

GDI + 1.0

by Stephen Rodriguez

General.

GDI+ is a replacement for the Windows GDI subsystem used for representing graphical objects and transmitting them to output devices such as monitors and printers etc.

GDI+ generally requires Windows XP onwards (32 or 64-bit) although there is a redistributable available for earlier versions of Windows covering GDI+ 1.0.

GDI+ exposes a flat c-style API which consists of about 600 functions. There is also a class-based API for use by c/c++ programmers and it is this class which this wrapper of ours seeks to mimic and which will be for use by Purebasic programmers.

Read more about GDI+'s flat API [here](#).

Read more about GDI+'s class-based API [here](#).

Our class-based GDI+ wrapper.

As stated above, this wrapper seeks to mimic the official c/c++ class wrapper around GDI+. We do not seek to mimic every aspect of the c/c++ class wrapper in the first instance, but will of course extend our support for more features of GDI+ as this project progresses.

We are targeting version 1.0 of GDI+ since the newer version 1.1 is not redistributable and is only installed on versions of Windows from Vista onwards. We shall add support for version 1.1 at a future date.

This wrapper supports both 32-bit and 64-bit versions of GDI+.

An important thing to note is that our class-based wrapper is not an attempt at constructing our own high level API through which to access GDI+. We are simply wrapping portions of the flat GDI+ API within a convenient class structure along the lines of the official MS class-based wrapper. There is no attempt at making GDI+ easier to use through our wrapper.

Having said this, however, there are instances in which we have simplified access to certain aspects of the GDI+ API where it was convenient to do so because of the differences between c/c++ and Purebasic.

For example, we have simplified the business of specifying *encoder parameters* which are generally used when saving images in order to specify certain encoder dependant parameters.

We have also, for example, added a 'CatchImage()' type constructor to our image class in order to keep things more in line with Purebasic's image library.

Just as important are the many demo programs which accompany each of our classes.

The structure of our class-based GDI+ wrapper.

Our class-based wrapper consists of the following files :

- "gdiPlus_Header.pbi"

This contains all required constants, structures and imports from the GDI+ library.

- "gdiPlus.pbi"

Our main file. This is the file we add to our own projects via '*XIncludeFile*' etc.

This file contains, amongst other things, some additional constants which we have introduced to simplify certain aspects of the GDI+ API. For example, when saving images, GDI+ requires that we provide the CLSID identifying the encoder we wish to use (e.g. JPEG). We have removed the need for monkeying around with CLSIDs and the like by using simple constants identifying the various encoders, constants listed within this file. An example of such a constant is : *#GDIPLUS_JPEG*.

This file is worth a look through on occasion.

- Individual class files.

An example is the source file : "gdiPlus_ImageClass.pbi" which contains the implementation of our 'image' class etc.

These individual class files will generally reflect/mimic the corresponding class files from the official c/c++ class-based wrapper. There will be some differences, however, as described below.

- Import libraries : "GdiPlusx86.lib" and "GdiPlusx64.lib".

32-bit and 64-bit import libraries respectively. The appropriate library is used automatically depending on whether we are compiling for 32-bit or 64-bit etc.

Adding our GDI+ wrapper to a project is a simple matter of adding the following line to the top of our project's source :

XIncludeFile "gdiPlus.pbi"

Individual class files.

For each of our individual class files (such as "gdiPlus_ImageClass.pbi") there will correspond one or more c/c++ class files from the official MS class-based wrapper.

For example, in the case of "gdiPlus_ImageClass.pbi", this file generally covers the **Bitmap** and **Image** classes from the c/c++ class wrapper. That is to say that the functionality wrapped up by these two c/c++ classes is to be found within our single Image class.

There will of course be some differences, most notably in our list of class constructors and in the fact that we do not give 100% coverage to all of the official c/c++ class methods etc.

Each of our class files comes equipped with it's own help document listing, amongst other things, the class constructors and class methods etc. Detailed documentation of each method is not included, however, since the majority of methods exactly mimic their c/c++ counterparts and are thus documented within the appropriate MSDN webpage(s).

Some of our classes are what we would term '*helper classes*' and do not contain any code of their own. For example, our *BitmapData* class (mimicking the c/c++ **BitmapData** class) contains a simple macro for use in populating a BitmapData structure.

List of individual classes.

The following lists the classes we have currently completed together with those c/c++ wrapper classes upon which each class is based.

- **BitmapData**

This is a *helper-class* containing macros only.

c/c++ wrapper classes.

BitmapData.

- **EncoderParameters**

This is a full OOP class.

c/c++ wrapper classes.

EncoderParameter.

EncoderParameters.

- **Brush**

This is an **abstract base class** from which all other brush classes inherit.

c/c++ wrapper classes.

Brush.

- **Font**

This is a full OOP class.

c/c++ wrapper classes.

Font.

- **FontCollection**

This is a full OOP class.

c/c++ wrapper classes.

FontCollection.

InstalledFontCollection.

PrivateFontCollection.

- **FontFamily**

This is a full OOP class.

c/c++ wrapper classes.

FontFamily.

- **HatchBrush**

This is a full OOP class.

c/c++ wrapper classes.
HatchBrush.

- **Image**

This is a full OOP class.

c/c++ wrapper classes.
Bitmap.
Image.

- **LinearGradientBrush**

This is a full OOP class.

c/c++ wrapper classes.
LinearGradientBrush.

- **Matrix**

This is a full OOP class.

c/c++ wrapper classes.
Matrix.

- **PointF**

This is a *helper-class* containing macros only.

c/c++ wrapper classes.
PointF.

- **PointI**

This is a *helper-class* containing macros only.

c/c++ wrapper classes.
Point.

- **PropertyItem**

This is a *helper-class* containing macros only.

c/c++ wrapper classes.
PropertyItem.

- **RectF**

This is a *helper-class* containing macros only.

c/c++ wrapper classes.
RectF.

- **RectI**

This is a *helper-class* containing macros only.

c/c++ wrapper classes.
Rect.