

By Charles E. Kindel, Jr.

Table of Contents	
µTable of Contents	1
Introduction	2
Exported Functions	2
Class Styles	3
Messages	3

Introduction

ItsyBitsy is a support module that allows you to create windows that look very much like a popup window with a system menu and caption bar, except that it is scaled down to about 2/3 scale. This document describes the ItsyBitsy support module along with the class styles and messages it supports. Below are two examples of ItsyBitsy windows; one with a horizontal caption and one with a vertical caption:

ItsyBitsy is implemented as a C callable support module that you can call from almost any window procedure. The only limitation on the kinds of windows that can use an itsybitsy caption is the window must not have a menu. The reason for this is that ItsyBitsy uses the WM_NC* (non-client area) messages to draw itself and handle user input. When Windows goes to draw a menu for a window it only looks at styles it knows and places the menu appropriately. Since Windows does not know about itsybitsy windows it does not know where to place the menu.

Also, currently the ItsyBitsy support module does not support minimize/maximize buttons. These could be added with little difficulty, however.

Exported Functions

The following functions are exported by ITSYBITSY.C. The most important function, and the one most likely to be used is `ibDefWindowProc()`.

`ibDefWindowProc`

`DWORD FAR PASCAL ibDefWindowProc(hWnd, wParam, lParam)`

This function is the heart of the ItsyBitsy support module. It should be called from your window procedure instead of `DefWindowProc()`. For example:

```
DWORD FAR PASCAL fnYourWndProc( HWND hWnd, WORD wParam,
LONG lParam )
{
```

```
/*
```

```
* Do your WM_CREATE stuff here.
```

```
*/
```

Charles E. Kindel, Jr.

11/19/2024 12:37 PM§

```
/*
```

```
* Do your WM_SYSCOMMAND processing. Make sure you pass
```

```
* WM_SYSCOMMANDS on to ibDefWindowProc because it uses them!
```

```
*/
```

```
/*
```

```
* Do your WM_PAINT stuff here.
```

```
*/
```

```
return ibDefWindowProc( hWnd, wParam, lParam );
```

```
}
```

ibAdjustWindowRect

VOID FAR PASCAL ibAdjustWindowRect(hWnd, lpRect)

This function works in a similar fashion to the SDK function AdjustWindowRect() except that it is more dynamic and works correctly on ItsyBitsy windows. Use this function to calculate the size your window needs to be (in window coordinates) in order for a client area defined by lpRect (in client coordinates) to fit within that window. The TOOLPAL window class uses this function.

An example:

RECT

```
/*
```

```
* Assume that xSize and ySize define the x and y sizes of the  
* desired client area.
```

```
*/
```

```
SetRect( &rc, 0, 0, xSize, ySize );
```

Charles E. Kindel, Jr.

11/19/2024 12:37 PM§

```
ibAdjustWindowRect( hWnd, &rc );  
// when this returns rc contains  
  
// the "WindowRect" needed.  
  
SetWindowPos( hWnd, NULL, 0, 0,  
// Make your window the right  
  
rc.right - rc.left,  
// size!  
  
rc.bottom - rc.top,  
  
SWP_NOMOVE | SWP_NOZORDER );
```

Class Styles

The following class styles are defined for the ItsyBitsy windows beyond the standard Windows WS_* window class styles. It is important that if you are using the ItsyBitsy support module within a custom control that defines its own window styles that you choose your style bits so they do not conflict with the ones defined below (found in ITSYBITS.H). For an example see the ToolPalette custom control.

Style

IBS_HORZCAPTION

IBS_VERTCAPTION

IBS_SIMPLEBORDER

IBS_NOSYSMENU

Messages

The following messages are processed by the ibDefWindowProc() function which is the heart of the ItsyBitsy support module. In most cases the only thing you have to do to enable the use of the ItsyBitsy module is to call ibDefWindowProc() instead of DefWindowProc() in your window procedure. There may be cases where you wish to handle certain messages before ibDefWindowProc() handles them.

Message

WM_SYSCOMMAND

WM_NCCALCSIZE

Charles E. Kindel, Jr.

11/19/2024 12:37 PM§

WM_NCHITTEST

WM_NCLBUTTONDBLCLK

WM_NCLBUTTONDOWN