



# JPEG ライブラリマニュアル

このマニュアルは、JPEG 画像を操作するライブラリの仕様について記載します。

#### **ご注意**

- このソフトウェアおよびマニュアルの、一部または全部を無断で使用、複製することはできません。
- このソフトウェアおよびマニュアルは、本製品の使用許諾契約書のもとでのみ使用することができます。
- このソフトウェアおよびマニュアルを運用した結果の影響については、一切の責任を負いかねますのでご了承ください。
- このソフトウェアの仕様、およびマニュアルに記載されている事柄は、将来予告なしに変更することがあります。
- このマニュアルの著作権はカシオ計算機株式会社に帰属します。
- 本書中に含まれている画面表示は、実際の画面とは若干異なる場合があります。予めご了承ください。

© 2015 カシオ計算機株式会社

Microsoft, MS, ActiveSync, Active Desktop, Outlook, Windows, Windows NT, および Windows ロゴは、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。Microsoft 社の製品は、OEM 各社に、Microsoft Corporation の 100%出資子会社である Microsoft Licensing, Inc.によりライセンス供与されています。



# 目次

1.	概要	1
2.	構造体	2
2.1	JPEGINFO	3
2.2	BITMAP	4
2.3	THUMBINFO	5
2.4	GPSINFO	5
3.	関数一覧	6
3.1	JPGEncodeToFile	6
3.2	JPGDecodeFromFile	9
3.3	JPGEncode	10
3.4	JPGDecode	11
3.5	JPGRelease	12
3.6	JPGGetThumbnail	13
3.7	JPGEncodeToFileEx	14
3.8	JPGEncodeToFileEx2	15
3.9	JPGGetGPSInfo	16
4.	サンプルプログラム	17
4.1	BMP ファイルを JPEG ファイルにエンコード	17
4.2	JPEG ファイルを画面上に表示	19

# 1. 概要

JPEG ライブラリは、JPEG ファイルの作成や、JPEG データから BMP データを作成する機能を提供します。Device Emulator でも、実機と同様に動作します。

## 対象機種

DT-5200 / DT-X7 / DT-9800 / DT-5300 / IT-300 / DT-X8 / IT-9000 / IT-G500 / DT-X100 / DT-X200

## 開発環境

- Microsoft embedded Visual C++ Version 4.0 + SP4
- Microsoft Visual Studio.NET 2003 +SP1
- Microsoft Visual Studio 2005 + SP1
- Microsoft Visual Studio 2008 + SP1

## 提供ファイル

種別	ファイル名	概要
ライブラリ本体	JpegCe.dll	"¥Windows"フォルダにインストールされています
インポートライブラリ	JpegCe.lib	リンクライブラリとして、プログラムにリンクしてください
インクルードファイル	JpegCe.h	プログラムのソースファイルにインクルードしてください

※ DT-5200 には上記ファイルがプレインストールされておりません。JPEG ライブラリを使用する場合は、"¥Windows"フォルダに"JpegCe.dll"をコピーしてください。

## 使用方法

- プログラムソース内に JpegCe.h をインクルードし、リンカの依存ファイルとして JpegCe.lib を指定してください

---

## 2. 構造体

JPEG ライブラリで提供する構造体を以下に示します。

## 2.1 JPEGINFO

JPEG 画像の情報を格納します。

```
typedef struct_jpeg_info
{
    DWORD dwWidth;           // 画像の実データの横幅を指定します。
    DWORD dwHeight;         // 画像の実データの高さを指定します。
    DWORD dwValidWidth;     // 画像の有効なデータの横幅
                            // (実際に表示される横幅) を指定します。
    DWORD dwValidHeight;   // 画像の有効なデータの高さ
                            // (実際に表示される高さ) を指定します。
    DWORD dwComponents;     // カラーチャンネル数を指定します。
                            // カラー画像の場合、通常は 3 を指定します。
    DWORD dwColorSpace;     // カラースペースを指定します。
                            // JPG_RGB:RGB888 カラースペース
                            // JPG_YUV:YUV422 カラースペース
    DWORD dwQuality;        // 画質を 0 (最低) ~100 (最高) の範囲で指定します。
    DWORD dwBaseLine;      // ベースライン or プログレッシブの設定を行います。
                            // JPG_BASELINE: ベースライン
                            // JPG_PROGRESSIVE: プログレッシブ
} JPEGINFO;
```

## 2.2 BITMAP

BITMAP の情報を格納します。

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

※ BITMAP 構造体については、[Win32 API リファレンス](#)を参照してください。



## 2.3 THUMBINFO

サムネイルの情報を格納します。

```
typedef _thumbnail_info
{
    BOOL    bThumbEnable;    // サムネイルの有無を指定します
                                // TRUE : サムネイル有
                                // FALSE: サムネイル無
    DWORD  dwWidth;         // サムネイルの横幅を指定します
    DWORD  dwHeight;        // サムネイルの縦幅を指定します
    DWORD  dwReserved;
} THUMBINFO;
```

## 2.4 GPSINFO

GPS の情報を格納します。

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

### *dbLatitude*

緯度を度単位(1度未満は小数点以下で指定)で指定します。北緯は正、南緯は負の値で指定します。

GPS\_POSITION 構造体の *dbLatitude* の値をそのまま指定することができます。

### *dbLongitude*

経度を度単位(1度未満は小数点以下で指定)で指定します。東経は正、西経は負の値で指定します。

GPS\_POSITION 構造体の *dbLongitude* の値をそのまま指定することができます。

### *fReserved*

0 を指定してください。

### *pMapDatum*

測地系を指定します。世界測地系の場合は文字列"WGS-84"を指定してください。

上記以外の場合は、NULL 終端する 15 文字以内の文字列を指定してください。

## 3. 関数一覧

### 3.1 JPGEncodeToFile

RGB データ、YUV データをエンコードし、JPEG ファイルとして出力します。

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

#### パラメータ

*hFile*

[IN] 出力する JPEG ファイルのハンドルを指定します。

*pbSource*

[IN] 画像データの先頭アドレスへのポインタを指定します。

*pJpegInfo*

[IN] JPEGINFO 構造体へのポインタを指定します。

#### 戻り値

JPG\_SUCCESS : 正常終了  
JPG\_INVALID\_PARAM : 引数が正しくありません  
JPG\_ENCODE\_ERROR : JPEG ファイルの出力に失敗しました

#### 説明

本関数は RGB データ、YUV データをエンコードし、JPEG ファイルとして出力します。

## 補足

JPGEncodeToFile 関数および JPGEncode 関数で指定する、画像データのフォーマットを以下に示します。

### ■ RGB888 データ

1 ピクセルを B, G, R 各 1 byte( 8 bit )で表したデータフォーマットです。

画像の左上が先頭データとなります。このため、上下の向きが逆であるビットマップデータをエンコードする場合は、データを上下反転する必要があります。

データには縦・横ともにダミーピクセルを含むことができます。このため、ビットマップファイルのようにダミーデータを含んだデータもエンコードすることができます(※)。

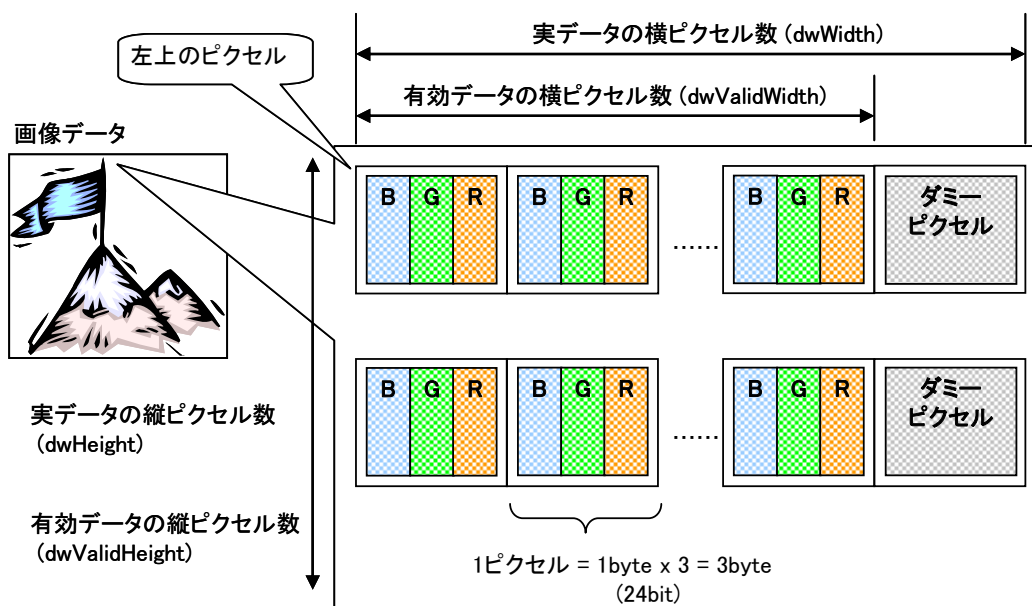


図 3.1

※ 24 bit ビットマップファイルは RGB888 データで構成されますが、横幅の実サイズが 4 の倍数でなければなりません。このため、画像の横幅が 4 の倍数でない場合は、上記のようにダミーデータを挿入し、横幅のサイズを調整しています。

### ■ YUV422 データ

2 ピクセルを、輝度(Y)1 byte×2、青の色差(U)1 byte、赤の色差(V)1 byte で表したデータフォーマットです。

先頭のデータが画像の左上になります。

データには縦・横ともにダミーピクセルを含むことができます。YUV422 フォーマットは横幅が偶数でなければなりません、これにより横幅が奇数の画像データもエンコードすることができます。

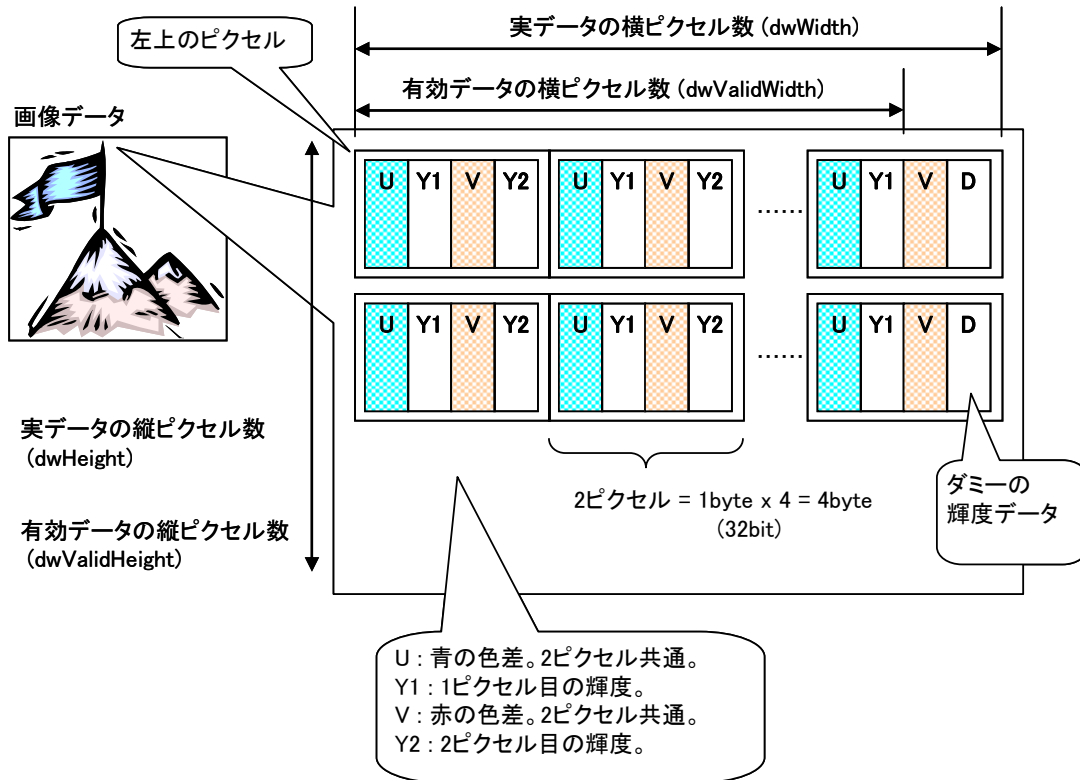


図 3.2

## 3.2 JPGDecodeFromFile

JPEG ファイルをデコードし、BMP データとして出力します。

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

### パラメータ

*hFile*

[IN] デコードする JPEG ファイルのハンドルを指定します。

*pBmpInfo*

[OUT] BITMAP 構造体へのポインタを指定します。

### 戻り値

JPG\_INVALID\_PARAM : 引数が正しくありません  
JPG\_DECODE\_ERROR : デコードに失敗しました  
上記以外 : デコードされたデータのハンドルが返されます

### 説明

本関数は JPEG ファイルをデコードし、BMP データとして出力します。

### 補足

本関数で取得したハンドルは、不要時には `JPGRelease` 関数を用いて開放してください。

## 3.3 JPGEncode

RGB データ、YUV データをエンコードし、エンコードしたデータへの資源を獲得します。

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

### パラメータ

*ppbTarget*

[OUT] JPEG データを格納する先頭アドレスを受け取るポインタを指定します。

*pbSource*

[IN] RGB もしくは YUV 画像データの先頭アドレスへのポインタを指定します。

*pdwSize*

[OUT] JPEG データのサイズをバイト単位で返します。

*pJpegInfo*

[IN] JPEGINFO 構造体へのポインタを指定します。

### 戻り値

JPG\_INVALID\_PARAM : 引数が正しくありません  
JPG\_ENCODE\_ERROR : エンコードに失敗しました  
上記以外 : エンコードされたデータのハンドルが返されます

### 説明

本関数は RGB データ、YUV データをエンコードし、エンコードしたデータへの資源を獲得します。

### 補足

本関数で取得したハンドルは、不要時には **JPGRelease** 関数を用いて開放してください。

## 3.4 JPGDecode

JPEG データをデコードし、デコードしたデータ(BMP データ)への資源を獲得します。

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

### パラメータ

*pbSource*

[IN] JPEG データの先頭アドレスへのポインタを指定します。

*dwSize*

[IN] JPEG データのサイズをバイト単位で指定します。

*pBmpInfo*

[OUT] BITMAP 構造体へのポインタを指定します。

### 戻り値

JPG\_INVALID\_PARAM : 引数が正しくありません  
JPG\_DECODE\_ERROR : デコードに失敗しました  
上記以外 : デコードされたデータのハンドルが返されます

### 説明

本関数は JPEG データをデコードし、デコードしたデータ(BMP データ)への資源を獲得します。

### 補足

本関数で取得したハンドルは、不要時には **JPGRelease** 関数を用いて開放してください。

## 3.5 JPGRelease

獲得した資源を解放します。

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

### パラメータ

*hImage*

[IN] 本ライブラリで取得したハンドルを指定します。

### 戻り値

JPG\_SUCCESS : 正常に開放しました  
JPG\_INVALID\_PARAM : 引数が正しくありません  
JPG\_RELEASE\_ERROR : 開放に失敗しました

### 説明

本関数は獲得した資源を解放します。



## 3.6 JPGGetThumbnail

JPEG ファイルからサムネイル画像を取得し、BMP データとして出力します。

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

### 説明

本関数は JPEG ファイルからサムネイル画像を取得し、BMP データとして出力します。

### パラメータ

#### *hFile*

デコードする JPEG ファイルのハンドルを指定します。

#### *pThumbBmpInfo*

BMP 構造体へのポインタを指定します。

#### *dwFlip*

取得するサムネイル画像のデータの順序を指定します。

- |                |   |
|----------------|---|
| JPG_NORMAL     | : 上から下へ画像を格納します。画像データを表示する場合は、この値を指定してください。     |
| JPG_UPSIDEDOWN | : 上下反転した画像を格納します。BMP ファイルを作成する場合は、この値を指定してください。 |

#### *dwReserved*

0 を指定してください。

### 戻り値

- |                   |                         |
|-------------------|-------------------------|
| JPG_NOTHUMBNAIL   | : ファイルにサムネイルが埋め込まれていません |
| JPG_INVALID_PARAM | : 引数が正しくありません           |
| JPG_RELEASE_ERROR | : 開放に失敗しました             |
| 上記以外              | : デコードされたデータのハンドルが返されます |

### 補足

本関数で取得したハンドルは、不要時には **JPGRelease** 関数を用いて開放してください。

## 3.7 JPGEncodeToFileEx

RGB データ、YUV データをエンコードし、JPEG ファイルとして出力します。JPEG ファイルに、指定したサイズのサムネイルを埋め込むことができます。

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

### 説明

本関数は RGB データ、YUV データをエンコードし、JPEG ファイルとして出力します。出力した JPEG ファイルに、指定したサイズのサムネイルを埋め込むことができます。

### パラメータ

#### *hFile*

出力する JPEG ファイルのハンドルを指定します。

#### *pbSource*

画像データの先頭アドレスのポインタを指定します。  
画像データの順序は *dwFlip* に従います。

#### *pJpegInfo*

JPEGINFO 構造体へのポインタを指定します。

#### *pThumbBmpInfo*

BMP 構造体へのポインタを指定します。

#### *dwFlip*

取得するサムネイル画像データの順序を指定します。

- |                |   |
|----------------|---|
| JPG_NORMAL     | : 上から下へ画像を格納します。画像データを表示する場合は、この値を指定してください。     |
| JPG_UPSIDEDOWN | : 上下反転した画像を格納します。BMP ファイルを作成する場合は、この値を指定してください。 |

### 戻り値

- |                   |                 |
|-------------------|-----------------|
| JPG_SUCCESS       | : 正常終了          |
| JPG_INVALID_PARAM | : 引数が正しくありません   |
| JPG_ENCODE_ERROR  | : ファイル出力に失敗しました |

## 3.8 JPGEncodeToFileEx2

RGB データ、YUV データをエンコードし、JPEG ファイルとして出力します。指定した GPS 位置情報を付加し、サムネイルの有無を指定することができます。

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

### 説明

本関数は RGB データ、YUV データをエンコードし、JPEG ファイルとして出力します。出力した JPEG ファイルに指定した GPS 位置情報を付加し、サムネイルの有無を指定することができます。

### パラメータ

#### *hFile*

出力する JPEG ファイルのハンドルを指定します。

#### *pbSource*

画像データの先頭アドレスのポインタを指定します。  
画像データの順序は *dwFlip* に従います。

#### *pJpegInfo*

JPGINFO 構造体へのポインタを指定します。

#### *pThumbBmpInfo*

BMP 構造体へのポインタを指定します。

#### *pGpsInfo*

GPSINFO 構造体へのポインタを指定します。

#### *dwFlip*

取得するサムネイル画像データの順序を指定します。

- |                |   |
|----------------|---|
| JPG_NORMAL     | : 上から下へ画像を格納します。画像データを表示する場合は、この値を指定してください。     |
| JPG_UPSIDEDOWN | : 上下反転した画像を格納します。BMP ファイルを作成する場合は、この値を指定してください。 |

### 戻り値

- |                   |                 |
|-------------------|-----------------|
| JPG_SUCCESS       | : 正常終了          |
| JPG_INVALID_PARAM | : 引数が正しくありません   |
| JPG_ENCODE_ERROR  | : ファイル出力に失敗しました |

## 3.9 JPGGetGPSInfo

GPS 位置情報を付加した JPEG ファイルから GPS 位置情報を取得します。

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

### 説明

本関数は GPS 位置情報を付加した JPEG ファイルから GPS 位置情報を取得します。

### パラメータ

*hFile*

GPS 位置情報を取得する JPEG ファイルのハンドルを指定します。

*pGpsInfo*

GPSINFO 構造体へのポインタを指定します。

*dwReserved*

0 を指定してください。

### 戻り値

JPG_SUCCESS	: 正常終了
JPG_NOGPSINFO	: ファイルに GPS 情報が埋め込まれていません
JPG_INVALID_PARAM	: 引数が正しくありません

## 4. サンプルプログラム

BMP ファイルを JPEG ファイルにエンコードする、および JPEG ファイルを画面上に表示するサンプルコードを示します。

### 4.1 BMP ファイルを JPEG ファイルにエンコード

24bit BMP ファイルをエンコードし、JPEG ファイルとして保存するプログラムを以下に示します。

```
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;
    DWORD           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 );
}
```

続く

```

// 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;
}

```

## 4.2 JPEG ファイルを画面上に表示

JPEG ファイルを読み込み、画面上に表示するプログラムを以下に示します。  
このプログラムでは、DecodeJpegFile 関数で取得した画像データをグローバル変数 g\_BITMAP に格納し、  
ウィンドウが WM\_PAINT メッセージを受け取ったときに表示しています。

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

BOOL DecodeJpegFile( HWND hWnd )
{
    HANDLE hFile;
    HANDLE 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 )
    {
```

続く

```

// .....
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;
}

```



## カシオ計算機お問い合わせ窓口

### 製品に関する最新情報

- 製品サポートサイト（カシオペア・ハンディターミナル）

<http://casio.jp/support/ht/>

## カシオ計算機株式会社

〒151-8543 東京都渋谷区本町 1-6-2

TEL 03-5334-4638(代)