

# **Batch Image Commander v2**

# Table of Contents

Introduction	3
Licensing	5
Trial Version	5
Licensing Options	5
Version History	7
32 Bit & 64 Bit Support	8
Filename convention	8
Command Line Module	9
Input File Switches	11
Resizing Switches	12
Manipulation Switches	14
Text Captioning Switches	17
Output File Switches	17
Feedback Switches	19
Template Editor	21
Loading a Profile	21
Saving a Profile	22
Open Command Prompt	22
License Management	22
Contact Information	23

# Introduction

Batch Image Commander is a pure Command Line based Batch Image Processor which can be operated manually or be evoked through another process.

The application is made up of two parts, the Command Line and the Configuration Template Editor.

As Batch Image Commander is a versatile batch image processor, it comes with a number of switches which you can use. It can get tedious when there are many switches to include and that's where the Configuration Template Editor comes into play. You can visually set the relevant options and save it into a configuration template which you can use with the application. Then all you need is to tag that template file name as "templatefile=" followed by the file name of the template file and you are good to go.

The Command Line module is the heart of the application and can be used to extend your own application via Shell/Command Line, extend a web server or an in-house server. You can evoke the command line module manually in Command Prompt or Powershell, run it on a regular schedule by adding it to the Windows Scheduler.

Batch Image Commander can be used to :-

- Resize images by pixels (downsize or upsize), by percentage or preset multipliers such as 1/2x, 1/3x, 1/4x, 2x and 4x. The Resizing by Pixels lets you downsize large images and upsize smaller images than the preset width and height settings. There is a switch which can be used to have it respect the dimension aspect ratio.
- Create Thumbnail Images of the original images.
- Supports Resample Methods option None, Lanczos3, Bell, Triangle, Hermite and Bspline which can enhance the quality of the resized images.
- Adjust Brightness and Contrast settings of the images.
- Create Negative or Grayscale Images.
- Add Text Captions to the images and align them using the preset options.
- Import JPEG, TIFF, PNG, BMP, PCX, JPEG 2000 and GIF (Non-animated)
- Export JPEG, TIFF, PNG, BMP, PCX, JPEG 2000 and GIF
- For JPEG images, you can adjust the Quality, Smoothing and turn on the Progressive load features
- For TIFF, it can handle both single and multipage TIFF files. When loading and saving to TIFF, all the pages would be preserved. When you export to a format which does not support multipage, each page would be extracted and save as their respective individual pages.

- The supported TIFF Compression includes LZW, CCITT G3, CCITT G4, Huffman, Packbit, ZIP, Deflate, Uncompressed and JPEG.
- Supports the Dots Per Inch option
- Add Prefix and Suffix to the saved filename

Batch Image Commander comes in both 32 bit and 64 bit variants.

# Licensing

## Trial Version

This application is distributed as a Try-Before-You-Buy application and lets you evaluate the application for free within a 14 days Window. Apart from the trial time frame, the trial version does not restrict any features of the application.

If you require an extension to the trial license, feel free to contact us for an extended evaluation license.

However to use the application beyond that time frame and/or for purposes other than evaluation, you will be required to purchase a license for every desktop/server you want to run this application on.

You can purchase the respective licenses from our fulfillment partner at

<https://sites.fastspring.com/batchimage/instant/batchimagecommander2>

## Licensing Options

There are 4 licensing options which includes

### **Commercial License US\$149**

This license lets you installed the application on one computer either to be operated as an extension of another application or manually evoked. The process would be used to assist the user to accomplish the user's tasks.

Case Example : B is task to resize all the TIFF files and uses the application to perform the task at hand.

### **Corporate Server License US\$549**

This license lets you installed the application on one computer/server which can be evoked by another process or through a scheduler which offer the resultant files for corporation/company/business.

Case Example : Batch Image Commander is evoked by a document management system to resize the files prior to deliver to the corporate employees, partners, coworkers or customers.

### **Web Site License US\$699**

This license lets you installed the application on one computer/server which can be evoked by a corporate web server either in an in-house or public website.

Case Example : Batch Image Commander is evoked by a web service to resize or add text captions to the documents before saving or delivery to the web users.

### **Unlimited Deployment US\$2999**

This license lets you installed the application on one or more computer(s)/server(s) within the organization within the same location.

### **Transfer of Licenses**

Apart from the Unlimited Deployment, you will be required to purchase a license for each computer/server to be used. We understand that computers/servers do get decommissioned and as such, the license agreement permits you to transfer any unused license to another machine within the organization.

The license does not permit you to distribute the application to a 3rd party either with or without monetary consideration.

### **Simultaneous Runs**

The License does not prevent you from launching multiple instance of the command line module. You can do this to utilize the multitasking features of the CPU. All instances will run simultaneously based on the processors of the computer.

### **Updates**

The purchase of a license will grant you free minor version updates which will include bug fixes and enhancements from time to time. An example of a minor update would be v2.0 to v2.9.

To update the current installation, do visit our website at <https://www.batchimage.com/downloads> and download the current installer.

As long as you are installing on an pre-licensed desktop/server, you will not need to reapply the license key.

# Version History

v2.00	<ul style="list-style-type: none"><li>• Initial Release</li></ul>
v2.00 Build 6997	<ul style="list-style-type: none"><li>• Reworked the Configuration Module</li></ul>

# 32 Bit & 64 Bit Support

Batch Image Commander v2.x supports both 32 Bit and 64 Bit Windows Operating System.

Batch Image Commander v2 is supported on

- Windows Vista
- Windows 7
- Windows 8
- Windows 10
- Windows 2012
- Windows 2016

Although the 64 Bit Windows Operating System is capable of running 32 Bit application, it does so in an emulator mode called Windows on Windows (WOW). This is a compatibility layer which effectively provides full support to 32 bit application. However since it is running on the emulator mode, it would not benefit from the larger memory access of the 64 Bit Windows and would be slightly slower as the application would not be running natively.

In Batch Image Commander v2, we provide both native 32 bit and 64 bit command line modules and configuration modules. This will let you run the application which will harness the full potential of the underlining operating system.

Included in the installed folder is both the 32 bit and 64 bit command line module.

## Filename convention

32 Bit Module : imagecom.exe  
64 Bit Module : imagecom64.exe

Both the 32 bit and 64 bit modules are features identical. On a 64 Bit Operating System, you can use either the 32 bit or 64 bit module. However, to harness the full potential of the application, using the 64 bit module would be advisable as it will work natively on the operating system and harness the full memory of the system.

The 64 Bit Installer includes both the 64 Bit and 32 Bit Command Line module while the 32 Bit Installer comes with the 32 Bit Command Line module.



# Command Line Module

In this documentation, all examples will use the 32 bit or 64 bit naming convention interchangeably. On the practical end, you can change the filename of the command line module based on your preference. It can also be moved to a different folder on the computer/server.

## TIP

Instead of having to provide the full path to the command line module, you can set up the Path parameters under System Properties > Advanced > Environment Variables. Just edit the Path parameter by tagging in the folder at which the batch image commander command line module is at. After which, you can evoke the command line without the folder path.

## HELP

To bring up the help option which would list all the switches that can be used, you could either evoke the command line using the following option

### *Usage Example*

```
imagecom.exe  
or  
imagecom.exe help
```

## COPYRIGHT

The copyright switch shows you the version number, build number and licensing information. In the trial version, it will show you the number of days left for the trial.

### *Usage Example*

```
imagecom.exe copyright
```

With the two basic switches out of the way, let's look an actual real life example on using the module.

```
imagecom.exe "k:\incoming documents\*.tif" resizingmethod=downsizebypixel  
resamplemethod=lanczos3 height=1200 width=800 keepdimensionratio=true  
colordepth=4 imageformat=tifflzw savein="k:\output\"
```

In the example above, the command line module would look for all tif files with the .tif file extensions in the "k:\incoming documents\" folder and resize them by pixels if they are larger than 1200x800 pixels while keeping the dimension ratio aspect using the Lanczso3 resampling method, saving it as a 4 bit color depth TIFF using the LZW compression into the "K:\output\" folder.

It can be tiresome typing the entire command line with all the switches. In Batch Image Commander, you can set up a profile template which will contain all the commonly used

switches you would use and thus making the command line a lot easier to manage. The example as below

```
imagecom.exe "k:\incoming documents\*.tif" templatefile="k:\templates\set1.bts"  
savein="k:\output\"
```

In the above example, the core settings would be taken from the template profile file "set1.bts". In settings in the template profile file can be temporarily overwritten by the switch in the current command line. For example, the "Savein" setting in the template file could be pointing to a different folder.

In the typically settings, you will need to assign the input and output path. The output path uses the "savein=" switch.

### **TIP**

Although file path and file name can include spacing, however in the command line, you will need to use double quotes to enclose the entire path so that the application knows that you are referring to that specific path and not consider them as two different switches.

For example, if you type

k:\incoming documents\\*.tif instead of "k:\incoming documents\\*.tif", the command line would look for files in "k:\incoming" and "documents" folders which will not yield the results you want.

At the time of writing, single quotes are not supported.

# Input File Switches

## INCOMING FILE PATH AND NAMES

There is no switch names for the file names and paths to be processed as the command line would scan through the entire tag for existing file paths and treat them as the incoming paths.

You can also provide the full path and filename to a specific file or use wildcards \* to include all files in that folder.

### *Usage Example*

**imagecom.exe "c:\incoming files\\*.jpg"**

- this would look for all files with the JPG file extensions in the C:\incoming files\ folder

**imagecom.exe "c:\incoming files\file01.jpg"**

- only the file01.jpg would be processed

**imagecom.exe "c:\incoming files\file01.jpg" "c:\incoming files\file03.jpg" "k:\document\abc.tif"**

- You are not restricted to having the command line module work with a single file as you can include all the file path and names of the files you want it to process in a single run.

## RECURSIVE

Without the recursive switch, files would only be searched in the provided folder. When you include this switch, the parent folders and all sub-folders under it would be searched.

### *Usage Example*

**imagecom.exe "c:\incoming files\\*.jpg" recursive**

## TEMPLATEFILE=

The template file is a INI style text readable file which contains the settings to be used by the command line module.

A default template might look like

```
height="0"  
width="0"  
percentage="75"  
keepdimensionratio=false  
colordepth="AsIs"  
resamplemethod="None"  
brightness="0"  
contrast="0"
```

Typically the template file can be created and edited using the provided Template Editor or by using a text editor. The typical file extension is .bts or the .ini (Batch Image Commander v1 style) but can be of any extension. You will need to include the full path with the extension to the template profile file otherwise it will search for the template file in the current folder.

### ***Usage Example***

**imagecom.exe "c:\incoming\\*.jpg" templatefile="d:\templates\set1.bts"**

## **Resizing Switches**

### **RESIZEMETHOD=**

Batch Image Commander offers several resizing methods which includes the following

- **1/2x** - Reduce by 50%
- **1/3x** - Reduce to 1/3rd of the original file dimensions
- **1/4x** - Reduce to 25%
- **2x** - Enlarge to 200% of the original file dimensions
- **3x** - Enlarge to 300%
- **4x** - Enlarge to 400%
- **resizebypercentage**
- **downsizebypixel**
- **upsizedbypixel**
- **resizebypixel**

The First 5 options are self-explanatory so we will go into more depths with the last 4 options

### **ResizebyPercentage**

This option uses the **percentage=** parameter to set the percentage to which the file would be resized to. When setting this to 100, no resizing would be performed.

### ***Usage Example***

**imagecom.exe "c:\incoming\\*.jpg" resizemethod=resizebypercentage percentage=60 savein="c:\outgoing\"**

In the above example, the photos would be resized to 60% of the original image dimensions.

**imagecom.exe "c:\incoming\\*.jpg" resizemethod=resizebypercentage percentage=150 savein="c:\outgoing\"**

In the above example, the photos would be enlarged to 150% of the original image dimensions.

### **Downsizebypixel**

This option uses the **width=**, **height=** and **keepdimensionratio=**.

With this option, images larger than the set width= and height= parameters would be downsize. By default, the Dimension Ratio Aspect would be respected unless it is set to false.

#### ***Usage Example***

**imagecom.exe "c:\incoming\\*.jpg" resizemethod=downsizebypixel height=400 savein="c:\outgoing\"**

The above would resize any images which has a height of greater than 400 pixels while respecting the aspect ratio

**imagecom.exe "c:\incoming\\*.jpg" resizemethod=downsizebypixel height=400 width=800 savein="c:\outgoing\"**

The above would check for images with height greater than 400 pixels and/or width greater than 800 pixels and downsize it to be smaller than that while respecting the aspect ratio

**imagecom.exe "c:\incoming\\*.jpg" resizemethod=downsizebypixel height=400 width=800 keepdimensionratio=false savein="c:\outgoing\"**

The above would just resize the images to the fixed height of 400 pixels by width of 800 pixels while not keeping the aspect ratio

#### **Upsizebypixel**

Upsizing by Pixel is the opposite of downsizing. This will use the same width and height and if the images are smaller than the width and height parameters, it will upsize the images to be larger than the set dimensions.

#### **Resizebypixel**

This option uses the minheight=, minwidth=, maxheight= and maxwidth= parameters.

What Resize by Pixel would do is to keep all images above the lower bounds set by the "min" dimensions and smaller than the upper bounds set by the "max" dimensions.

#### ***Usage Example***

**imagecom.exe "c:\incoming\\*.tif" resizemethod=resizebypixel minheight=300 midwidth=200 maxheight=900 maxwidth=600 savein="c:\outgoing\"**

In the above example, all images would not be smaller than the minheight and minwidth parameters and not larger than the maxheight and maxwidth parameters

Below is a chart for the pairing the resizing parameters

ResizingMethod=	ResizebyPercentage	DownsizebyPixel / UpsizebyPixel	ResizebyPixel	1/4x, 1/3x, 1/2x, 3/4x, 2x, 3x, 4x
Height=		✓		
Width=		✓		
MidWidth=			✓	
MinHeight=			✓	
MaxWidth=			✓	
MaxHeight=			✓	
KeepAspectRatio=		✓	✓	
Percentage=	✓			

## RESAMPLEMETHOD=

When images are resized, the edges may suffer from blocky edges especially for diagonal lines, using the resampling method can help smooth out the edges.

The options includes None, Lanczos3, Bell, Triangle, Hermite and BSpline. The options are rather technical and are beyond the scope of this documentation. However, you can check out [https://en.wikipedia.org/wiki/Image\\_scaling](https://en.wikipedia.org/wiki/Image_scaling) for an in-depth explanation of for the various resampling options.

Generally speaking for photos, using the resampling method Lanczos3 will offer the slowest but best image reproductions. The other options would be more apt for vector line drawings such as architectural drawings.

The default setting is None

### *Usage Example*

```
imagecom.exe "c:\incoming\*.jpg" resizemethod=1/2x resamplemethod=lanczos3
savein="c:\outgoing\"
```

## Manipulation Switches

### ORIENTATION=

The options are left90, right90, rotate180, flipdown and flipleft

The Orientation settings let you change the orientation of the images and/or pages.

Left90 will rotate the images 90° counter-clockwise. Right90 will rotate the images 90° clockwise.

Flipdown will flip the images top to bottom and the flipleft option will create the mirror image of the photo or pages.

### ***Usage Example***

**imagecom.exe "C:\incoming\\*.jpg" orientation=left90 savein="L:\outgoing\"**

### **BRIGHTNESS=**

The range is -100 to 100. 0 being the current brightness. To darken the image, use a brightness setting below 0 or to brighten the image, use a brightness setting above 0.

### ***Usage Example***

**imagecom.exe "c:\incoming\\*.jpg" brightness=20 savein="c:\outgoing\"**

### **CONTRAST=**

The range is -100 to 100. 0 being no current contrast. To increase the contrast, use a setting above 0 and to reduce it, use a setting below it.

### ***Usage Example***

**imagecom.exe "c:\incoming\\*.jpg" contrast=-15 savein="c:\outgoing\"**

### **GRAYSCALE**

This option will transform a color image to grayscale. If the output file is JPEG or TIFF with the JPEG compression, it would save as a grayscale 8 bit JPEG image.

### ***Usage Example***

**imagecom.exe "c:\incoming\\*.jpg" grayscale savein="c:\outgoing\"**

### **NEGATIVE**

This option will inverse the colors of the image.

### ***Usage Example***

**imagecom.exe "c:\incoming\\*.jpg" negative savein="c:\outgoing\"**

### **COLORDEPTH=**

As and where applicable, this option sets the color depth for the output file. The options are AsIs, 1, 4, 8 and 24.

The default setting is AsIs which will try to do the original file's Color Depth for the output file.

Setting a low color depth can drastically decrease the file size.

### ***Usage Example***

**imagecom.exe "c:\incoming\\*.tif" colordepth=8 savein="c:\outgoing\"**

## **PROGRESSIVEJPEG**

When the JPEG images are used on the internet especially when it is large and the internet connection is slow, enabling Progressive JPEG would allow the image to show up progressively instead of showing up when it is fully loaded. However with the current internet speeds, you might not see the difference

### ***Usage Example***

**imagecom.exe "c:\incoming\\*.jpg" progressivejpeg imageformat=jpg savein="c:\outgoing\"**

## **QUALITY**

This setting is used for JPEG, PNG, JPEG 2000 and TIFFJPEG Image Format. Setting a lower quality would generally result in a smaller file size but inevitably reduce the quality. A Quality Setting of between 70 to 80 would give you a best compromise between quality and file size.

The quality range is between 10 to 100.

JPEG images because of the lossy compression algorithm may yield a larger file size if the quality is set to 100.

### ***Usage Example***

**imagecom.exe "c:\incoming\\*.jpg" quality=75 imageformat=jpg savein="c:\outgoing\"**

## **SMOOTHING**

The Smoothing setting is used for JPEG images. It is particularly useful if the JPEG Quality setting is set low which may result in pixelization. Smoothing is a method used to blur the edges so that they should smooth. However excessive smoothing will result in a blur image.

Although the smooth range is between 1 to 100. It is best to keep it below 6.

### ***Usage Example***

**imagecom.exe "c:\incoming\\*.jpg" quality=50 smoothing=3 imageformat=jpg savein="c:\outgoing\"**

## **DPI**

The DPI or Dots Per Inch lets you determine the density of the dots. Typically for an on-screen image, the DPI settings is irrelevant as the display and even modern printers are able to interpolate the DPI.



### *Usage Example*

```
imagecom.exe "c:\incoming\*.jpg" dpi=300 imageformat=jpg savein="c:\outgoing\"
```

## **Text Captioning Switches**

The Text Captioning Switches lets you add a simple text into the images. Typically this can be a copyright tag or some descriptive text.

### **TEXTCAPTION=**

This contains the text caption to be added to the images. Currently, the application supports a single line text caption. Do enclose your text caption within the double quotes.

If this is left blank, the text captioning option is disabled.

### **TEXTCOLOR=**

The Text Color option uses the RGB convention which is 6 digits in the form of RRGGBB without the hex.

### **TEXTFONT=**

The text font to be used.

### **TEXTSIZE=**

This determines the text font size in pixels to be used. In Batch Image Commander, Adding of the Text Caption is performed after the image resizing.

### **TEXTALIGNMENT=**

The options are topleft, topcenter, topright, bottomleft, bottomcenter and bottomright.

If this switch is not included, no text caption will be added.

### *Usage Example*

```
imagecom.exe "c:\incoming\*.jpg" textcaption="Copyright 2017" textcolor="344FC9"  
textfont="Times New Roman" textsize=30 textalignment=topcenter savein="c:  
\outgoing\"
```

## **Output File Switches**

### **IMAGEFORMAT=**

The Image Format option is used to determine the image format of the output image file. The options includes JPG, BMP, GIF, PNG, PCX, JPEG2000, TIFFLZW, TIFFG3, TIFFG4, TIFFHuffman, TIFFPackbit, TIFFZIP, TIFFDeflate, TIFFUncompressed and TIFFJPEG.

When this option is left out, the image format of the output file would be the same as the input file format.

Let's look at the various tiff compression options which are supported.

- TIFFLZW - TIFF LZW
- TIFFG3 - CCITT G3 (Black and White)
- TIFFG4 - CCITT G4 (Black and White)
- TIFFHuffman - CCITT Huffman Encoding (Black and White)
- TIFFPackbit - Packbit Encoding
- TIFFZIP - The new ZIP Compression
- TIFFDeflate - Uses the Deflate Compression
- TIFFUncompressed - No TIFF Compression
- TIFFJPEG - JPEG Compression

## **PREFIX=**

**Add a Prefix to the output file name**

## **SUFFIX=**

**Add a Suffix to the output file name**

### ***Usage Example***

**imagecom.exe "c:\incoming\\*.tif" prefix="Time\_" suffix="\_copy" savein="c:\outgoing\"**

Assuming the original filename is "Schedule.tif", the new filename after the prefix and suffix are applied would become "Time\_Schedule\_copy.tif".

## **NEWNAME=**

The filename of the output file name. If this switch is not included, the original filename would be used. You can include a file extension would not have any effect as the image format settings would be used to determine the output file extension.

### ***Usage Example***

**imagecom.exe "c:\incoming\a.tif" newname="b" savein="c:\outgoing\"**

## **SAVEIN=**

The SaveIn switch is used to determine the folder to save the processed files in. It is advisable to set it to a different folder than the original folder unless you set it to rename the filenames.

If you do not include this switch, the processed files would be saved in the original file folder. If the newname, prefix or suffix switches are not used or the output image format is the same as the original, the original files would be overwritten. Any overwritten files cannot be recovered.

During the testing stage, it is best to keep the savein path different from the original.

## DELETEORIGINAL

With this switch added, the original files would be deleted after it is processed. Do ensure that processed file has a different filename if its to be saved in the same folder as the original or be saved in a new folder.

If you do not include the SaveIn path or set it to the same as the input path and keeping the filename the same as the original, using the DeleteOriginal switch will delete the output file.

### *Usage Example*

```
imagecom.exe "c:\incoming\*.tif" prefix="Time_" suffix="_copy" savein="c:\outgoing\  
deleteoriginal
```

## SAVEORIGINALTO=

This switch lets you save the original file to a new folder for backup.

```
imagecom.exe "c:\incoming\*.tif" prefix="Time_" suffix="_copy" savein="c:\outgoing\  
deleteoriginal saveoriginalto="c:\backupfiles\  
"
```

## RETAINORIGINALDATE

This switch when used would use the original file creation date.

### *Usage Example*

```
imagecom.exe "c:\incoming\*.tif" prefix="Time_" suffix="_copy" savein="c:\outgoing\  
retainoriginaldate
```

## Feedback Switches

### DONE=

Typically when the command line completes it's task, it will just terminate its operation and exit to the command prompt. If for instance, your process needs to monitor for an output to determine when the command line app ends, you can use the Done= switch. From here, you can just tag the string in which you want to look for.

### *Usage Example*

```
imagecom.exe "c:\incoming\*.tif" prefix="Time_" suffix="_copy" savein="c:\outgoing\  
done="ok"
```

When the command line ends, it will feedback "ok".

### ECHO

The Echo switch is mostly used as a diagnostic tool as it would display the processes it goes through in the entire process.

It will typically show the filenames of the files loaded to the app and would iterate through the files and pages. It can be useful to test if the switches in the command lines are working.

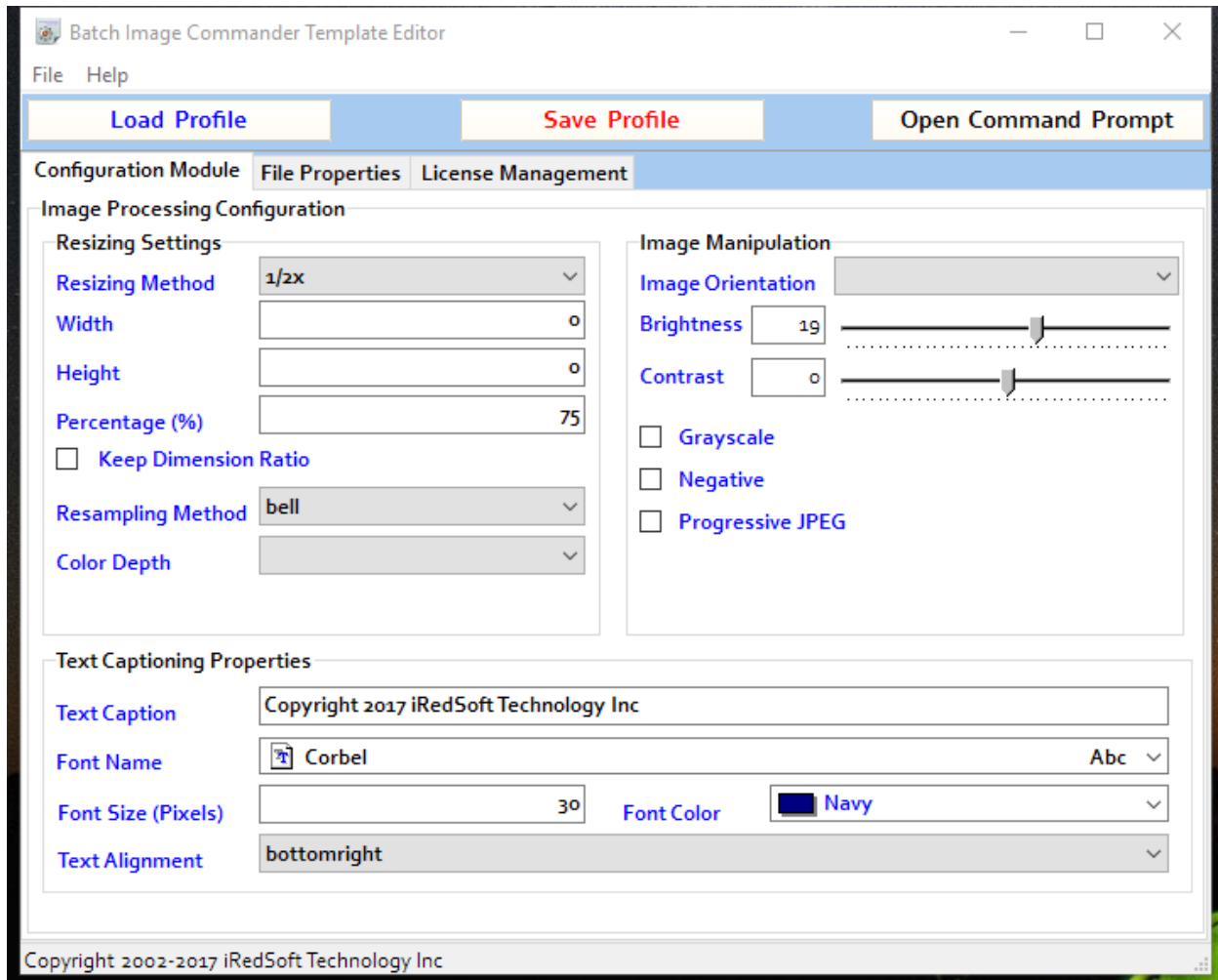
***Usage Example***

```
imagecom.exe "c:\incoming\*.tif" prefix="Time_" suffix="_copy" savein="c:\outgoing\  
echo
```

# Template Editor

The Template Editor is a tray-icon application which assist in the creation and editing of the Template File to be used with the Command Line module.

The other purpose is that of the License Management which let's you apply the license file once you purchase the license to to unlock the time usage restriction of the trial version.



## Loading a Profile

Use the Load Profile button to select an existing Template Profile File.

The new Batch Image Commander v2 default file extension for the template profile file is .bts but any file extensions included the v1 .ini would work.

Once loaded, the settings would be reflected in the template editor.

If you are coming from v1, there are some minor changes to the switches so it would be a good idea to load the prior template, make the necessary adjustments and save it before using it with command line module.

## **Saving a Profile**

Once you are done making the necessary changes, use the Save Profile button to save the current settings into file.

Although the suggested profile filename is template.bts, you can change it to what fits your requirements. You can make multiple profile files for each circumstances.

## **Open Command Prompt**

This button will open up the Command Prompt Window to the folder which the command line module is located.

## **License Management**

In the licensing tab, you can view the current license information including the remaining trial period for the trial version.

The Buy a License button will bring you to the order page for Batch Image Commander v2. And the Apply License File button will update the licensing status of the application.

When you purchase a copy of Batch Image Commander v2, we will send you a license file within 24 hours of your purchase.

# Contact Information

For any technical inquiries, feel free to email us at [support@batchimage.com](mailto:support@batchimage.com) and for sales related matters, do contact us at [sales@batchimage.com](mailto:sales@batchimage.com) .

Thank you for reading this documentation and hope that this application serves you well.