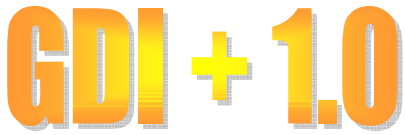

by Stephen Rodriguez

Image Class

General.

Our GDI+ Image class is housed within the "gdiPlus_ImageClass.pbi" source file.

The class contains elements of the following c/c++ classes :

Bitmap
Image

This is not a *helper-class* (containing just simple macros etc.) but a full OOP class.

Interface / base-class.

Our image class exposes a single **interface** with the name gdiPlus_Image.

This interface extends our gdiPlus_BaseClass interface which exposes a single method :

GetGdiPlusHandle.i()

which can be used to retrieve the bitmap/image handle used natively by GDI+.

'*gdiPlus_Image*' constructors.

The following all return (if successful) instances of our gdiPlus_Image class whose methods are listed in the following section. Some of these correspond exactly to constructors from the appropriate c/c++ wrapper classes and so the reader is advised to look on the appropriate MSDN pages for detailed descriptions.

gdiPlus_CatchImage(*address, size)

Similar to Purebasic's CatchImage() function.

gdiPlus_CreateBitmap(width, height, pixelFormat=#PixelFormat24bppRGB)

See the "gdiPlus_Header.pbi" for a listing of all possible pixel formats.
You can use the ***\LockBits()*** and ***\UnlockBits()*** methods to initialise the pixel bits.

gdiPlus_CreateBitmapFromBitmapInfo(*bitmapinfo.BITMAPINFO, *pxBits)

gdiPlus_CreateBitmapFromGraphics(width, height,
****graphics._gdiPlus_GenericObject_Template)***

gdiPlus_CreateBitmapFromHBITMAP(hbm, hpal=0)

gdiPlus_CreateBitmapFromHICON(hicon)

gdiPlus_CreateBitmapFromResource(hInstance, bitmapName$)

gdiPlus_CreateBitmapFromScan0(width, height, stride, pixelFormat, pixBytes)

See the "gdiPlus_Header.pbi" for a listing of all possible pixel formats.
Stride should be negative for a bottom-up arrangement of pixels and always a multiple of 4.

gdiPlus_LoadImage(fileName$)

'*gdiPlus_Image*' methods.

Unless specified otherwise, the following all return a gdiPlus status code (beginning with *#Ok*).

Some of these methods correspond exactly to methods from the appropriate c/c++ wrapper classes and so the reader is advised to look on the appropriate MSDN pages for detailed descriptions.

Destroy()

Clone.i()

Returns, if successful, a new gdiPlus_Image object containing an exact copy of the original image.

CloneAreaF.i(x.f, y.f, width.f, height.f, format)

Returns, if successful, a new gdiPlus_Image object.

CloneAreaI.i(x, y, width, height, format)

Returns, if successful, a new gdiPlus_Image object.

GetBounds.i(*srcRect.RectF, *srcUnit.INTEGER)

GetFlags.i()

Returns a combination of Image flags, e.g. *#ImageFlagsHasAlpha* etc.

GetHBITMAP.i(backColor)

Returns, if successful, a gdi bitmap. The backColor parameter is ignored if the bitmap is totally opaque.

GetHeight.i()

Returns the image height.

GetHICON.i()

Returns, if successful, a icon handle.

GetHorizontalResolution.f()

Returns the resolution in dpi.

GetPalette.i(*palette.ColorPalette, size)

See the "gdiPlus_Header.pbi" for a listing of the ColorPalette structure.

GetPaletteSize.i()

Returns the size, in bytes, of the image's palette. This will always be at least 12 bytes because the GDI+ ColorPalette structure always includes at least 1 ARGB entry.

GetPixel.i(x, y)
Returns the pixel color in 32-bit ARGB format.

GetPixelFormat.i()
Returns a pixel format constant, e.g. *#PixelFormat32bppARGB*.

GetRawFormat.i()
Returns a image format constant, e.g. *#GDIPLUS_BMP*.

GetVerticalResolution.f()
Returns the resolution in dpi.

GetWidth.i()
Returns the image width.

RotateFlip.i(rfType)
See the "gdiPlus_Header.pbi" for a listing of the rotate-flip enumeration.

Save.i(fileName$, type, *params._gdiPlus_EncoderParameters_Template=0)
Type = one of the save image format constants : *#GDIPLUS_BITMAP* etc.
*params should point to an object of type gdiPlus_EncoderParameters.

SaveToMemory.i(type, *size.INTEGER=0, *params._gdiPlus_EncoderParameters_Template=0)
*size points to a buffer which receives the number of bytes in the saved image.
*params should point to an object of type gdiPlus_EncoderParameters.
Returns, if successful, the address of a buffer containing the image data. Use FreeMemory() when finished.

SetPalette.i(*palette.ColorPalette)
Doesn't seem possible to set the palette of an image in which a palette has already been set.
See the "gdiPlus_Header.pbi" for a listing of the ColorPalette structure.

SetPixel.i(x, y, ARGBColor)

SetResolution.i(xdpi.f, ydpi.f)

Locking / unlocking pixel bits. For use with retrieving and/or setting scan lines etc. Much like GetDIBits_() and so on.

LockBits.i(*rc.RectI, flags, pixelFormat, *lockedBitmapData.BitmapData)
flags = a combination of the 'Image lock mode' constants, e.g. *#ImageLockModeUserInputBuf*.
pixelFormat = *#PixelFormat32bppARGB* etc.
*lockedBitmapData points to a BitmapData structure (see the BitmapData helper class) and its use depends upon the flags parameter.
See also the RectI helper class.

UnlockBits.i(*lockedBitmapData.BitmapData)
Call after using the \LockBits() method, passing the same *lockedBitmapData pointer.

Multi-page images.

GetFrameCount.i(dimension)

Dimension is one of the multi-frame dimension IDs, e.g. *#GDIPLUS_FRAMEDIMENSIONPAGE*.
Returns the number of frames in the specified dimension.

SaveAdd.i(*params._gdiPlus_EncoderParameters_Template=0)

Use to save the currently selected frame to the most recently saved image file (must be a multi-image file).
*params should point to an object of type gdiPlus_EncoderParameters.

SaveAddImage.i(*newImage, *params._gdiPlus_EncoderParameters_Template=0)

Use to add a new image to the most recently saved image file (must be a multi-image file).
*params should point to an object of type gdiPlus_EncoderParameters.

SelectActiveFrame.i(dimension, index)

Dimension is one of the multi-frame dimension IDs, e.g. *#GDIPLUS_FRAMEDIMENSIONPAGE*.
Index is 0-based.

Properties.

GetPropertyCount.i()

Returns the number of properties stored in the image.

GetPropertyIdList.i(numProperties, *buffer)

GetPropertyItem.i(propId, propSize, *propItem.PropertyItem)

propId is one of the Image property Id tags, e.g. *#PropertyTagExifIFD*.
*propItem points to a PropertyItem structure (see the PropertyItem helper class).

GetPropertyItemSize.i(propId)

Returns a Image property Id Tags constant, e.g. *#PropertyTagExifIFD*.

RemovePropertyItem.i(propId)

SetPropertyItem(*item.PropertyItem)

*item points to a PropertyItem structure (see the PropertyItem helper class).