

JPEG Library Manual

(Version 1.08)

CASIO Computer Co., Ltd.

Copyright ©2011. All rights reserved.

January 2011

Table of the Contents

	Editorial Record	3
Chapter 1.	Overview	4
Chapter 2.	Operation Environment	5
Chapter 3.	Structures	6
3.1	JPEGINFO	6
3.2	BITMAP	7
3.3	THUMBINFO	8
3.4	GPSINFO	9
Chapter 4.	Functions List	10
4.1	JPGEncodeToFile	11
4.2	JPGDecodeFromFile	14
4.3	JPGEncode	15
4.4	JPGDecode	16
4.5	JPGRelease	17
4.6	JPGGetThumbnail	18
4.7	JPGEncodeToFileEx	19
4.8	JPGEncodeToFileEx2	20
4.9	JPGGetGPSInfo	22
Chapter 5.	Sample Program	23
5.1	Encoding BMP File into JPEG File	23
5.2	Displaying JPEG File on Screen	25

No part of this document may be produced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of CASIO Computer Co., Ltd. in Tokyo Japan. Information in this document is subject to change without advance notice. CASIO Computer Co., Ltd. makes no representations or warranties with respect to the contents or use of this manual and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose.

© 2011 CASIO Computer Co., Ltd. All rights reserved.

Editorial Record

[illegible]

1. Overview

The **JPEG Library** provides functions for creating a JPEG file or creating BMP data from the JPEG data. The library runs also in the Device Emulator as if it runs in an actual handheld terminal.

2. Operation Environment

Applicable Handheld Terminals

- DT-X11 series
- IT-600 series
- DT-X7 series
- IT-3100 series
- DT-X30 series
- IT-800 series
- IT-300 series
- DT-X8 series

OS

- Microsoft® WindowsCE 5.0
- Microsoft® WindowsCE 6.0
- Microsoft® Windows Mobile 6.1
- Microsoft® Windows Mobile 6.5
- Microsoft® Windows Mobile 6.5.3

Development Environment

- Microsoft® eMbedded Visual C++ Version 4.0 + SP4
- Microsoft® Visual Studio 2005 + SP1
- Microsoft® Visual Studio 2008 + SP1

Supplied Files

Table 2.1

Category	File	Description
Library main unit	JpegCe.dll	Already installed under the "Windows" folder. See note below.
Import library	JpegCe.lib	Link with an appropriate application program.
Include file	JpegCe.h	Include in the source of the application program.

Notes:

- **JpegCe.dll** is not pre-installed in the IT-600. Copy **JpegCe.dll** to the windows folder of the IT-600 for use by **JPEG Library**.
- Under the development environment include the **JpegCe.h** header file in the program source and declare the imported **JpegCe.lib** library is to be used.

3. Structures

3.1 JPEGINFO

Calling Sequence

```
typedef struct_jpeg_info
{
    DWORD dwWidth;           : Specifies the width of actual data of image.
    DWORD dwHeight;          : Specifies the height of actual data of image.
    DWORD dwValidWidth;      : Width of effective data of image.
                             (Specify the actually display width.)
    DWORD dwValidHeight;     : Height of effective data of image
                             (Specify the actually display height.)
    DWORD dwComponents;      : Specifies the number of color channels.
                             For color images, generally specify 3.
    DWORD dwColorSpace;      : Specifies the color space.
                             JPG_RGB : RGB888 color space
                             JPG_YUV : YUV422 color space
    DWORD dwQuality;         : Specifies the image quality in the range
                             of 0 (lowest) to 100 (highest).
    DWORD dwBaseLine;        : Sets up the baseline or progressive.
                             JPG_BASELINE      : Baseline
                             JPG_PROGRESSIVE    : Progressive
}JPEGINFO;
```

3.2 BITMAP

Calling Sequence

```
typedef struct tagBITMAP
{
    LONG    bmType;
    LONG    bmWidth;
    LONG    bmHeight;
    LONG    bmWidthBytes;
    WORD    bmPlanes;
    WORD    bmBitsPixel;
    LPVOID  bmBits;
} BITMAP;
```

Note:

For the BITMAP structure see the Win32API reference.

3.3 THUMBINFO

Calling Sequence

```
typedef struct _thumbnail_info
{
    BOOL    bThumbEnable; : TRUE-Thumbnail enable/
                        FALSE-Thumbnail disable.
    DWORD   dwWidth;      : Specifies the width of actual data
                        of thumbnail image.
    DWORD   dwHeight;     : Specifies the height of actual data
                        of thumbnail image.
    DWORD   dwReserved;   : Reserved
} THUMBINFO;
```


3.4 GPSINFO

Calling Sequence

```
typedef struct _gps_info
{
    double dbLatitude;      :
    double dbLongitude;    :
    float  fReserved;      : Reserved
    char   pMapDatum[16];  :
} GPSINFO;
```

dbLatitude

Specify either north latitude with positive value or south latitude with negative value in the unit of degree (any degree less than one degree (1°) is denoted by decimal numbers). A variable in *dbLatitude* parameter for GPS_POSITION structure can be specified as is.

dbLongitude

Specify either east longitude with positive value or west longitude with negative value in the unit of degree (any degree less than one degree (1°) is denoted by decimal numbers). A variable in *dbLongitude* parameter for GPS_POSITION structure can be specified as is.

pMapDatum[16]

Specify datum. For World Geodetic System, specify "WGS-84" with character string. For other systems, specify with character string formed in 15 characters or less and "NULL" at its end.

4. Functions List

Table 4.1

Function	Description	DT-X11	IT-600	DT-X7	DT-X30	IT-3100	IT-800	IT-300	DT-X8
JPGEncodeToFile	Encodes RGB data or YUV data to output the result as a JPEG file.	Y	Y	Y	Y	Y	Y	Y	Y
JPGDecodeFromFile	Decodes a JPEG file to output the result as BMP data.	Y	Y	Y	Y	Y	Y	Y	Y
JPGEncode	Encodes RGB data or YUV data to retrieve the resource of the encoded data.	Y	Y	Y	Y	Y	Y	Y	Y
JPGDecode	Decodes JPEG data to retrieve the resource of the decoded data (BMP data).	Y	Y	Y	Y	Y	Y	Y	Y
JPGRelease	Releases the retrieved resources.	Y	Y	Y	Y	Y	Y	Y	Y
JPGGetThumbnail	Outputs thumbnail of JPEG file to BMP file.	-	Y	Y	Y	Y	Y	Y	Y
JPGEncodeToFileEx	Encodes RGB data or YUV data to output the result as a JPEG file. It specifies "with thumbnail" or "without thumbnail".	-	Y	Y	Y	Y	Y	Y	Y
JPGEncodeToFileEx2	Encodes RGB data or YUV data to output the result as a JPEG file. It adds specified GPS positioning information and specifies "with thumbnail" or "without thumbnail"	-	-	-	Y	-	Y	Y	Y
JPGGetGPSInfo	Retrieves GPS information from JPEG file where GPS information is stored.	-	-	-	Y	-	Y	Y	Y

Y: Supported.

- : Not Supported.

Note:

The following models of DT-X30 series integrate the GPS feature.

DT-X30GR-10
 DT-X30GR-10C
 DT-X30GR-30
 DT-X30GR-30C
 DT-X30GR-15
 DT-X30GR-15C
 DT-X30G-35U
 DT-X30G-35UC
 DT-X30GR-35
 DT-X30GR-35C

The available models above are current as of April 2009.

4.1 JPGEncodeToFile

This function encodes RGB data or YUV data and outputs the result in JPEG file.

Calling Sequence

```
int JPGEncodeToFile(  
    HANDLE          hFile,  
    const LPBYTE    pbSource,  
    const JPEGINFO* pJpegInfo  
)
```

Parameters

hFile

[IN] : Specifies the handle to the JPEG file to be outputted.

pbSource

[IN] : Specifies the pointer to the top address of image data.

pJpegInfo

[IN] : Specifies the pointer to the JPEGINFO Structure.

Return Values

JPG_SUCCESS	: Normal end
JPG_INVALID_PARAM	: Incorrect parameter
JPG_ENCODE_ERROR	: Failed to output the JPEG file.

Note:

The image data format specified with **JPGEncodeToFile** and **JPGEncode** functions is described in the following pages.

RGB888 Data

This is a data format formed by one pixel as the element consisting of B, G and R (one byte (8 bits) for each element). The forefront data is at upper-left position on the image. To encode bit map data with its up and bottom directions being reversed, the data must be reversed for its up and bottom directions. Dummy pixels can be inserted into the horizontal and vertical widths. Because of this, data with dummy pixels inserted can be encoded (see note below).

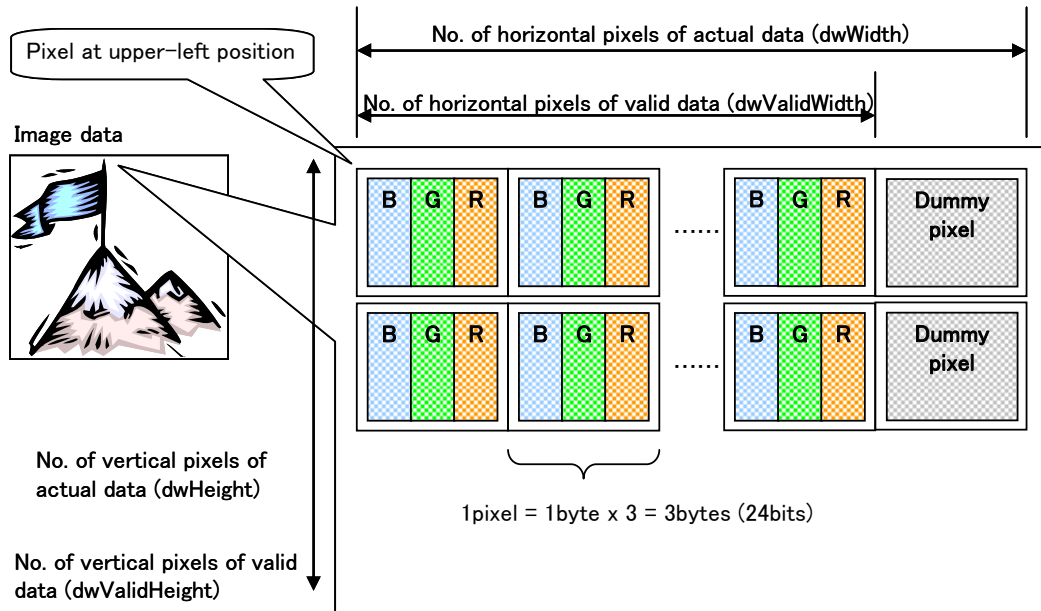


Figure 4.1

Note:

The 24-bit bit map file consists of RGB888 data. Since actual size of the horizontal width must be a multiple of 4, adjust it to a value if it is not that is a multiple of 4 by inserting dummy data.

YUV422 Data

This is a data format formed by two pixels as the element consisting of brightness (Y) 1 byte x 2, color difference of blue (U) 1 byte and color difference of red (V) 1 byte. The forefront data is at the upper-left side on the image. Because dummy pixels can be inserted in the horizontal and vertical widths, an image data consisting of odd pixel in the horizontal width can be encoded even if the YUV422 format defines that the horizontal width must be an even value.

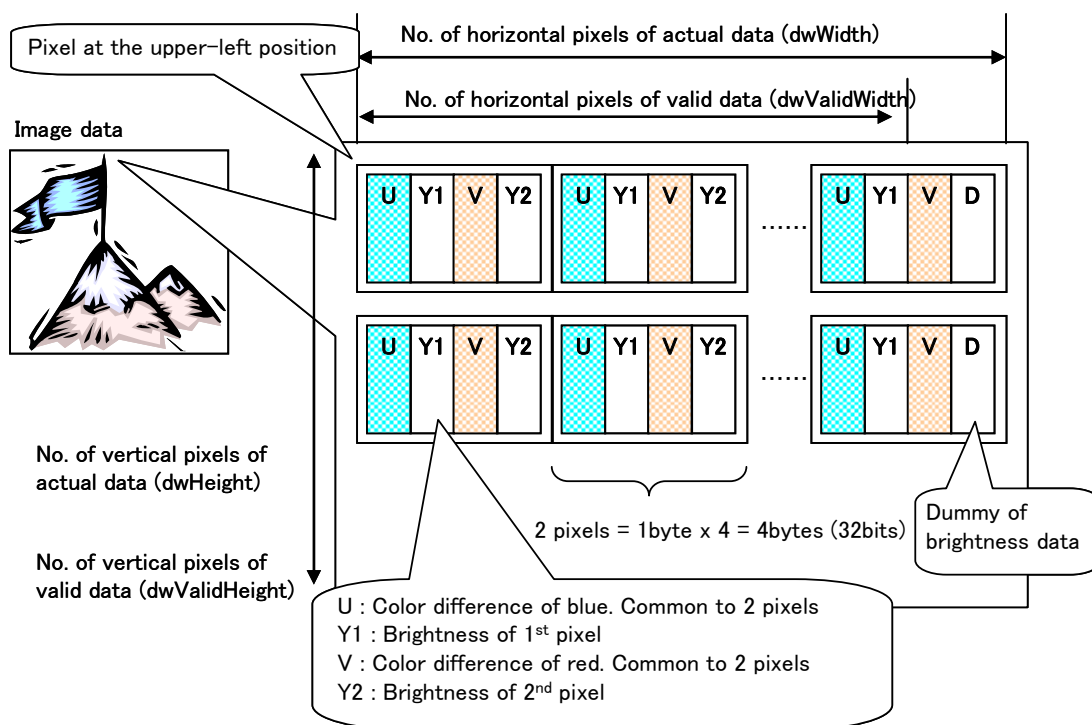


Figure 4.2

4.2 JPGDecodeFromFile

This function decodes the JPEG file and then outputs the result in BMP data. Note that the handle retrieved with this function should be released with **JPGRelease** function if it is not necessary to hold.

Calling Sequence

```
HANDLE JPGDecodeFromFile(  
    HANDLE      hFile,  
    BITMAP*     pBmpInfo  
)
```

Parameters

hFile

[IN] : Specifies the handle to the JPEG file to be outputted.

pBmpInfo

[OUT] : Specifies the pointer to the BITMAP Structure.

Return Values

JPG_INVALID_PARAM	: Incorrect parameter
JPG_DECODE_ERROR	: Failed to decode.
Others than above	: The handle to the decoded data will be returned.

4.3 JPGEncode

This function encodes RGB data or YUV data and then retrieves its resource of encoded data. Note that the handle retrieved with this function should be released with **JPGRelease** function if it is not necessary to hold.

Calling Sequence

```
HANDLE JPGEncode(  
    LPBYTE*      ppbTarget,  
    const LPBYTE pbSource,  
    LPDWORD      pdwSize,  
    const JPEGINFO* pJpegInfo  
)
```

Parameters

ppbTarget

[OUT] : Specifies the pointer to receive the top address for JPEG data to be stored.

pbSource

[IN] : Specifies the pointer to the top address of RGB or YUV image data.

pdwSize

[OUT] : The size of the JPEG data will be returned in number of bytes.

pJpegInfo

[IN] : Specifies the pointer to the JPEGINFO Structure.

Return Values

JPG_INVALID_PARAM : Incorrect parameter

JPG_ENCODE_ERROR : Failed to encode.

Others than above : The handle to the encoded data is returned.

4.4 JPGDecode

This function decodes JPEG data and then retrieves its resource of the decoded data (BMP data). Note that the handle retrieved with this function should be released with **JPGRelease** function if it is not necessary to hold.

Calling Sequence

```
HANDLE JPGDecode(  
    const LPBYTE  pbSource,  
    DWORD         dwSize,  
    BITMAP*       pBmpInfo  
)
```

Parameters

pbSource

[IN] : Specifies the pointer to the top address JPEG data.

dwSize

[IN] : Specifies the size of the JPEG data in bytes.

pBmpInfo

[OUT] : Specifies the pointer to the BITMAP Structure.

Return Values

JPG_INVALID_PARAM	: Incorrect parameter
JPG_DECODE_ERROR	: Failed to decode.
Others than above	: The handle to the decoded data is returned.

4.5 JPGRelease

This function releases the retrieved resources.

Calling Sequence

```
int JPGRelease(  
    HANDLE hImage  
)
```

Parameters

hImage

[IN] : Specify the handle retrieved with either **JPGDecodeFromFile**, **JPGEncode**, or **JPGDecode** function.

Return Values

JPG_SUCCESS	: Normally released.
JPG_INVALID_PARAM	: Incorrect parameter
JPG_RELEASE_ERROR	: Failed to release.

4.6 JPGGetThumbnail

This function outputs thumbnail of JPEG file in BMP data. Note that the handle retrieved with this function should be released with **JPGRelease** function if it is not necessary to hold.

Calling Sequence

```
HANDLE JPGGetThumbnail(  
HANDLE      hFile,  
BITMAP*     pThumbBmpInfo,  
DWORD       dwFlip,  
DWORD       dwReserved  
)
```

Parameters

hFile

[IN] : Specifies the handle to the JPEG file to be decoded.

pThumbBmpInfo

[OUT] : Specifies the pointer to the BITMAP.

dwFlip

[IN] : Specify the order of row in image data that is input.

JPG_NORMAL

Specify image data stored ordinarily in the descend order, from up to bottom.

JPG_UPSIDEDOWN

Specify image data stored reversely in the ascend order, from bottom to up. For RGB data loaded from bitmap file, specify this variable.

dwReserved

[IN] : Reserved.

Return Values

JPG_INVALID_PARAM : Incorrect parameter

JPG_NOTHUMBNAIL : No thumbnail data.

JPG_MEMORY_ERROR : Out of system memory

Others than above : The handle to the decoded data is returned.

4.7 JPGEncodeToFileEx

This function encodes RGB or YUV data and then outputs in JPEG file. It specifies "with thumbnail" or "without thumbnail".

Calling Sequence

```
int JPGEncodeToFileEx(  
    HANDLE          hFile,  
    const LPBYTE    pbSource,  
    const JPEGINFO* pJpegInfo,  
    const THUMBINFO* pThumbInfo,  
    DWORD           dwFlip  
)
```

Parameters

hFile

[IN] : Specifies the handle to the JPEG file to be outputted.

pbSource

[IN] : Specifies the pointer to the top address of RGB or YUV image data.

pJpegInfo

[IN] : Specifies the pointer to the JPEGINFO.

pThumbInfo

[IN] : Specifies the pointer to the THUMBINFO.

dwFlip

[IN] : Specify the order of row in image data that is input.

JPG_NORMAL

Specify image data stored ordinarily in the descend order, from up to bottom.

JPG_UPSIDEDOWN

Specify image data stored reversely in the ascend order, from bottom to up. For RGB data loaded from bitmap file, specify this variable.

Return Values

JPG_SUCCESS : Normally released.

JPG_INVALID_PARAM : Incorrect parameter

JPG_ENCODE_ERROR : Failed to encode.

4.8 JPGEncodeToFileEx2

This function encodes RGB or YUV data and then outputs in JPEG file. It adds specified GPS positioning information and specifies "with thumbnail" or "without thumbnail".

Calling Sequence

```
int JPGEncodeToFileEx2(  
    HANDLE          hFile,  
    const LPBYTE    pbSource,  
    const JPEGINFO* pJpegInfo,  
    const THUMBINFO* pThumbInfo,  
    const GPSINFO*  pGpsInfo,  
    DWORD           dwFlip  
)
```

Parameters

hFile

[IN] : Specifies the handle to the JPEG file to be outputted.

pbSource

[IN] : Specifies the pointer to the top address of RGB or YUV image data.

pJpegInfo

[IN] : Specifies the pointer to the JPEGINFO.

pThumbInfo

[IN] : Specifies the pointer to the THUMBINFO.

pGpsInfo

[IN] : Specifies the pointer to the GPSINFO.

dwFlip

[IN] : Specify the order of row in image data that is input.

JPG_NORMAL

Specify image data stored ordinarily in the descend order, from up to bottom.

JPG_UPSIDEDOWN

Specify image data stored reversely in the ascend order, from bottom to up. For RGB data loaded from bitmap file, specify this variable.

Return Values

JPG_SUCCESS	: Normally released.
JPG_INVALID_PARAM	: Incorrect parameter
JPG_ENCODE_ERROR	: Failed to encode.

Note:

GPS positioning information can be retrieved with **GPSGetPosition** function and stored in the GPS_POSITION structure. For detail, refer to Help annexed to Microsoft® Visual Studio.

4.9 JPGGetGPSInfo

This function retrieves GPS information from JPEG file where the GPS information is stored.

Calling Sequence

```
int JPGGetGPSInfo(  
    HANDLE          hFile,  
    const GPSINFO*  pGpsInfo,  
    DWORD           dwReserved  
)
```

Parameters

hFile

[IN] : Specifies the handle to the JPEG file to be decoded.

pGpsInfo

[OUT] : Specifies the pointer to the GPSINFO.

dwReserved

[IN] : Reserved.

Return Values

JPG_SUCCESS	: Normally released.
JPG_INVALID_PARAM	: Incorrect parameter
JPG_NOGPSINFO	: GPS information is not included in the file.

5. Sample Program

5.1 Encoding BMP File into JPEG File

The program below is a sample program to save a JPEG file encoded from 24-bit BMP file.

```
BOOL EncodeBmpFile( HWND hWnd)
{
    HANDLE hFile;
    TCHAR tInputFile[ ] = TEXT("//Test.bmp");
    TCHAR tOutputFile[ ] = TEXT("//Test.jpg");
    BITMAPINFOHEADER InfoHeader;
    BITMAPFILEHEADER FileHeader;
    JPEGINFO JpegInfo;
    LPBYTE lpbRGB;
    DWORD dwBuffSize, dwReadSize;

    // Open a Bitmap File
    hFile = CreateFile( tInputFile, GENERIC_READ , 0 , NULL ,
                      OPEN_EXISTING , FILE_ATTRIBUTE_NORMAL , NULL);
    //Read BitmapFileHeader
    ReadFile( hFile , &FileHeader , sizeof (BITMAPFILEHEADER) ,
             &dwReadSize , NULL);

    // Check the File format
    if (FileHeader.bfType != 0x4D42) {
        CloseHandle(hFile);
        return FALSE;
    }

    // Read BitmapInfoHeader
    ReadFile( hFile , &InfoHeader , sizeof( BITMAPINFOHEADER),
             &dwReadSize, NULL);

    // Read RGB data
    dwBuffSize = InfoHeader.biWidth * InfoHeader.biHeight * 3 + 100;
    lpbRGB = ( LPBYTE)VirtualAlloc( NULL, dwBuffSize, MEM_COMMIT,
                                   PAGE_READWRITE);

    ReadFile( hFile, lpbRGB, dwBuffSize, &dwReadSize, NULL);
    CloseHandle(hFile);
}
```

Continue.

```

    // Create Preservation file
    hFile = CreateFile( tOutputFile, GENERIC_READ|GENERIC_WRITE , 0 ,
NULL ,
                        CREATE_ALWAYS , FILE_ATTRIBUTE_NORMAL , NULL);

    // Setup JPEGINFO structure object
    JpegInfo.dwWidth      = InfoHeader.biWidth;
    JpegInfo.dwHeight     = InfoHeader.biHeight;
    JpegInfo.dwValidWidth  = InfoHeader.biWidth;
    JpegInfo.dwValidHeight = InfoHeader.biHeight;
    JpegInfo.dwComponents  = 3;
    JpegInfo.dwColorSpace  = JPG_RGB;
    JpegInfo.dwQuality     = 100;
    JpegInfo.dwBaseLine    = JPG_BASELINE;

    //Create Jpeg File
    if( JPGEncodeToFile( hFile, lpbRGB, &JpegInfo) != JPG_SUCCESS){
        VirtualFree( lpbRGB, dwBuffSize, MEM_DECOMMIT);
        CloseHandle( hFile);
        return FALSE;
    }
    CloseHandle( hFile);
    VirtualFree( lpbRGB, dwBuffSize, MEM_DECOMMIT);
    return TRUE;
}

```


5.2 Displaying JPEG File on Screen

The program below is a sample program to get a JPEG file and to display it on the screen. It saves an image data retrieved with **DecodeJpegFile** function to get Bitmap global variable and displays it on the screen when Windows receives WM_PAINT message.

```
BITMAP g_Bitmap;
int ShowBitmap( HWND, HDC, LPBITMAP bm);

BOOL DecodeJpegFile( HWND hWnd)
{
    HANDLE hFile, hImage;
    TCHAR tInputFile[ ] = TEXT("//Test.jpg");

    // Open a JPEG file
    hFile = CreateFile( tInputFile, GENERIC_READ, 0,
                       NULL, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL);

    // JPEG File Decode Function
    hImage = JPGDecodeFromFile( hFile, &g_Bitmap);
    if (( hImage == ( LPVOID)JPG_DECODE_ERROR) ||
        ( hImage == ( LPVOID)JPG_INVALID_PARAM))
    {
        CloseHandle(hFile);
        return FALSE;
    }

    // Close the File
    CloseHandle( hFile);

    //Decode Memory opening
    JPGRelease( hImage);

    return TRUE;
}

LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam,
                          LPARAM lParam)
{
    PAINTSTRUCT ps;
    HDC hdc;

    switch ( message)
    {
```

Continue.

```

// .....
case WM_PAINT:
    hdc = BeginPaint( hWnd, &ps);
    ShowBitmap( hWnd, hdc, &g_Bitmap);
    EndPaint( hWnd, &ps);
    break;
}
return FALSE;
}

int ShowBitmap(HWND hWnd, HDC hdc, LPBITMAP bm)
{
    BITMAPINFOHEADER InfoHeader;
    RECT rec;
    double width_tmp;
    double height_tmp;
    int ShowLine;

    // Makes the header of image information
    InfoHeader.biSize      = ( DWORD)sizeof( BITMAPINFOHEADER);
    InfoHeader.biWidth     = bm->bmWidth;
    InfoHeader.biHeight    = bm->bmHeight;
    InfoHeader.biPlanes    = bm->bmPlanes;
    InfoHeader.biBitCount  = bm->bmBitsPixel;
    InfoHeader.biCompression = 0;
    InfoHeader.biSizeImage  = 0;
    InfoHeader.biXPelsPerMeter = 0;
    InfoHeader.biYPelsPerMeter = 0;
    InfoHeader.biClrUsed    = 0;
    InfoHeader.biClrImportant = 0;

    //Display position calculation
    GetClientRect( hWnd, &rec);

    width_tmp = rec.right - 20;
    height_tmp = width_tmp / ( bm->bmWidth / bm->bmHeight);

    // Show the Image
    ShowLine = StretchDIBits( hdc, 10, 30, ( int)width_tmp,
( int)height_tmp,
        0, 0, bm->bmWidth, bm->bmHeight, bm->bmBits,
( LPBITMAPINFO)&InfoHeader,
        DIB_RGB_COLORS, SRCCOPY);

    return ShowLine;
}

```