoesCopiesKey`
The value of this key is a `CFBoolean`; `true` if the printer supports copies.

`kPMDoesCopyCollateStr`
The collation string.

`kPMDoesCopyCollateKey`
The value of this key is a `CFBoolean`; `true` if the printer supports collation.

`kPMDoesReverseOrderStr`
The reverse-order string.

`kPMDoesReverseOrderKey`
The value of this key is a `CFBoolean`; `true` if the printer can print in reverse order.

`kPMInputFileTypeListStr`
The file type list string.

`kPMInputFileTypeListKey`
The value of this key is a `CFArray` of `CFString` data that indicates file types.

`kPMOutputTypeListStr`
The output type list string.

`kPMOutputTypeListKey`
The value of this key is a `CFArray` of `CFString` data that indicates the MIME types for the data that can be sent to an I/O module.

Printing Phase Types

Specify the phase of the printing process.

```
typedef UInt16 PMPrintingPhaseType;
enum {
    kPMPhaseUnknown = 0,
    kPMPhasePreDialog = 1,
    kPMPhaseDialogsUp = 2,
    kPMPhasePostDialogs = 3,
    kPMPhasePreAppDrawing = 4,
    kPMPhaseAppDrawing = 5,
    kPMPhasePostAppDrawing = 6,
    kPMPhasePreConversion = 7,
    kPMPhaseConverting = 8,
    kPMPhasePostConversion = 9,
    kPMPhasePrinting = 10
};
```

Constants

`kPMPhaseUnknown`
Phase unknown.

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

Declared in `PMTicket.h`.

`kPMPhasePreDialog`
Just before the code to open the dialog.

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

Declared in `PMTicket.h`.

`kPMPhaseDialogsUp`
A printing dialogs is open.

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

Declared in `PMTicket.h`.

`kPMPhasePostDialogs`
The printing dialogs have been opened and are now closed, but the job is not yet spooling.
Available in Mac OS X v10.0 through Mac OS X v10.4.
Declared in `PMTicket.h`.

`kPMPhasePreAppDrawing`
The job is just about to spool.
Available in Mac OS X v10.0 through Mac OS X v10.4.
Declared in `PMTicket.h`.

`kPMPhaseAppDrawing`
Drawing commands are now being spooled from the application.
Available in Mac OS X v10.0 through Mac OS X v10.4.
Declared in `PMTicket.h`.

`kPMPhasePostAppDrawing`
Spooling is finished, not the job is not yet rendered or converted.
Available in Mac OS X v10.0 through Mac OS X v10.4.
Declared in `PMTicket.h`.

`kPMPhasePreConversion`
The job is just about to be converted to its final format (PostScript, Raster, or other).
Available in Mac OS X v10.0 through Mac OS X v10.4.
Declared in `PMTicket.h`.

`kPMPhaseConverting`
The job is being converted from the spool file to the final format.
Available in Mac OS X v10.0 through Mac OS X v10.4.
Declared in `PMTicket.h`.

`kPMPhasePostConversion`
The data is ready for the printer and waiting for completion.
Available in Mac OS X v10.0 through Mac OS X v10.4.
Declared in `PMTicket.h`.

`kPMPhasePrinting`
The job is waiting for the printer.
Available in Mac OS X v10.0 through Mac OS X v10.4.
Declared in `PMTicket.h`.

Discussion
Printing phase types signal a shift from one printing phase to the next and are set by many parts of the printing system. You can check the phase by testing for greater-than and less-than conditions.

Rasterizer Options

Specify options for rasterizing.

```
enum {
    kPMPSTTRasterizerUnknown = 0,
    kPMPSTTRasterizerNone = 1,
    kPMPSTTRasterizerAccept68K = 2,
    kPMPSTTRasterizerType42 = 3
};
```

Constants

`kPMPSTTRasterizerUnknown`
Rasterizer unknown.
Available in Mac OS X v10.0 and later.
Declared in `PMTicket.h`.

`kPMPSTTRasterizerNone`
No rasterizer.
Available in Mac OS X v10.0 and later.
Declared in `PMTicket.h`.

`kPMPSTTRasterizerAccept68K`
Accepts 68 K.
Available in Mac OS X v10.0 and later.
Declared in `PMTicket.h`.

`kPMPSTTRasterizerType42`
Uses type 42.
Available in Mac OS X v10.0 and later.
Declared in `PMTicket.h`.

Template Entry Data Types

Specify the data type of a job template entry.

```
typedef SInt32 PMValueType;
enum {
    kPMValueUndefined = 0,
    kPMValueBoolean = 1,
    kPMValueData = 2,
    kPMValueString = 3,
    kPMValueSInt32 = 4,
    kPMValueSInt32Range = 5,
    kPMValueUInt32 = 6,
    kPMValueUInt32Range = 7,
    kPMValueDouble = 8,
    kPMValueDoubleRange = 9,
    kPMValuePMRect = 10,
    kPMValueDate = 11,
    kPMValueArray = 12,
    kPMValueDict = 13,
    kPMValueTicket = 14
};
```

Constants

`kPMValueUndefined`
: Template entry is unknown or undefined.

  Available in Mac OS X v10.0 and later.

  Declared in `PMTemplate.h`.

`kPMValueBoolean`
: A `CFBoolean` value.

  Available in Mac OS X v10.0 and later.

  Declared in `PMTemplate.h`.

`kPMValueData`
: A `CFData` value. This is generic data converted to Core Foundation data. Template entries of this type should have a default value, but no other constraints.

  Available in Mac OS X v10.0 and later.

  Declared in `PMTemplate.h`.

`kPMValueString`
: A `CFString` value. Template entries of this type should have a default value, but no other constraints.

  Available in Mac OS X v10.0 and later.

  Declared in `PMTemplate.h`.

`kPMValueSInt32`
: A `CFNumber` value.

  Available in Mac OS X v10.0 and later.

  Declared in `PMTemplate.h`.

`kPMValueSInt32Range`
: A pair of `CFNumber` values (`SInt32` data type).

  Available in Mac OS X v10.0 and later.

  Declared in `PMTemplate.h`.

`kPMValueUInt32`
A `CFNumber` value (unsigned long data types).

Available in Mac OS X v10.0 and later.

Declared in `PMTemplate.h`.

`kPMValueUInt32Range`
A pair of `CFNumber` values (`UInt32` values).

Available in Mac OS X v10.0 and later.

Declared in `PMTemplate.h`.

`kPMValueDouble`
A `CFNumber` value (`double` data type.)

Available in Mac OS X v10.0 and later.

Declared in `PMTemplate.h`.

`kPMValueDoubleRange`
A pair of `CFNumber` values (`double` data types).

Available in Mac OS X v10.0 and later.

Declared in `PMTemplate.h`.

`kPMValuePMRect`
A Core Foundation array that contains four `CFNumbers` values (`double` data types).

Available in Mac OS X v10.0 and later.

Declared in `PMTemplate.h`.

`kPMValueDate`
A `CFDate` value.

Available in Mac OS X v10.0 and later.

Declared in `PMTemplate.h`.

`kPMValueArray`
A Core Foundation array.

Available in Mac OS X v10.0 and later.

Declared in `PMTemplate.h`.

`kPMValueDict`
A `CFDictionary` data structure. Template entries of this type should have a default value, but no other constraints.

Available in Mac OS X v10.0 and later.

Declared in `PMTemplate.h`.

`kPMValueTicket`
A ticket.

Available in Mac OS X v10.0 and later.

Declared in `PMTemplate.h`.

Discussion
A template entry data type determines what other fields and functions are available for the entry.

Template Strings

Define a string associated with a template.

```
#define kPMTemplatePrelude     "com.apple.print.TemplateSpecific."
#define kPMPaperInfoListStr     "PMTemplatePaperInfoTicket"
#define kPMPaperInfoList CFSTR( kPMPaperInfoListStr )
```

Constants

`kPMTemplatePrelude`
A template prelude string.

`kPMPaperInfoListStr`
A paper info list string.

`kPMPaperInfoList`
A paper info list.

Ticket Levels

Specify the level of an item within a ticket.

```
enum {
    kPMTopLevel = 0
};
```

Constants

`kPMTopLevel`
Specifies the top level ticket.

Discussion

You should pass the constant `kPMTopLevel` to any function that has parameters to specify ticket level calls. No other options are currently available.

Ticket Types

Specify a ticket type, that is, a ticket kind.

```
typedef SInt16 PMTicketType;
enum {
    kPMTicketTypeUnknown = -1,
    kPMJobTicketType = 1,
    kPMDocumentTicketType = 2,
    kPMPageTicketType = 3,
    kPMPageFormatTicketType = 4,
    kPMPrintSettingsTicketType = 5,
    kPMPrinterInfoTicketType = 6,
    kPMDestinationTicketType = 7,
    kPMConverterSetupTicketType = 8,
    kPMModuleInfoTicketType = 9,
    kPMTicketListType = 10,
    kPMPaperInfoTicketType = 11
};
```

Constants

`kPMTicketTypeUnknown`
Specifies the ticket type is not know.

Available in Mac OS X v10.0 and later.

Declared in `PMTicket.h`.

`kPMJobTicketType`
Specifies a job ticket.
Available in Mac OS X v10.0 and later.
Declared in `PMTicket.h`.

`kPMDocumentTicketType`
Specifies a document ticket.
Available in Mac OS X v10.0 and later.
Declared in `PMTicket.h`.

`kPMPageTicketType`
Specifies a page ticket.
Available in Mac OS X v10.0 and later.
Declared in `PMTicket.h`.

`kPMPageFormatTicketType`
Specifies a page format ticket.
Available in Mac OS X v10.0 and later.
Declared in `PMTicket.h`.

`kPMPrintSettingsTicketType`
Specifies a print settings ticket.
Available in Mac OS X v10.0 and later.
Declared in `PMTicket.h`.

`kPMPrinterInfoTicketType`
Specifies a printer info ticket.
Available in Mac OS X v10.0 and later.
Declared in `PMTicket.h`.

`kPMDestinationTicketType`
Specifies a destination ticket.
Available in Mac OS X v10.0 and later.
Declared in `PMTicket.h`.

`kPMConverterSetupTicketType`
Specifies a converter setup ticket.
Available in Mac OS X v10.0 and later.
Declared in `PMTicket.h`.

`kPMModuleInfoTicketType`
Specifies a printer module info ticket.
Available in Mac OS X v10.0 and later.
Declared in `PMTicket.h`.

`kPMTicketListType`
Specifies a list ticket.
Available in Mac OS X v10.0 and later.
Declared in `PMTicket.h`.

`kPMPaperInfoTicketType`
Specifies a paper info ticket.

Available in Mac OS X v10.0 and later.

Declared in `PMTicket.h`.

Ticket Type Strings

Specify ticket types.

```
#define kPMJobTicket CFSTR("com.apple.print.JobTicket")
#define kPMDocumentTicket CFSTR("com.apple.print.DocumentTicket")
#define kPMPageTicket CFSTR("com.apple.print.PageTicket")
#define kPMPageFormatTicket CFSTR("com.apple.print.PageFormatTicket")
#define kPMPrintSettingsTicket CFSTR("com.apple.print.PrintSettingsTicket")
#define kPMDestinationTicket CFSTR("com.apple.print.DestinationTicket")
#define kPMConverterSetupTicket CFSTR("com.apple.print.ConverterSetupTicket")
#define kPMPrinterInfoTicket CFSTR("com.apple.print.PrinterInfoTicket")
#define kPMModuleInfoTicket CFSTR("com.apple.print.ModuleInfoTicket")
#define kPMTicketList CFSTR("com.apple.print.TicketList")
#define kPMPaperInfoTicket CFSTR("com.apple.print.PaperInfoTicket")
```

Constants

`kPMJobTicket`
Specifies a job ticket; the top-level ticket for a print job.

`kPMDocumentTicket`
Specifies a document ticket.

`kPMPageTicket`
Specifies a page ticket.

`kPMPageFormatTicket`
Specifies a page format ticket.

`kPMPrintSettingsTicket`
Specifies a print settings ticket.

`kPMDestinationTicket`
Specifies a destination ticket.

`kPMConverterSetupTicket`
Specifies a converter setup ticket.

`kPMPrinterInfoTicket`
Specifies a printer info ticket.

`kPMModuleInfoTicket`
Specifies a module info ticket.

`kPMTicketList`
Specifies a list ticket.

`kPMPaperInfoTicket`
Specifies a paper info ticket.

Discussion

Use these ticket type strings of ticket types to create a ticket.

Result Codes

This table lists result codes defined for Ticket Services.

Result Code	Value	Description
kPMTicketTypeNotFound	-9580	The ticket type was not found. Available in Mac OS X v10.0 and later.
kPMUpdateTicketFailed	-9581	Could not update the ticket. Available in Mac OS X v10.0 and later.
kPMValidateTicketFailed	-9582	Could not validate the ticket. Available in Mac OS X v10.0 and later.
kPMSubTicketNotFound	-9583	Could not find the subticket. Available in Mac OS X v10.0 and later.
kPMInvalidSubTicket	-9584	The subticket is invalid. Available in Mac OS X v10.0 and later.
kPMDeleteSubTicketFailed	-9585	Could not delete the subticket. Available in Mac OS X v10.0 and later.
kPMItemIsLocked	-9586	The item is locked. Available in Mac OS X v10.0 and later.
kPMTicketIsLocked	-9587	The ticket is locked. Available in Mac OS X v10.0 and later.
kPMTemplateIsLocked	-9588	The job template is locked. Available in Mac OS X v10.0 and later.
kPMKeyNotFound	-9589	Could not find the key. Available in Mac OS X v10.0 and later.
kPMKeyNotUnique	-9590	The key is not unique. Available in Mac OS X v10.0 and later.
kPMUnknownDataType	-9591	The data type is unknown. Available in Mac OS X v10.0 and later.

Document Revision History

This table describes the changes to *Ticket Services Reference*.

Date	Notes
2002-10-23	Corrected the description of the `dict` parameter in the `PMTicketGetPPDDict` (page 67) function.
	Updated formatting.
2001-04-01	First release of this document.

Document Revision History

Index

L

M

P

R

S

T