USING STOCK PENS



The text describes how to use stock pens in Windows programming. Stock pens are predefined pens that Windows provides.

The three stock pens are BLACK\_PEN, WHITE\_PEN, and NULL\_PEN.

* BLACK\_PEN draws a solid black line with a width of one pixel.
* WHITE\_PEN draws a solid white line with a width of one pixel.
* NULL\_PEN is a pen that doesn't draw.

To use a stock pen, you first need to obtain a handle to it using the GetStockObject function. The GetStockObject function takes the name of the stock pen as an argument and returns a handle to the pen. For example, the following code obtains a handle to the WHITE\_PEN:

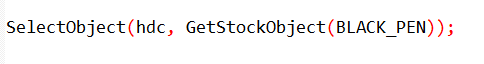


Once you have a handle to a pen, you need to select it into the device context using the SelectObject function. The SelectObject function takes two arguments: the device context and the pen handle. The following code selects the WHITE\_PEN into the device context:

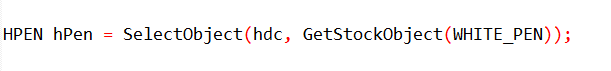


Now any lines that you draw will use the WHITE\_PEN until you select another pen into the device context or release the device context handle.

To return to using the BLACK\_PEN, you can get the handle to that stock object and select it into the device context in one statement:



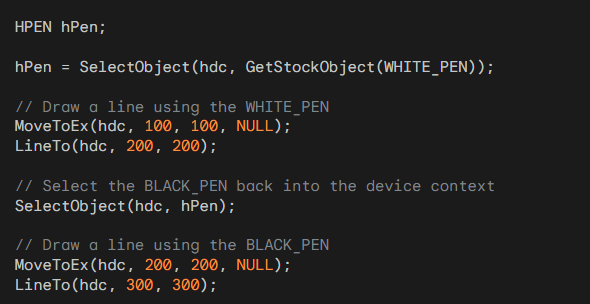
The SelectObject function returns the handle to the pen that had been previously selected into the device context. If you start off with a fresh device context and call:



The current pen in the device context will be WHITE\_PEN and the variable hPen will be the handle to BLACK\_PEN. You can then select BLACK\_PEN into the device context by calling:



Example Code:



This code will draw a white line from (100, 100) to (200, 200) and a black line from (200, 200) to (300, 300).

CREATING, SELECTING AND DELETING PENS

Creating Pens

To create a custom pen, you use the CreatePen or CreatePenIndirect functions. These functions take several parameters that define the appearance of the pen, such as the line style, line width, and color. The functions return a handle to the pen, which you can then select into the device context using the SelectObject function.

Selecting Pens

Only one pen can be selected into the device context at a time. To select a pen, you call the SelectObject function with the device context handle and the pen handle as arguments. Once a pen is selected, all lines that you draw will use that pen until you select another pen or release the device context.

Deleting Pens

When you are finished with a pen, you should delete it using the DeleteObject function. This will free up the resources that the pen was using. However, you should not delete a pen while it is still selected into a device context.

GDI Objects

A logical pen is a type of GDI object. GDI objects are resources that are managed by the Graphics Device Interface (GDI). There are six types of GDI objects: brushes, bitmaps, regions, fonts, palettes, and pens. GDI objects are selected into the device context using the SelectObject function.

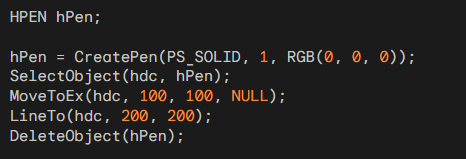
*Rules for Using GDI Objects.*

There are three rules for using GDI objects:

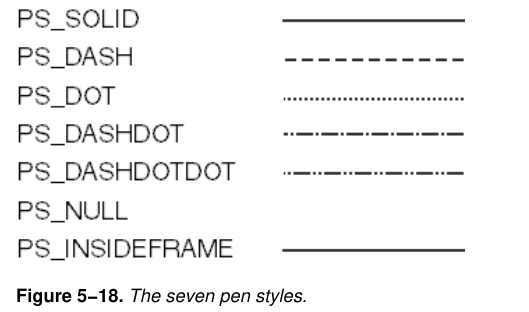
* You should eventually delete all GDI objects that you create.
* Don't delete GDI objects while they are selected in a valid device context.
* Don't delete stock objects.

*Example Code*

The following code shows how to create a pen, select it into the device context, draw a line with the pen, and then delete the pen:

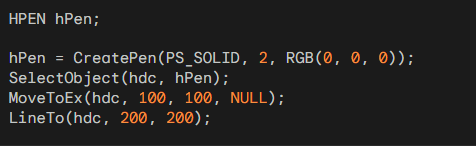


This code will draw a black line from (100, 100) to (200, 200).



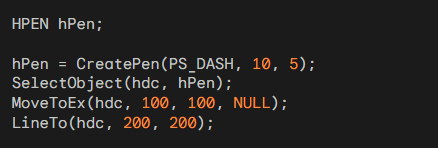
PS\_SOLID

A solid pen draws a solid line with a constant width. The width of the line is specified by the iWidth parameter to the CreatePen function. The following code shows how to draw a solid black line with a width of 2 pixels:



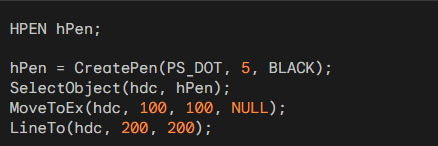
PS\_DASH

A dashed pen draws a line that is made up of a series of dashes. The length of the dashes and the spacing between them is specified by the iWidth parameter to the CreatePen function. The following code shows how to draw a dashed black line with a dash length of 10 pixels and a spacing of 5 pixels:



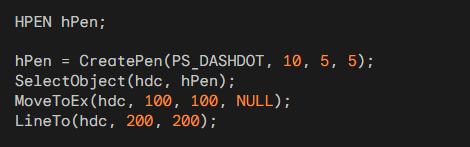
PS\_DOT

A dotted pen draws a line that is made up of a series of dots. The size of the dots is specified by the iWidth parameter to the CreatePen function. The following code shows how to draw a dotted black line with a dot size of 5 pixels:



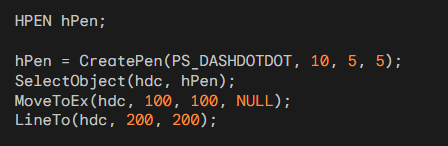
PS\_DASHDOT

A dash-dot pen draws a line that is made up of a series of alternating dashes and dots. The length of the dashes and the size of the dots are specified by the iWidth parameter to the CreatePen function. The following code shows how to draw a dash-dot black line with a dash length of 10 pixels, a dot size of 5 pixels, and a spacing of 5 pixels between the dashes and dots:



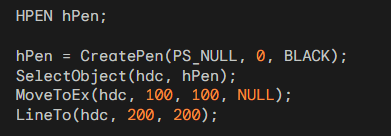
PS\_DASHDOTDOT

A dash-dot-dot pen draws a line that is made up of a series of alternating dashes and double dots. The length of the dashes and the size of the dots are specified by the iWidth parameter to the CreatePen function. The following code shows how to draw a dash-dot-dot black line with a dash length of 10 pixels, a dot size of 5 pixels, and a spacing of 5 pixels between the dashes and dots:



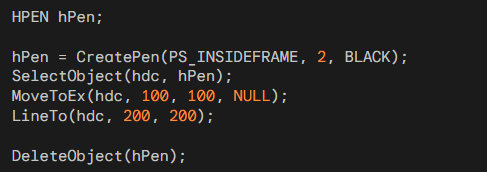
PS\_NULL

A null pen does not draw anything. The following code shows how to draw a null line:



PS\_INSIDEFRAME

The PS\_INSIDEFRAME pen style is a special pen style that is used to draw lines that are clipped to the inside of a frame. The frame is specified by the iWidth parameter to the CreatePen function. The following code shows how to draw a solid black line with a width of 2 pixels that is clipped to the inside of a frame with a width of 10 pixels:



The provided code snippet is a complete and functional code segment that draws a line using the PS\_INSIDEFRAME pen style, a width of 2 pixels, and a black color. It also deletes the pen when it is no longer needed.

Creating and Managing Pens

Pens are used to draw lines and shapes in Windows programming. There are three types of pens: solid pens, dotted pens, and dashed pens.

* *Solid pens draw solid lines, dotted pens draw lines that consist of a series of dots, and dashed pens draw lines that consist of a series of dashes.*

The CreatePen and CreatePenIndirect functions are used to create pens. The CreatePen function takes three arguments: the pen style, the line width, and the color of the pen. The CreatePenIndirect function takes a pointer to a structure of type LOGPEN, which contains the pen style, line width, and color.

Once a pen has been created, it must be selected into the device context before it can be used. The SelectObject function is used to select a pen into the device context. The SelectObject function takes two arguments: the device context and the pen handle.

When a pen is no longer needed, it should be deleted using the DeleteObject function. The DeleteObject function takes a pen handle as an argument.

Line Width

The line width is the width of the line that the pen draws. The line width is specified by the iWidth parameter to the CreatePen function. The iWidth parameter can be a positive integer or zero. A positive integer specifies the width of the line in pixels. Zero specifies a line width of one pixel.

Color

The color of the pen is the color of the line that the pen draws. The color is specified by the crColor parameter to the CreatePen function. The crColor parameter is a COLORREF value, which is a 32-bit value that contains the red, green, and blue components of the color.

Dithered Colors

The PS\_INSIDEFRAME pen style is the only pen style that can use a dithered color. A dithered color is a color that is made up of a pattern of other colors. Dithered colors are used to simulate colors that cannot be displayed by the device.

Creating Pens at Initialization

If your program uses a lot of different pens that you initialize in your source code, you can create the pens using the CreatePenIndirect function and store the pen handles in static variables. This can be more efficient than creating the pens each time you need to use them.

Selecting and Deleting Pens

To select a pen into the device context, you use the SelectObject function. The SelectObject function takes two arguments: the device context and the pen handle. To delete a pen, you use the DeleteObject function. The DeleteObject function takes a pen handle as an argument.

Example Code

The following code shows how to create, select, and delete pens:

