



PureBasic CDPrint Module

The CDPrint Module

The CDPrint module is a module designed to help with printing from a Purebasic programme.

PureBasic CDPrint Module

Printed: Febtuary 2017

Special thanks to:

infratec for reprogramming almost all the original code to get to where this is today. And to all the contributors on the PB forum for putting up with what now seem to be rather dumb questions at times over the last year.

Publisher

collectordave

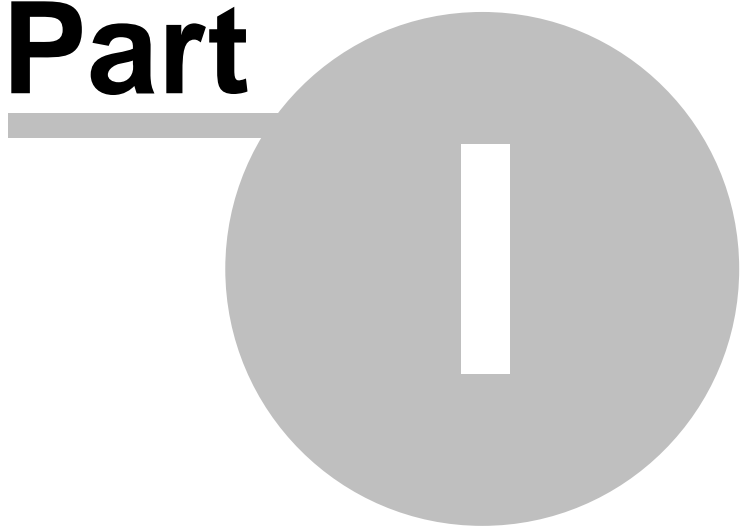
Technical Editor

...enter name...

Table of Contents

Foreword	0
Part I Introduction	3
1 Overview	3
2 Getting Started	3
Part II Printing Commands	8
1 Open Print Job	8
2 Add Page	9
3 Print Line	9
4 Print Box	10
5 Print Image	10
6 Print Image From File	10
7 Print Canvas	11
8 Print Text	11
Index	0

Part



1 Introduction

1.1 Overview

Adding printing capabilities to your programme can be one the most difficult aspects of writing an application. You may have a really beautiful painting application, or a business invoicing application that uses some of the most advanced techniques ever thought of, but it all starts to fall flat when you need to print the output from your application.

You will find almost from the very start that you cannot control which printer your user will want to use or indeed any other aspects of printing from your application, such as paper size and orientation, these can also change from one minute to the next. The worst is when you set up all your print routines to cope with paper size A4 in portrait mode and the user chooses Letter sized paper in landscape on their printer. This is the first choice you have to make, do you go merrily ahead and ignore the users choice and tell them that they must choose A4 in portrait mode or they will get strange results or do you attempt to cope with the users choice.

The next choice surrounds the data available in your application. Most will print clearly in portrait mode but some portions will need to be printed in landscape, faced with this scenario you can attempt to control the users printer, forcing it to print in landscape when required where you will need low level control of the printer (hopefully your user has a printer that can be controlled) or warn the user that he is attempting to print a landscape page in portrait mode.

Further considerations are how is the data presented in your application, have you provided a full WYSIWYG application or is the data placed on the screen to be assembled later for printing. With a full WYSIWYG application you will need to control the printing and page size selection as well as ensuring that the printed page matches the displayed page.

After sorting through all those thoughts you will need to consider that your user may not want to print all the pages of data from your application but may want to choose which pages to print. To achieve this you need to programme a preview feature to show your user what will be printed on each page and then allow your user to choose the pages to print.

The CDPrint module is designed to help you with all the above. There are however some things you will need to know. First, and most important, you will need a copy of PureBasic installed on your computer, and be familiar with it's operation. The module has been tested with PureBasic Version 5.6 Beta 3, the latest version at this time. If new to programming or just new to PB visit the forum, introduce yourself and all on the forum will help as much as they can to get your application up and running. the second thing is you need a printer. Some PDF printers are available which need no paper but produce PDF documents from your applications printer output.

This manual assumes you are familiar with PB and makes no attempt to explain the keywords or other aspects of PB as these are all explained in detail in the help files and documentation available with PB and of course all users on the forum are there to help when required.

When ready move on to the getting started chapter of this manual where the operation of the module is explained. A small test programme is provided for you to type in and test.

1.2 Getting Started

In this chapter we will concentrate on getting those first very important pages out of that printer. To do this start by creating a directory "CDPrint test" and copy the CDPrint.pbi module into that directory. Then simply copy the example programme into a new window in the PureBasic IDE and save as "CD Test Prog 001.pb" then run. The programme simply asks the user to choose a printer and paper size and then shows a preview of two lines on the paper which cross 20mm from the top edge of the paper

chosen. See the illustrations after the programme listing to see what your printer should print. Remember here that in portrait the top edge is the shorter side and in landscape the top edge is the longer side.

Programme 1. CD Test Prog 001.pb

```
IncludeFile "CDPrint.pbi"

Enumeration WinMain
#WinMain
#btnPrint
EndEnumeration

Define Event.i
Define Orientation.i

OpenWindow(#WinMain, 5, 5, 600, 400, "CDPrint Test Programme", #PB_Window_SystemMenu)
ButtonGadget(#btnPrint, 130, 230, 110, 20, "Print")

Repeat

    Event = WaitWindowEvent()

    Select Event

        Case #PB_Event_Gadget

            Select EventGadget()

                Case #btnPrint

                    If CDPrint::Open("Test Print",CDPrint::#Preview)
                        If CDPrint::Printer\Height > CDPrint::Printer\Width
                            Orientation = CDPrint::#Portrait
                        Else
                            Orientation = CDPrint::#Landscape
                        EndIf

                        CDPrint::AddPage(Orientation)
                        CDPrint::PrintLine(20,20,CDPrint::Printer\Width - 20,20,1)
                        CDPrint::PrintLine(CDPrint::Printer\Width/2,10,CDPrint::Printer\Width/2,CDPrint::Printer\Height - 20,1)
                        CDPrint::Finished()
                    EndIf

                EndSelect

            EndSelect

    Until Event = #PB_Event_CloseWindow
```

Programme End

If you chose Portrait orientation on your printer you should see something similar to the next illustration in the preview window.

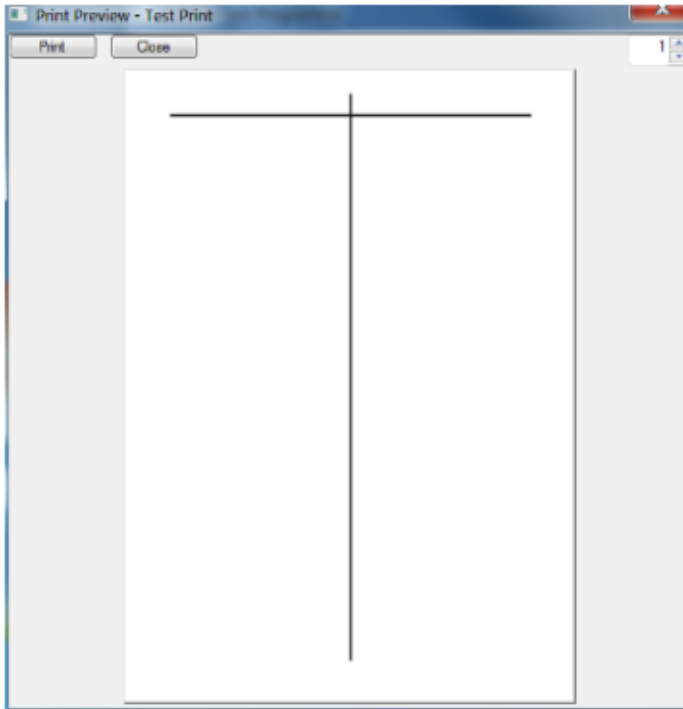


Figure 1. Portrait Output From CD Print

At this stage do not worry if you do not understand what is happening in the programme the PureBasic documentation is a great source to start in understanding the programme and all the good folks on the forum will explain in even greater detail if needed. All the parts of the programme dealing with printing are explained below.

The first line of the programme tells PureBasic to include all the functions in the file CDPrint.pbi into the programme nothing more is need to gain access to all the printing methods.

```
IncludeFile "CDPrint.pbi"
```

The next significant line for printing is:-

```
If CDPrint::Open("Test Print",CDPrint::#Preview)
```

This tells the print module to open a new print job called "Test Print" and to show a preview of all that is printed. More about preview and no preview later.

The next part is a few lines of code:-

```
If CDPrint::PrinterHeight > CDPrint::PrinterWidth  
  Orientation = CDPrint::#Portrait  
Else  
  Orientation = CDPrint::#Landscape  
EndIf
```

These lines sort out the orientation the user has selected and allows the module to print on the paper in the user selected orientation. The CDPrint module can override the user selected orientation, more about that in the chapter on orientation.

The next few lines of code :-


```
CDPrint::AddPage(Orientation)
CDPrint::PrintLine(20,20,CDPrint::Printer\Width - 20,20,1)
CDPrint::PrintLine(CDPrint::Printer\Width/2, 10,CDPrint::Printer\Width/2,CDPrint::Printer\Height - 20,1)
```

Tells the module to open the first page of the print job and then prints two lines.

Finally:-

```
CDPrint::Finished()
```

Tells the module that you have finished printing and dependant upon the mode to display the preview or not.

Now you have the preview window shown the user can carry out various actions.

The Print Button

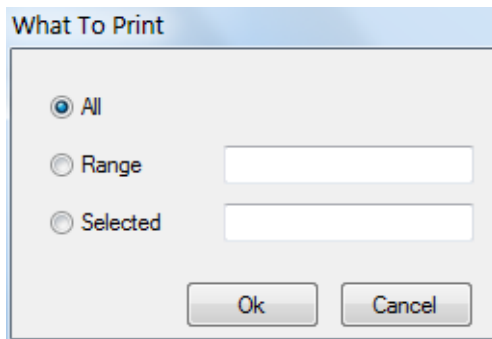


Figure 2. What To Print Dialogue

Clicking the Print button will display the What To Print dialog. Here we only have one page so selecting all and then clicking Ok will simply print this page. When the document contains more than one page a range of pages can be selected or individual pages can be selected for printing.

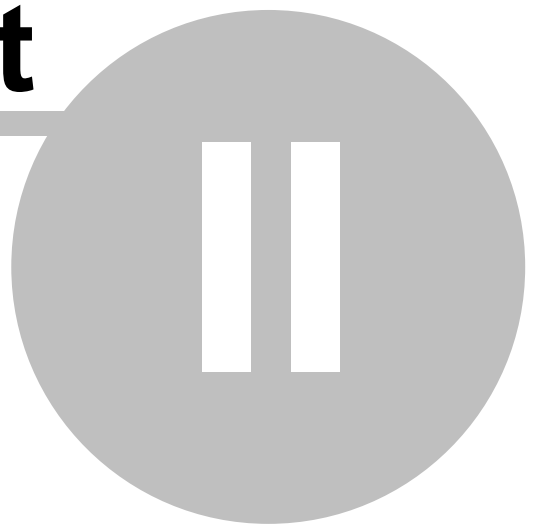
The Close Button

Clicking the Close button will close the preview screen and the print job will be terminated.

The Page Selection Spin Gadget

Selecting page numbers with the page selection spin gadget on the right will display the selected page in the preview window. There is only one page in this example so nothing will happen.

Part



2 Printing Commands

2.1 Open Print Job

CDPrint::Open()

Syntax

CDPrint::Open(JobName.s, Mode.i = #Preview)

Description

Opens a new print job and sets up the print environment by displaying the PrintRequester and waiting for the user to select a printer and paper. After opening a print job the printer parameters are available in the CDPrint::Printer structure.

Parameters

JobName.s The title of this print job. This will be displayed in the printers print queue when actual printing starts.

Mode.i The operating mode of the print module. Valid values are CDPrint::#Preview which will display a print preview window after all application printing is finished or CDPrint::NoPreview which does not display a preview window but immediately opens the Print What dialog after all printing is finished.

Return value

None.

Example

```
IncludeFile "CDPrint.pbi"
```

```
Enumeration WinMain
  #WinMain
  #btnPrint
EndEnumeration
```

```
Define Event.i
Define Orientation.i
```

```
OpenWindow(#WinMain, 5, 5, 600, 400, "CDPrint Test Programme", #PB_Window_SystemMenu)
ButtonGadget(#btnPrint, 130, 230, 110, 20, "Print")
```

```
Repeat
```

```
  Event = WaitWindowEvent()
```

```
  Select Event
```

```
    Case #PB_Event_Gadget
```

```
      Select EventGadget()
```

```
        Case #btnPrint
```

```
          If CDPrint::Open("Test Print",CDPrint::#Preview)
            If CDPrint::Printer\Height > CDPrint::Printer\Width
              Orientation = CDPrint::#Portrait
            Else
              Orientation = CDPrint::#Landscape
            EndIf
```

```
CDPrint::AddPage(Orientation)
CDPrint::PrintLine(20,20,CDPrint::Printer\Width - 20,20,1)
CDPrint::PrintLine(CDPrint::Printer\Width/2,10,CDPrint::Printer\Width/2,CDPrint::Printer\Height - 20,1)
CDPrint::Finished()
EndIf

EndSelect

EndSelect

Until Event = #PB_Event_CloseWindowEvent = #PB_Event_CloseWindow
```

2.2 Add Page

CDPrint::AddPage()

Syntax

CDPrint::AddPage(Orientation.i)

Description

Adds a new page to the current print job. This is called at least once after opening a new print job with CDPrint::Open() and as many times as needed when printing.

Parameters

Orientation.i Sets the orientation for this page. This can be the same as the user selected orientation or can be overridden by the print module. Valid values are CDPrint::Portrait or CDPrint::Landscape.

Return value

None.

2.3 Print Line

CDPrint::PrintLine()

Syntax

CDPrint::PrintLine(X1.i,Y1.i,X2.i,Y2.i,Width.f,Color.i=\$FF000000)

Description

Prints a straight line at the given co-ordinates, at the given width and colour.

Parameters

X1,Y1 Co-ordinates of the start point of the line
X2,Y2 Co-ordinates of the endpoint of the line
Width the width of the line in mm
Color The RGBA colour of the line.

Return value

None.

2.4 Print Box

CDPrint::PrintBox()

Syntax

PrintBox(X1.i,Y1.i,X2.i,Y2.i,Width.f,Color.i=\$FF000000)

Description

Prints an open box at the given co-ordinates, using the given width and colour.

Parameters

X1,Y1 Co-ordinates of the top and left corner of the box
X2,Y2 Co-ordinates of the bottom right corner of the box
Width the width of the lines used to draw the box in mm
Color The RGBA colour of the box.

Return value

None.

2.5 Print Image

CDPrint::PrintImage()

Syntax

CDPrint::PrintImage(Image.i,X1.i,Y1.i,X2.i,Y2.i,Transparency.i=255)

Description

Prints an image at the given co-ordinates and at the size defined by the co-ordinates, using the given transparency.

Parameters

Image The handle of the image to be printed
X1,Y1 Co-ordinates of the top and left corner of the image
X2,Y2 Co-ordinates of the bottom right corner of the image in mm
Transparency The transparency to be applied when printing the image

Return value

None.

2.6 Print Image From File

CDPrint::PrintImageFromFile()

Syntax

CDPrint::PrintImageFromFile(Image.s,X1.i,Y1.i,X2.i,Y2,Transparency.i=255)**Description**

Prints an image at the given co-ordinates and at the size defined by the co-ordinates, using the given transparency.

Parameters

Image The fully qualified filename of the image to be printed
X1,Y1 Co-ordinates of the top and left corner of the image
X2,Y2 Co-ordinates of the bottom right corner of the image in mm
Transparency The transparency to be applied when printing the image

Return value

None.

2.7 Print Canvas

CDPrint::Canvas()**Syntax**

CDPrint::PrintCanvas(Canvas.i,X1.i,Y1.i,X2.i=0,Y2=0,Transparency.i=255)

Description

Obtains the image on the specified canvasGadget and prints that image at the given co-ordinates and at the size defined by the co-ordinates, using the given transparency.

Parameters

Canvas The handle of the canvas gadget whose image is to be printed.
X1,Y1 Co-ordinates of the top and left corner of the image
X2,Y2 Co-ordinates of the bottom right corner of the image in mm
Transparency The transparency to be applied when printing the image

Return value

None.

2.8 Print Text

CDPrint::Text()**Syntax**

CDPrint::PrintText(X1.i,Y1.i,Font.s,Size.i,Text.s,Color.i=\$FF000000)

Description

Prints text at the given co-ordinates using the given font,size and colour.

Parameters

X1,Y1 Co-ordinates of the top and left corner of the text
Font The name of the font to be used when printing

Size The size, in points, for the font
Text The text to be printed
Color the colour for the printed text

Return value

None.