
Subject: CBitmapEx - Free C++ Bitmap Manipulation Class

Posted by [captainc](#) on Thu, 18 Sep 2008 00:40:04 GMT

[View Forum Message](#) <> [Reply to Message](#)

Not sure if this could be useful to us...

<http://www.codeproject.com/KB/graphics/CBitmapEx.aspx>

List of public methods:

```
// void Create(long width, long height);
// void Create(CBitmapEx& bitmapEx);
// void Load(LPTSTR lpszBitmapFile);
// void Load(LPBYTE lpBitmapData);
// void Save(LPTSTR lpszBitmapFile);
// void Save(LPBYTE lpBitmapData);
// void Scale(long horizontalPercent=100, long verticalPercent=100);
// void Rotate(long degrees=0, _PIXEL bgColor=_RGB(0,0,0));
// void FlipHorizontal();
// void FlipVertical();
// void MirrorLeft();
// void MirrorRight();
// void MirrorTop();
// void MirrorBottom();
// void Clear(_PIXEL clearColor=_RGB(0,0,0));
// void Negative();
// void Grayscale();
// void Sepia(long depth=34);
// void Emboss();
// void Engrave();
// void Pixelize(long size=4);
// void Draw(HDC hDC);
// void Draw(HDC hDC, long dstX, long dstY);
// void Draw(long dstX, long dstY, long width, long height,
//   CBitmapEx& bitmapEx, long srcX, long srcY);
// void Draw(long dstX, long dstY, long width, long height,
//   CBitmapEx& bitmapEx, long srcX, long srcY, long alpha);
// void Draw(long dstX, long dstY, long dstWidth, long dstHeight,
//   CBitmapEx& bitmapEx, long srcX, long srcY,
//   long srcWidth, long srcHeight);
// void Draw(long dstX, long dstY, long dstWidth, long dstHeight,
//   CBitmapEx& bitmapEx, long srcX, long srcY,
//   long srcWidth, long srcHeight, long alpha);
// void Draw(_QUAD dstQuad, CBitmapEx& bitmapEx);
// void Draw(_QUAD dstQuad, CBitmapEx& bitmapEx, long alpha);
// void Draw(_QUAD dstQuad, CBitmapEx& bitmapEx, long srcX,
//   long srcY, long srcWidth, long srcHeight);
// void Draw(_QUAD dstQuad, CBitmapEx& bitmapEx, long srcX,
//   long srcY, long srcWidth, long srcHeight, long alpha);
// void DrawTransparent(long dstX, long dstY, long width,
```

```

// long height, CBitmapEx& bitmapEx, long srcX, long srcY,
// _PIXEL transparentColor=_RGB(0,0,0));
// void DrawTransparent(long dstX, long dstY, long width,
// long height, CBitmapEx& bitmapEx, long srcX, long srcY,
// long alpha, _PIXEL transparentColor=_RGB(0,0,0));
// void DrawTransparent(long dstX, long dstY, long dstWidth,
// long dstHeight, CBitmapEx& bitmapEx, long srcX,
// long srcY, long srcWidth, long srcHeight,
// _PIXEL transparentColor=_RGB(0,0,0));
// void DrawTransparent(long dstX, long dstY,
// long dstWidth, long dstHeight, CBitmapEx& bitmapEx,
// long srcX, long srcY, long srcWidth, long srcHeight,
// long alpha, _PIXEL transparentColor=_RGB(0,0,0));
// void DrawTransparent(_QUAD dstQuad, CBitmapEx& bitmapEx,
// _PIXEL transparentColor=_RGB(0,0,0));
// void DrawTransparent(_QUAD dstQuad, CBitmapEx& bitmapEx,
// long alpha, _PIXEL transparentColor=_RGB(0,0,0));
// void DrawTransparent(_QUAD dstQuad, CBitmapEx& bitmapEx,
// long srcX, long srcY, long srcWidth, long srcHeight,
// _PIXEL transparentColor=_RGB(0,0,0));
// void DrawTransparent(_QUAD dstQuad, CBitmapEx& bitmapEx,
// long srcX, long srcY, long srcWidth, long srcHeight,
// long alpha, _PIXEL transparentColor=_RGB(0,0,0));
// LPBITMAPFILEHEADER GetFileInfo() {return &m_bfh;}
// LPBITMAPINFOHEADER GetInfo() {return &m_bih;}
// long GetWidth() {return m_bih.biWidth;}
// long GetHeight() {return m_bih.biHeight;}
// long GetPitch() {return m_iPitch;}
// long GetBpp() {return m_iBpp;}
// long GetPaletteEntries() {return m_iPaletteEntries;}
// LPRGBQUAD GetPalette() {return m_lpPalette;}
// DWORD GetSize() {return m_dwSize;}
// LPBYTE GetData() {return m_lpData;}
// void SetResampleMode(RESAMPLE_MODE mode=RM_NEARESTNEIGHBOUR)
// {m_ResampleMode = mode;}
// RESAMPLE_MODE GetResampleMode() {return m_ResampleMode;}
// BOOL IsValid() {return (m_lpData != NULL);}
// _PIXEL GetPixel(long x, long y);
// void SetPixel(long x, long y, _PIXEL pixel);

```
