

JpegSizer DLL Reference Manual

Release 4: May 10 2010

© Copyright 2008-2010 by TangoTools.com
All rights reserved

Prepared by Peter Martin
Email: pgm@tangotools.com
Web site: <http://www.tangotools.com/jpegsizer/dll.htm>

Introduction

The JpegSizer DLL (Dynamic Linked Library) provides basic image resizing capabilities for software developers wishing to add resizing functionality to their Windows applications.

The DLL shares the following features of JpegSizer:

- Processes original files in JPEG, GIF, TIFF, PNG and BMP formats.
- Constrains both image width *and* height.
- Allows sharpening images after resizing
- Lets you specify the JPEG file size you want, instead of specifying a compression level (which would result in a wildly unpredictable file size).
- Always outputs resized files in JPEG format.

The DLL's filename is **JpegSizer.dll**.

License Terms

A JpegSizer DLL license grants to you perpetual, worldwide, royalty-free, run-time, non-exclusive rights to distribute the DLL as an integral part of end-user Windows applications developed by you. All other rights are reserved.

To purchase a JpegSizer DLL license, please visit this web page:

<http://www.tangotools.com/jpegsizer/dll.htm>

Functions and Parameters

The DLL contains a single function named **ResizeImage**. The following parameters are passed to the function:

- Complete path to the original image file to be resized (**FN**)
- Complete path to the folder where the resized image will be saved (**Dir**)
- Width constraint of 100..2000 pixels (**MaxW**)
- Height constraint of 100..2000 pixels (**MaxH**)
- Sharpening value of 0..20 (**Sharp**)
- Desired JPEG file size of 5..500 KBytes (**KB**)
- License key (**Key**)

The first two parameters are null-terminated strings (type **String** in VB; **pChar** in Borland Delphi). All other parameters are 32-bit signed integers (type **Integer** in VB2008; **Long** in earlier VB versions; and **longint** in Borland Delphi); and are automatically limited to the ranges of values shown above.

During evaluation of the DLL, use a license **Key** of zero. This will cause the caption "SAMPLE" to be added to the top left corner of each resized image. After purchasing the DLL, replace zero with the license **Key** number provided. This turns off the "SAMPLE" caption.

The resized image's filename is the same as the original, except that the extension is automatically changed to **.jpg** as needed.

The **ResizeImage** function returns a result code, with the following possible values:

- 0 = operation successful
- 1 = unknown error
- 2 = original file does not exist
- 3 = unable to process original file
- 4 = source and destination folders are the same
- 5 = destination folder does not exist

A non-zero result code indicates that the image was not resized.

Installation on a Developer System

The installer for the JpegSizer DLL can be downloaded from here:

http://www.tangotools.com/jpegsizer/JpegSizerDLL_Setup.exe

Run the downloaded installer, and install to any desired folder.

Copy the **JpegSizer.dll** file to a suitable location for use by your application (see next section).

Installing the DLL has no effect on the host system's registry or any folder other than the install folder.

Installation on End-User Systems

In situations where the DLL will be used by a single application, it is recommended to copy the **JpegSizer.dll** file to the same folder as the application. Only this file needs to be installed.

Using the DLL Tester Program

The JpegSizer DLL is delivered with a test program for exploring the available features.

> To use the DLL tester:

1. Run the executable **JSDLLTest.exe**.
2. Enter values for the width and height (pixels), the re-sharpening value (0..20), and the desired file size (KB)
3. Select the folder where resized images will be saved
4. Click the **Resize Image** button, and select an original file for resizing.
5. The original image's file path and the result code are displayed.

Calling the DLL from a C# Application

Here is a simple example of the C# code needed to use the DLL. (Long lines are wrapped for display purposes only.)

```
using System.Runtime.InteropServices;

[DllImport("JpegSizer.dll")]
public static extern int ResizeImage(string inputfile, string
outdirpath, int MaxW, int MaxH, int Sharp, int KB, int Key);

int resultCode = 0;
string filename = "C:\\Users\\MyUserName\\Desktop\\MyImage.jpg";
string outdirpath = "C:\\Users\\MyUserName\\Pictures";
int MaxW = 640;
int MaxH = 480;
int Sharp = 4;
int KB = 80;
int Key = 0;
resultCode = ResizeImage(fileName, outdirpath, MaxW, MaxH, Sharp, KB, Key);
string wrkStr = "ResizeImage Result Code = " + resultCode;
MessageBox.Show(wrkStr);
```

This code was kindly provided by Roger Dawson, using Microsoft Visual C# 2008 (Microsoft .NET Framework Version 3.5 SP1) on Microsoft Windows 7.

Calling the DLL from a VB Application

Here is a simple example of the VB6 code needed to use the DLL. (Long lines are wrapped for display purposes only.)

```
Declare Function ResizeImage Lib "JpegSizer.dll" (ByVal FN As String,
ByVal Dir As String, ByVal MaxW As Long, ByVal MaxH As Long, ByVal Sharp
as Long, ByVal KB As Long, ByVal Key as Long) As Long

Sub Command1_Click()
Dim FN As String
Dim Dir As String
Dim I As Long
```

```
FN = "C:\MySource\MyFilename.jpg"
Dir = "C:\MyDest"
I = ResizeImage(FN, Dir, 640, 480, 4, 50, 0)
End Sub
```

Note: VB2008 replaced the `Long` data type of earlier VB versions with `Integer`. When using VB2008, the above code must be modified to use `Integer`.

Calling the DLL from a Borland Delphi Application

Here is a simple example of the Object Pascal code needed to use the DLL. Note that strings must be passed to the DLL function as `pChars`.

```
interface

function ResizeImage (FN, DestDir : pChar; MaxW, MaxH, Sharp, KB, Key
    : longint) : longint; stdcall;

implementation

function ResizeImage; external 'JpegSizer.dll';

procedure TForm1.Button1Click (Sender : TObject);
var
    I : longint;
    FN, Dir : array [0..200] of char;
begin
    StrPCopy(FN, 'C:\MySource\MyFilename.jpg');
    StrPCopy(Dir, ' C:\MyDest');
    I:= ResizeImage(FN, Dir, 640, 480, 4, 50, 0);
end;
```