

## wxPrintout

このクラスは、アプリケーションドキュメントを印刷するための機能を隠蔽している。このクラス派生して新しいクラスを作成し、OnPrintPage や HasPage などの呼び出しに応じるために、メンバをオーバーライドしなければならない。このクラスのインスタンスは、印刷やプレビューを初期化するために、wxPrinter::Print や [wxPrintPreview](#) オブジェクトに渡される。

plugin::pdf::PDFParser=HASH(0x53e16c) 基本クラス

wxObject

plugin::pdf::PDFParser=HASH(0x53e16c) インクルードファイル

<wx/print.h>

plugin::pdf::PDFParser=HASH(0x53e16c) 参考

[印刷の概要](#) , [wxPrinterDC](#), [wxPrintDialog](#), [wxPrinter](#), [wxPrintPreview](#)

plugin::pdf::PDFParser=HASH(0x53e16c) メンバ

[wxPrintout::wxPrintout](#)

[wxPrintout::~wxPrintout](#)

[wxPrintout::GetDC](#)

[wxPrintout::GetPageInfo](#)

[wxPrintout::GetPageSizeMM](#)

[wxPrintout::GetPageSizePixels](#)

[wxPrintout::GetPPIPrinter](#)

[wxPrintout::GetPPIScreen](#)

[wxPrintout::GetTitle](#)

[wxPrintout::HasPage](#)

[wxPrintout::IsPreview](#)

[wxPrintout::OnBeginDocument](#)

[wxPrintout::OnEndDocument](#)

[wxPrintout::OnBeginPrinting](#)

[wxPrintout::OnEndPrinting](#)

[wxPrintout::OnPreparePrinting](#)

[wxPrintout::OnPrintPage](#)

wxPrintout::wxPrintout

[wxPrintout](#)(const wxString& title = "Printout")

コンストラクタ。オプションで title 引数を渡す (現在のファイル名を渡すのは良いアイデアで

ある)。これは印刷リスト(少なくとも MSW では)に現れる。

**wxPrintout::~~wxPrintout**

~wxPrintout()

デストラクタ。

**wxPrintout::~GetDC**

wxDC \* GetDC()

印刷に関連付けられた(印刷やプレビューの始めに与えられた)デバイスコンテキストを返す。これは、Windows 環境下での印刷であれば wxPrinterDC であり、他のプラットフォームであれば wxPostScriptDC であり、プレビューであれば wxMemoryDC である。

**wxPrintout::~GetPageInfo**

void GetPageInfo(int \*minPage, int \*maxPage, int \*pageFrom, int \*pageTo)

ユーザが指定可能な最小ページと最大ページの値と、印刷するように要求されたページの幅に関する情報を得るために、アプリケーションからフレームワークによって呼ばれる。デフォルトでは、ページの最小値と最大値として 1 と 32000 を、要求されたページ幅として 1 と 1 を返す。

minPage がゼロの場合、ページ番号コントロールは印刷ダイアログ内で無効化される。

plugin::pdf::PDFParser=HASH(0x53e16c)wxPython note: When this method is implemented in a derived Python class, it should be designed to take no parameters (other than the self reference) and to return a tuple of four integers.

plugin::pdf::PDFParser=HASH(0x53e16c)wxPerl note: When this method is overridden in a derived class, it must not take any parameters, and returns a 4-element list.

**wxPrintout::~GetPageSizeMM**

void GetPageSizeMM(int \*w, int \*h)

印刷ページのサイズをミリメートル単位で返す。

plugin::pdf::PDFParser=HASH(0x53e16c)wxPython note: This method returns the output-only parameters as a tuple.

plugin::pdf::PDFParser=HASH(0x53e16c)wxPerl note: In wxPerl this method takes no arguments and returns a 2-element list ( w, h )

**wxPrintout::~GetPageSizePixels**

```
void GetPageSizePixels(int *w, int *h)
```

印刷ページのサイズをピクセル単位で返す。印刷出力がプレビューに対して使用される場合には、wxDC::GetSize が返す値と異なる場合がある。なぜなら、この場合、メモリデバイスコンテキストが使用され、現在のプレビューズームを繁殖したビットマップサイズが使用されるためである。プレビューがサポートされる場合には、アプリケーションは、この不一致を考慮しなければならない。

plugin::pdf::PDFParser=HASH(0x53e16c)wxPython note: This method returns the output-only parameters as a tuple.

plugin::pdf::PDFParser=HASH(0x53e16c)wxPerl note: In wxPerl this method takes no arguments and returns a 2-element list ( w, h )

### **wxPrintout::GetPPIPrinter**

```
void GetPPIPrinter(int *w, int *h)
```

プリンタデバイスコンテキストの、論理インチあたりのピクセル (ppi) を返す。プリンタ PPI をスクリーン PPI で割ると、プリンタ上に文字を表示するための適切なスケール係数が求められる。プレビュー DC のサイズを考慮するために、この値にスケール係数を掛けることを忘れてはならない。

plugin::pdf::PDFParser=HASH(0x53e16c)wxPython note: This method returns the output-only parameters as a tuple.

plugin::pdf::PDFParser=HASH(0x53e16c)wxPerl note: In wxPerl this method takes no arguments and returns a 2-element list ( w, h )

### **wxPrintout::GetPPIScreen**

```
void GetPPIScreen(int *w, int *h)
```

スクリーンデバイスコンテキストの、論理インチあたりのピクセル (ppi) を返す。プリンタ PPI をスクリーン PPI で割ると、プリンタ上に文字を表示するための適切なスケール係数が求められる。プレビュー DC のサイズを考慮するために、この値にスケール係数を掛けることを忘れてはならない。

### **wxPrintout::GetTitle**

```
wxString GetTitle()
```

印刷出力のタイトルを返す。

plugin::pdf::PDFParser=HASH(0x53e16c)wxPython note: This method returns the output-only parameters as a tuple.

plugin::pdf::PDFParser=HASH(0x53e16c)wxPerl note: In wxPerl this method takes no arguments and returns a 2-element list ( w, h )

### wxPrintout::HasPage

bool HasPage(int pageNum)

ドキュメントがページを持っている場合には、true を返すように、そうでなければ、false を返すようにオーバーライドしなければならない。false を返すのは、ドキュメントの終わりを意味する。デフォルトでは、HasPage は、ドキュメントが1ページだけ持っているかのように振舞う。

### wxPrintout::IsPreview

bool IsPreview()

現在の印刷出力がプレビューのために使用されている場合には、true を返す。

### wxPrintout::OnBeginDocument

bool OnBeginDocument(int startPage, int endPage)

ドキュメントの印刷を始めるときにフレームワークから呼ばれる。印刷ジョブをキャンセルすると、この関数は false を返す。OnBeginDocument は、印刷される部数ごと、1回ずつ呼ばれる。

基本クラスの wxPrintout::OnBeginDocument は、オーバーライドされた関数の中から呼ばれなくてはならない(そして、戻り値がチェックされなければならない)。なぜなら、wxPrintout::OnBeginDocument は、wxDC::StartDoc を呼ぶからである。

plugin::pdf::PDFParser=HASH(0x53e16c)wxPython note: If this method is overridden in a Python class then the base class version can be called by using the method base\_OnBeginDocument(startPage, endPage).

### wxPrintout::OnEndDocument

void OnEndDocument()

ドキュメントを印刷する最後に、フレームワークから呼ばれる。OnEndDocument は、印刷される部数ごとに、1回ずつ呼ばれる。

基本クラスの wxPrintout::OnEndDocument は、オーバーライドされた関数の中から呼ばれなくてはならない。なぜなら、wxPrintout::OnEndDocument は、wxDC::EndDoc を呼ぶからである。

### wxPrintout::OnBeginPrinting

void OnBeginPrinting()

印刷を開始するときに、フレームワークから呼ばれる。OnBeginPrinting は、(何部印刷されるかに関わらず) 印刷ジョブごとに 1 回ずつ呼ばれる。

**wxPrintout::OnEndPrinting**

void OnEndPrinting()

印刷を終了するときに、フレームワークから呼ばれる。OnEndPrinting は、(何部印刷されるに関わらず) 印刷ジョブごとに 1 回ずつ呼ばれる。

**wxPrintout::OnPreparePrinting**

void OnPreparePrinting()

wxPrintout オブジェクトで発生するどの要求よりも前に、フレームワークから 1 回だけ呼ばれる。オブジェクトに対して、例えば、ドキュメント内のページ数を計算するための機会を与える。

**wxPrintout::OnPrintPage**

bool OnPrintPage(int pageNum)

1 つのページが印刷されるときに、フレームワークから呼ばれる。印刷ジョブをキャンセルすると、false を返す。アプリケーションは、利用するデバイスコンテキストを得るために、wxPrintout::GetDC を使用することができる。