Smaller Animals™ ThumbNailer

v 8.0

Copyright 2006, Smaller Animals Software, Inc.

Main Window

JPG support from IJG 6b
TIFF support from LibTiff
JPEG2000 support from LibJasper
TGA Support from Tim Bish
PNG support from LIBPNG
Color quantization code from Dennis Lee
PCX Support from Bob Johnson's PCXLib
Some code from the www.codeproject.com and www.codeguru.com.

TIFF is a trademark of the Adobe Corporation.

All other code written by Smaller Animals Software, Inc.

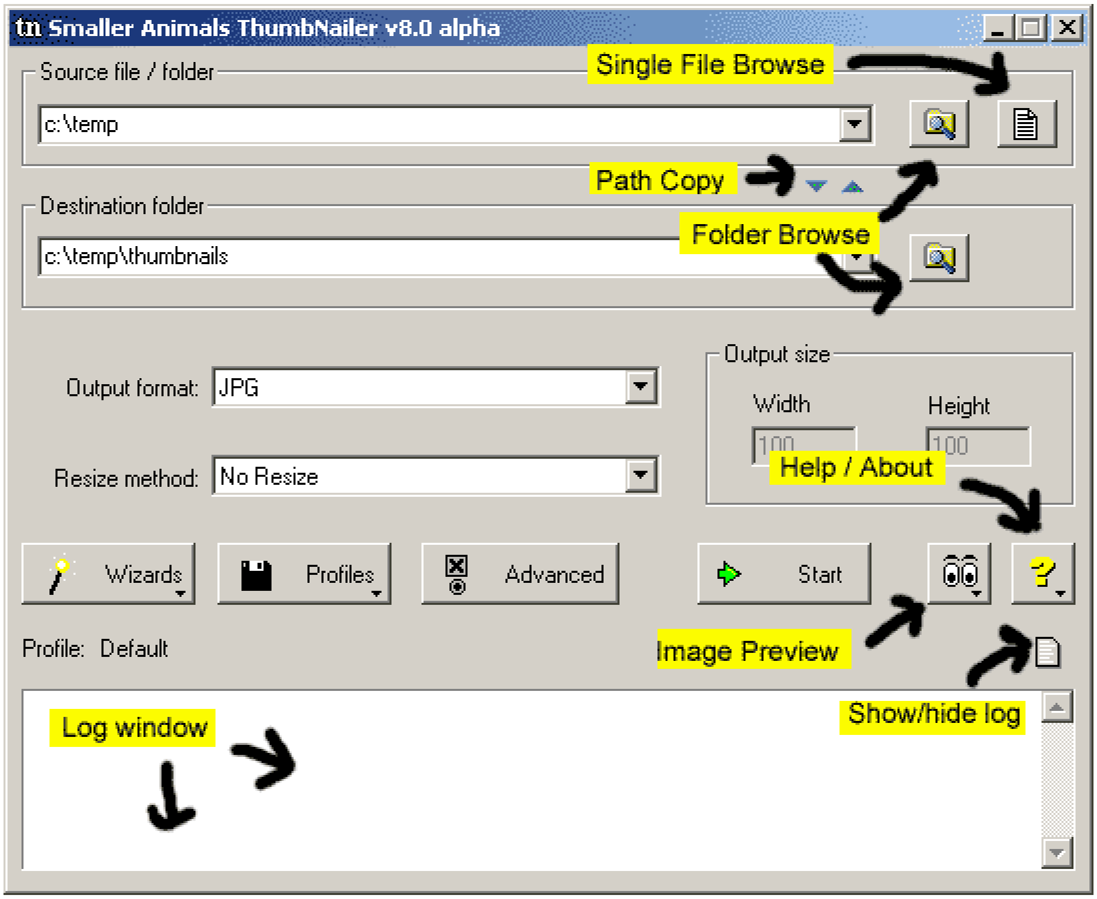
Image processing handled by *ImgSource, a separate product of Smaller Animals Software, Inc.*

http://www.smalleranimals.com/index.htm

S.A. ThumbNailer FAQ at : http://www.smalleranimals.com/thumb.htm

Registration Information

Main Window



Source file / folder: c:\temp

Destination folder: c:\temp\thumbnails

Output format: JPG

Resize method: No Resize

Output size
Width: 100
Height: 100

Profile: Default

From this screen, you can do everything S.A. ThumbNailer was originally designed to do - resize directories full of images quickly and easily. Everything else ThumbNailer does is just an enhancement to this basic idea.

Controls :

Image Process Preview

Width and Height Edit

Output Size and Resize Method Interaction

Use these controls to specify the sizes of the output images.

When you are creating thumbnails using the **Best Fit**, **Fixed Width**, **Fixed Height** or either of the cropping resize methods, the **Width** and **Height** fields represent the maximum dimensions your output images will have.

ThumbNailer goes to great lengths to insure that the aspect ratio of the output images is the same as the source images.

If you have selected the **Exact Fit** resize method, the dimensions of the output images will be exactly as you specify. This means that most output images will be distorted. E.g. you can’t make a 100 x 150 image into a 200 x 200 image without changing the aspect ratio and thus distorting the image.

Selecting the **Pixel Area** resize method causes the Width field to become a Pixel Area field. This allows you to specify a pixel area for your output images. All output images will be resized so that the product of their width and height is equal (as close as is possible) to your specified area without distorting the aspect ratio of the image.

Selecting the **Percent** resize method causes the Width field to become a Percent field. This allows you to specify a percentage value by which the images will be resized. This can be greater or less than 100.

Selecting the **Change DPI** resize method causes the Width field to become a DPI field. This is the DPI your output images will have. The number of pixels in the output image will be adjusted so that the physical size of the image stays the same.

Auto Crop sizes the smaller dimension of the image to fit the size you specify then crops the other dimension to fit. This will crop the edges of the image equally.

Fixed Width, Crop Vertical sizes the image to fit the width you specify, without distortion. Then, it crops any excess image in the vertical direction to fit the height you’ve specified. The crop position can be adjusted on the Advanced / Image options page. **Fixed Height, Crop Horizontal** does the same thing, but sizes vertically and crops horizontally.

If you select the **No Resize** resize method, the Width and Height fields are ignored and the output images are the same size as the input images.

Aspect Ratio

Aspect ratio is the ratio of the width of an image to its height.

If your input image is 100 x 200 pixels, its aspect ratio is 100/200, or .5 .

Changing an image’s aspect ratio causes a stretching or compressing of the image in one or both dimensions.

Wizards

Currently, ThumbNailer offers 4 wizards:

Basic Thumbnails – Create thumbnail images.
Advanced Thumbnails – Create thumbnails with an extended set of options.
Basic Web Pages – Create thumbnails and basic web pages to display them in.
Copy images to a new Format –change the image format, without modification.

Each of these wizards can be used to quickly set up the proper options for the particular operation. They are designed so that you can get good results with a minimum amount of work. In each case, you can go through the wizard, then press Start and get good results; or you can use the wizards to set up basic options that you can fine tune using the Advanced settings pages.

We will add more wizards as time allows.

Source Folder / File Edit

This is where you specify the root folder from which input images will be read. This path must exist.

You can drag a folder from Explorer or the Desktop into this control.

Use the Folder button to browse for folders.

You can also enter the name of a single file in this control. This will allow you to process this file. The resulting image file will be placed in the output folder.

Note:
It is recommended that you do **not** use network paths for folders. ie, server\\network\folder\. You should use physical drive paths whenever possible: H:\network\folder\.

Note:
If you are making HTML pages and wish to use custom output folders for the HTML pages: this folder, the destination folder, the thumbnail HTML folder and the source HTML folder should all be on the same drive. If these folders are not all on the same drive, Thumbnailer will be unable to generate relative paths for the image references and hyperlinks – the HTML pages won’t work.

Destination Folder Edit

This is where you specify the root directory for all output images to be placed.

You can drag a folder from Explorer or the Desktop into this control.

Use the Folder button to look for folders.

Note: It is recommended that you do **not** use network paths for folders. ie, server\\network\folder\. You should use physical drive paths whenever possible: H:\network\folder\.

Note:
If you are making HTML pages and wish to use custom output folders for the HTML pages: this folder, the source image folder, the thumbnail HTML folder and the source HTML folder should all be on the same drive. If these folders are not all on the same drive, Thumbnailer will be unable to generate relative paths for the image references and hyperlinks – the HTML pages won't work.

Resize Method

ThumbNailer provides a variety of resizing options. Most of these methods are included because people have requested them.

No Resize

Output images are written with the same dimensions as the input images.

Best Fit

Output images are sized such that the final size is the largest that will fit in the dimensions specified in the Width and Height fields, while maintaining the source image’s aspect ratio.

Fixed Width

Output images are sized such that the width is exactly as specified in the Width field, while maintaining the source image’s aspect ratio.

Fixed Height

Output images are sized such that the height is exactly as specified in the Height field, while maintaining the source image’s aspect ratio.

Exact Size

Output images are sized such that the width and height are exactly as specified in the Width and Height fields. This will usually change the image’s aspect ratio.

DPI Change

Some image formats allow the writer to specify a Dots Per Inch value. This value, along with the number of pixels in the image, determines the physical size of the on-screen or printed image (size = DPI / pixels). This resize method will change the DPI to the value you give, and will adjust the number of pixels in the image so that the image output size remains constant. ThumbNailer allows you to set a Default output DPI value for image formats that do not allow for a DPI value.

Pixel Area

Output images are sized so that the product of the image’s width and height are equal to the area specified in the Pixel Area field. The image’s aspect ratio is maintained.

Percent

Output images are resized by the specified percentage.

Fixed Width, Crop Vertical

Output images are first sized to fit the

	output width, while maintaining the source image’s aspect ratio, any excess is cropped from the top and/or bottom of the image. You can choose where the crop happens on the Advanced / Image page.
Fixed Height, Crop Horizontal	Output images are first sized to fit the output height, while maintaining the source image’s aspect ratio, any excess is cropped from the left and/or right of the image. You can choose where the crop happens on the Advanced / Image page.

Note that images are automatically enlarged if necessary, when using the Percent, Exact Fit or Pixel Area resize methods. For all other methods, the images will not be expanded to fit, unless you check the Allow Enlarge button.

Output Format

ThumbNailer supports the following file formats for output:

24 bit, same as input	Output images will have the same format as the input images, but in 24-bit mode. *
8 bit, same as input	Output images will have the same format as the input images, but in 8-bit mode. JPG has no 8-bit mode; JPGs will be written as 24-bit JPGs. *
8 bit grayscale, same as input	Output images will have the same format as the input images, but in 8-bit grayscale mode. JPG has no 8-bit mode; JPGs will be written as JPGs.*
JPG	Output images will be in the JPG format
GIF 8-bit	Output images will be in the GIF format, 8-bits per pixel (256 colors)
GIF 4-bit	Output images will be in the GIF format, 4-bits per pixel (16 colors)
GIF grayscale	Output images will be in the GIF format, 8-bits per pixel (256 colors), grayscale.
BMP 24-bit	Output images will be BMP 24-bit, non compressed
BMP 16-bit	Output images will be BMP 16-bit, non compressed
BMP 8-bit	Output images will be BMP 8-bit, non compressed (256 colors)
BMP 1-bit black and white**	Output images will be BMP 1-bit, black and white
BMP 8 bit RLE	Output image will be BMP 8-bit, Run Length Encoded. (256 colors)
BMP 4-bit	Output images will be BMP 4-bit, non compressed (16 colors)
BMP 8-bit grayscale	Output images will be BMP 8-bit grayscale, non compressed (256 colors)
PNG 24 bit	Output images will be PNG 24-bit
PNG 32 bit	Output images will be PNG 32 bit. This allows for an alpha (transparency) channel if you are doing image fades.
PNG 1-bit black and white**	Output images will be PNG 1-bit (black and white)
PNG 8 bit	Output images will be PNG 8-bit (256 colors)
PNG 8 bit grayscale	Output images will be PNG 8-bit grayscale
PSD 24 bit	Output image to PSD, 24-bit (Adobe Photoshop)
PSD 8-bit	Output images will be 8-bit PSD (Photoshop) (256 colors)
PSD 8 bit grayscale	Output images will be 8-bit grayscale PSD (256 colors)
TIFF 24 bit	Output images will be 24-bit TIFF
TIFF 32 bit	Output images as 32-bit TIFF. This allows for an alpha (transparency) channel, if you are doing image fades.
TIFF 8 bit	Output images will be 8-bit TIFF (256 colors)
TIFF 1 bit black and white**	Output images will be 1-bit TIFF (black and white)
TIFF 8 bit grayscale	Output images will be 8-bit grayscale TIFF (256 colors)
PCX 24 bit	Output images will be PCX 24-bit
PCX 8-bit	Output images will be PCX 8-bit (256 colors)
PCX grayscale	Output images will be PCX 8-bit with a gray palette (256 colors)
TGA 24 bit	Output Images will be TGA, 24 bit
TGA 8-bit	Output Images will be TGA, 8-bit (256 colors)
TGA grayscale	Output Images will be TGA, 8-bit grayscale
Raw (all)	Output will be as ASCII text
WMF 24-bit	Output as 24-bit Windows Metafile
EMF 24-bit	Output as 24-bit Enhanced Metafile. Used by MSOffice applications
JPEG-2000	Output images are 24-bit JPEG-2000
WBMP (Wireless Bitmap)	Output images are 1-bit WBMP.

* - If ThumbNailer does not output files of the same type as input (PhotoCD, AVI and TTF for example), ThumbNailer will use the Default output format.

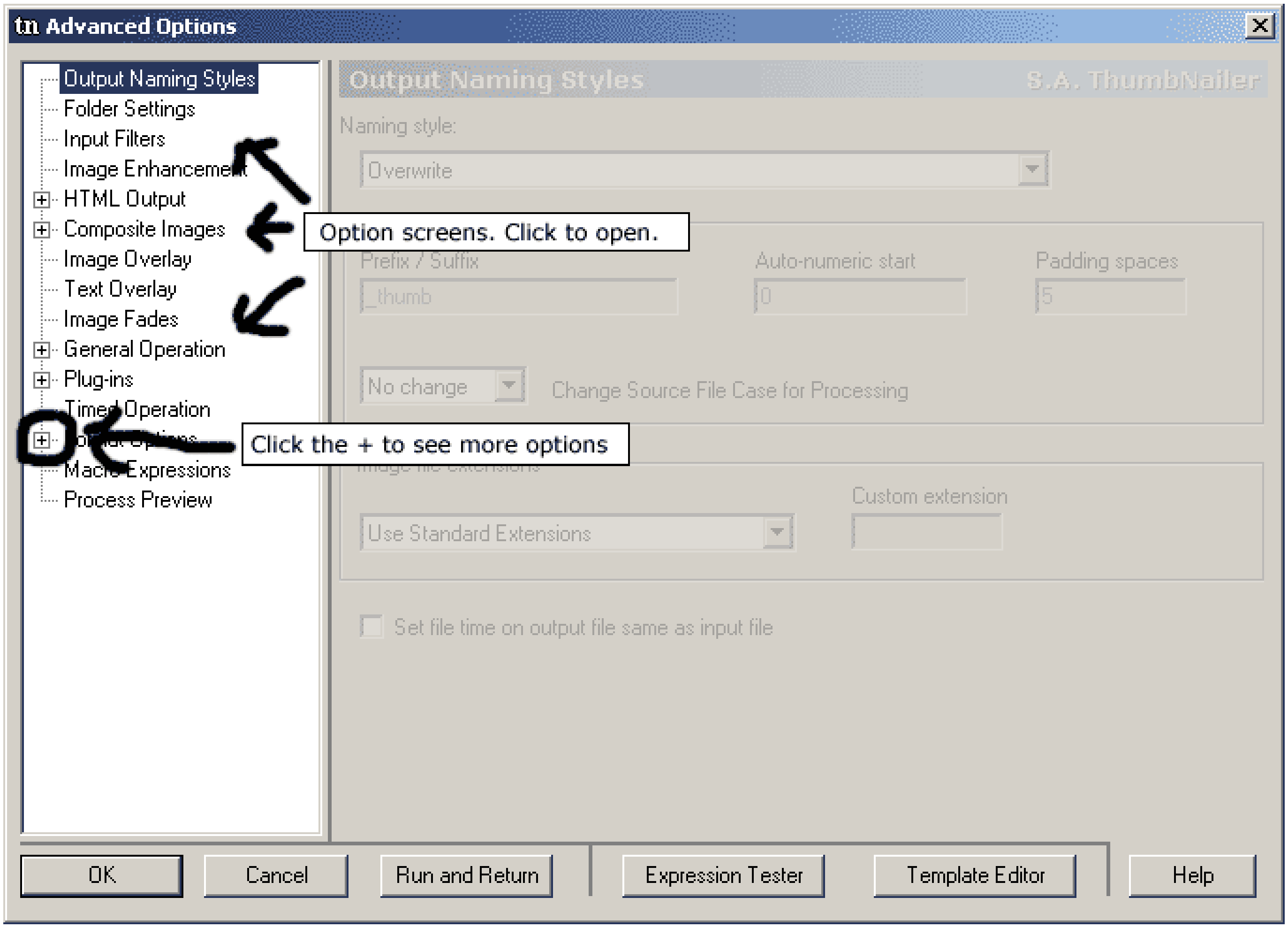
** - Images are converted to black and white using a brightness 50% threshold. You can sometimes get better results from photographic images, if you use the TNAutoBright plugin to stretch the brightness levels.

Start

Pressing Start will cause the program to begin processing files. A progress dialog will be presented, and the status bars at the bottom of the main window will display the actions that it is performing.

All options you have set will be applied to all output images.

Advanced



Option screens. Click to open.

Click the + to see more options

Naming style: Overwrite

Prefix / Suffix: _thumb

Auto-numeric start: 0

Padding spaces: 5

No change — Change Source File Case for Processing

Image file extensions: Use Standard Extensions

Custom extension:

☐ Set file time on output file same as input file

This is how you get to the hundreds of advanced options that Smaller Animals ThumbNailer provides.

Output Naming Styles
Folder Options
Input Filters
Operation
Image Options
Composites
Overlay Page
HTML Options
Fades
Registration
Text Overlay
Buttons

OK
Accept the changes and return to the main ThumbNailer screen.

Cancel
Discard any changes and return to the main ThumbNailer screen

Expression TesterLaunch the Expression Tester

Template Editor Launch the Template Editor

Run and Return
This is equivalent to pressing OK, then Start, then Advanced. This can be very useful if you’re editing your settings and want to run the batch to see what your changes look like, then change your settings and see what the new ones look like, etc. but get sick of pressing OK, Start, Advanced.

Help
Show help for the current options page.

Single File Operation

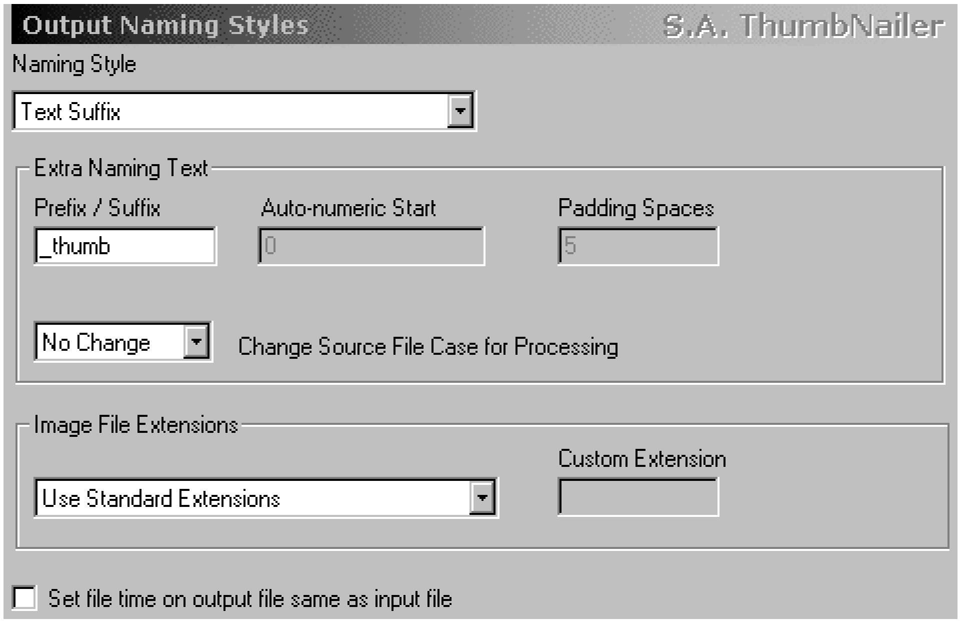
You can use S.A. ThumbNailer to process images one at a time.

Place the name of the file you want processed in the Input Folder / File control. You may also drag files from the desktop onto this control.

Press Start and the program will process the file using all of the options you have set. Output images are written to the Output Folder.

Note: Log files (both text and HTML) and composites are not written when processing single files.

Output Naming Styles



Output Naming Styles — S.A. ThumbNailer

Naming Style: Text Suffix

Extra Naming Text

Prefix / Suffix: _thumb

Auto-numeric Start: 0

Padding Spaces: 5

No Change — Change Source File Case for Processing

Image File Extensions

Use Standard Extensions

Custom Extension:

□ Set file time on output file same as input file

(Advanced / Output Naming Styles)

Output Naming Style
Output files normally retain the name of the input file. You can choose to have S.A. ThumbNailer create new names using one of the methods described below:

Overwrite
The program writes the output file with the same name as the input file. Existing files are overwritten. This is dangerous; you can overwrite valuable files this way!

Auto Numeric Suffix
The program insures that a file with the output name doesn't already exist in the output directory. If one does exist, S.A. ThumbNailer changes the output name by appending a "1" to the end of the file name. If there is a file that matches this name, the "1" becomes a "2", and so on.

Note *: do not use this style when doing multiple frame extraction from AVIs or multi-page TIFFs. The frame extractor will generate its own numeric suffixes. Choosing this option will cause unpredictable results.*

Text Prefix
The prefix/suffix text is added to the beginning of the output file name.

Text Suffix
The prefix/suffix text is added to the end of the output file name.

Text Prefix + Auto Numeric Suffix
This allows you to create a list of thumbnails with names that are created by adding a number to your text prefix text. The files will be named like this :
File1.jpg
File2.jpg
File3.jpg
and so on.

Auto-numeric (zero-padded)

The program insures that a file with the output name doesn't already exist in the output directory. If one does exist, the program changes the output name by appending a "00000" to the end of the file name. If there is a file that matches this name, the "00000" becomes a "00001", and so on.

Bob.jpg -> Bob00000.jpg

***Note** : do not use this style when doing multiple frame extraction from AVIs, or multi-page TIFFs. The frame extractor will generate its own numeric suffixes. Choosing this option will cause unpredictable results.*

Prefix Plus Auto numeric (zero-padded)
This allows you to create a list of thumbnails with names that are created by adding a zero-padded number to your text prefix text. The files will be named like this :
File00000.jpg
File00001.jpg
File00002.jpg
…and so on.

Base Folder + Auto Numeric Suffix
This is similar to the **Prefix Plus Auto Numeric** style, except that the prefix used is the name of the image's parent folder.

Base Folder + Auto Numeric Suffix (zero-padded)
This is similar to the **Prefix Plus Auto numeric (zero-padded)** style, except that the prefix used is the name of the image's parent folder.

Base Folder + Text + Auto Numeric Suffix
The output file name is the name of the image's parent folder plus the text from the prefix/suffix field plus an auto-numeric suffix.

Base Folder + Text + Auto Numeric Suffix
Similar to the above, but the number is zero-padded.

Prefix / Suffix
Enter your prefix or suffix here.

AutoNumeric Start
Enter the number at which the auto-numeric suffixes will start.

Zero-padding spaces
Enter the maximum number of '0's to be used for zero-padding.

Case

No Change
Leave upper/lower case as set in input filename

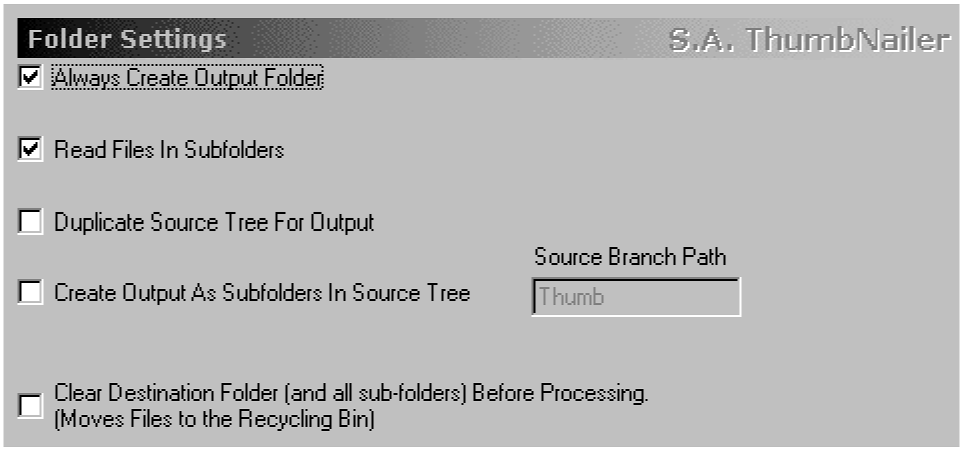
Lower
Force output filename to lower case

Upper
Force output filename to upper case

Set File Time on Output File Same as Input File

This will change the creation, modification and access dates on the output file to match the modification date of the input file.

Folder Settings



Folder Settings

S.A. ThumbNailer

☑ Always Create Output Folder

☑ Read Files In Subfolders

☐ Duplicate Source Tree For Output

☐ Create Output As Subfolders In Source Tree

Source Branch Path: Thumb

☐ Clear Destination Folder (and all sub-folders) Before Processing.
(Moves Files to the Recycling Bin)

(Advanced / Folder Settings)

This is where you control how ThumbNailer spans folders on input and creates folders on output.

Always create output folder
The program will create the specified output directory. This may include any number of subdirectories.

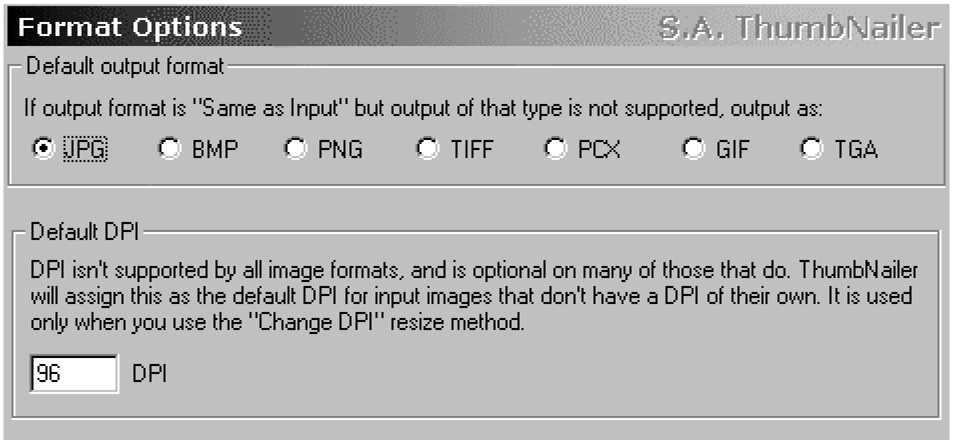
Read files in subfolders
The program will process all image files in all subdirectories of the specified input directory.

Duplicate source tree for output
If the input directory contains a subdirectory, the program creates a corresponding subdirectory below the output directory. All subdirectories of the input directory will be duplicated and all files within those subdirectories will be processed.

Create Output as Subfolders in Source Tree + Source Branch Path
This allows you to put all thumbnails into subfolders off of the folder where the original images reside. The program will create a subfolder with the specified name in the Input folder. The Output folder is ignored. This can be used in conjunction with the Read Files in Subfolders option to create a thumbnail subfolder in each of the source folders.

Clear Destination Folder Before Processing
The program will move all files and subfolders in the destination folder to the Recycle Bin before starting to process.

Format Options


Format Options

S.A. ThumbNailer

Default output format

If output format is "Same as Input" but output of that type is not supported, output as:

● JPG ○ BMP ○ PNG ○ TIFF ○ PCX ○ GIF ○ TGA

Default DPI

DPI isn't supported by all image formats, and is optional on many of those that do. ThumbNailer will assign this as the default DPI for input images that don't have a DPI of their own. It is used only when you use the "Change DPI" resize method.

96 DPI

(Advanced / Format Options)

Default Output Format

When you’ve selected any of the “Same as Input” options for the output format, and the input image format is of a type that ThumbNailer doesn’t write (such as AVI or TTF), output files will be written with this format.

Image Format Options

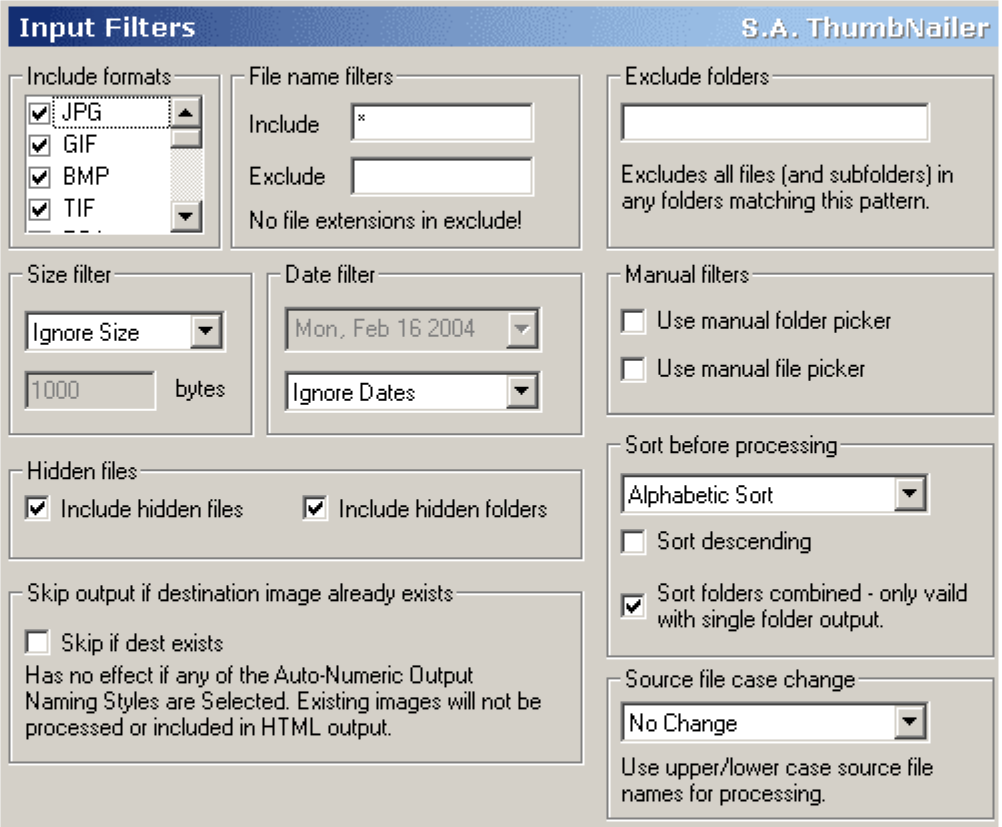
WMF	Metafile options
JPG	JPG Options
PNG	PNG Options
AVI	AVI Options
Raw	Raw / ASCII Options
GIF	GIF Options
TIFF	TIFF Options
TFF	TTF Options

Default DPI

Some image formats allow the writer to specify a DPI (Dots Per Inch) value. This value, along with the number of pixels in the image, can be used to determine the physical size of the image, when printed or shown on-screen. DPI works by telling the printer that each inch (or centimeter) in the image will hold a certain number of pixels: Ex. If the image is 300 x 300 pixels, and the DPI value is 100, the image will be 3x3 inches when printed (300dots / 100 dots/inch = 3 inches).

When you use the Change DPI resizing method, ThumbNailer will use the value you provide here for images that do not have a DPI value of their own.   Currently, ThumbNailer will only read DPI from JPG, PNG, TIFF and BMP images; images from all other formats will be assigned this default DPI.

Input Filters


Input Filters S.A. ThumbNailer

Include formats
☑ JPG
☑ GIF
☑ BMP
☑ TIF

File name filters
Include *
Exclude
No file extensions in exclude!

Exclude folders
Excludes all files (and subfolders) in any folders matching this pattern.

Size filter
Ignore Size
1000 bytes

Date filter
Mon, Feb 16 2004
Ignore Dates

Manual filters
□ Use manual folder picker
□ Use manual file picker

Hidden files
☑ Include hidden files
☑ Include hidden folders

Sort before processing
Alphabetic Sort
□ Sort descending
☑ Sort folders combined - only vaild with single folder output.

Skip output if destination image already exists
□ Skip if dest exists
Has no effect if any of the Auto-Numeric Output Naming Styles are Selected. Existing images will not be processed or included in HTML output.

Source file case change
No Change
Use upper/lower case source file names for processing.

(Advanced / Input Filters)

The program can filter, for or against, many different file attributes. Note that it is very easy to set the filters so that no files are included.

The filters are processed in this order:

1. Format
1. File Name Include
1. File Name Exclude
1. Date
1. Size

Include Formats

BMP, GIF, JPG, PSD, TIFF, PNG, PCX, TGA, AVI, WMF, EMF
Check the file formats you wish to use for input.

File Name Filters

Note **:** Do not use file extensions in these fields.

Include
Enter a filename-matching pattern. This will cause the program to include only those files that match your pattern. Do not enter file extensions here.

Exclude

Enter a filename-matching pattern. This will cause the program to ignore those files that match your pattern. The file extension is included as part of the filename.

The rules below apply only to the Exclude filter (not to the Include filter):

This pattern can contain any combination of valid filename characters and the special pattern matching characters: * and ? .

A "?" matches a single character:

"?-foo.jpg" matches "a-foo.jpg", "b-foo.jpg", "c-foo.jpg", etc..

A "*" matches an unlimited number of characters:

"*foo.jpg" matches any filename that ends with "foo.jpg". ex. : "bunnyfoofoo.jpg".

"foo*.jpg" matches any filename that starts with "foo" and ends with ".jpg". ex. "football.jpg"

Multiple matching patterns:
You can match multiple patterns by separating them with a semicolon:

Ex.: "Foo*.jpg;*foo.jpg" will exclude all filenames that begin with "foo" and all filenames that end with "foo.jpg".

Note : "*" will cause all files to be excluded.

Exclude Folders

Exclude Folders
Enter a folder name. If the program finds a folder matching this pattern as it scans for files, it will ignore this folder (and all subfolders of this folder). See the filename exclude filter above for matching rules.

Size Filter

Enter a size, in ***bytes***, of the files you want the program to filter.

Ignore Size
The program will ignore the file size.

Include Smaller
The program will only process files that are smaller than the file size limit you have specified.

Include Larger
The program will only process files that are larger than the file size limit you have specified.

Date Filter

You can choose to have S.A. ThumbNailer process only files that were last modified before or after the date you specify.

Date
Enter the date here in dd/mm/yyyy format.

Include After Last Run
Only include files which have a modification time after the time that the last ThumbNailer batch completed.

Manual Filters

These tools help you refine the files and folders that ThumbNailer will process during a batch of files.

Use Manual Folder Picker
See Manual Folder Picker

Use Manual File Picker
See Manual File Picker

Sort Before Processing

No Sort
Images will be processed in the same (apparently random) order that they are listed in when you type DIR from a DOS prompt. If you really want random, though, use the "Random" sort method.

Alphabetic
This causes the program to perform an alphabetic sort on the input file names. This will insure that the files are processed in alphabetical order. Note : alphabetic order means that the name "pic10.jpg" comes before "pic2.jpg"; if you need to sort names ending with a trailing numeric field, see the Alpha-Num method.

Date
This causes the program to perform a date sort on the input file names. The newest files will be processed first.

EXIF Date
Some files, especially files taken with digital cameras, contain additional information about the image, including the date when the picture was taken. ThumbNailer can extract this date information from these images and use it to sort them. The newest files will be processed first.

Alpha-Num
This is similar to the Alphabetic method, but with a twist : names ending with a numeric sequence ("pic01.jpg" ,   "image_200.pcx", etc.) are sorted so that the numeric sequences are treated as numbers, not as alphabetic strings.
Ex. with the Alphabetic sorting method, "pic10.jpg" will be placed before "pic2.jpg", because the "10" and "2" are treated as sequences of characters, not as numeric values. But, using the Alpha-Num method, "pic2.jpg" comes before "pic10.jpg".

Note: filenames are also sorted by file extension, so "pic10.bmp" comes before "pic2.jpg".
Note: this only works when the numeric sequence is at the end of a file name.

Widest First
Images that are wider than they are tall will appear first. Images that are taller than they are wide will appear last.

Random
Files names are thorough shuffled.

Descending
This will change the sort order to descending. Z..A, older dates first, etc..

Sort Folders Combined
Normally, ThumbNailer sorts files against other files in their source folder. But, if you want to put all of your thumbnails and HTML pages in a single output folder, it might makes more sense to sort all the files together. This option allows you to do that – it gets all the input filenames, and sorts them by whatever criteria you specify, as a single group, rather than sorting all the files from one source folder, then the next source folder, then the next, etc.

Note that this only applies when you are outputting to a single output folder. That means you cannot have the "Duplicate Source Tree" or the "Create output as subfolders" options set on the Folder Settings- both of those options cause TN to create multiple output folders.

Include Hidden Files/Folders

If these buttons are not checked, ThumbNailer will ignore hidden files and folders.

Skip Output if Destination Exists

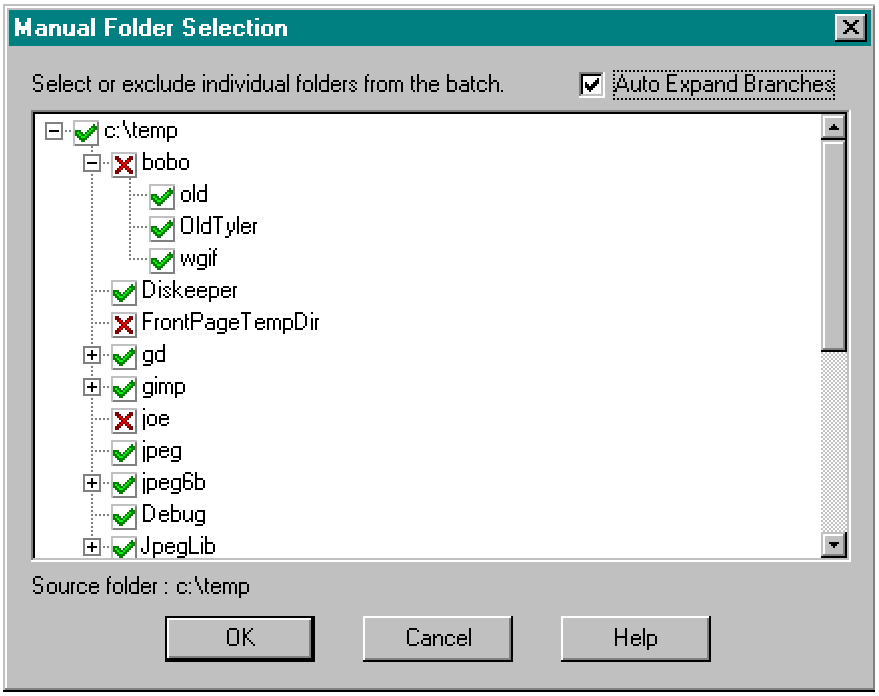
This will cause the program to skip the output step if the destination file exists. Note that this has no effect if you are using one the Auto-Numeric naming methods.

Source Case Change

If you choose Upper Case or Lower Case, the program will use the upper or lower case versions of the source file names for all further processing. This is useful, for example, when you want to force the source file links in the HTML pages to all lower case because of server restrictions (Windows is not case sensitive for filenames, Unix is).

Note: This only changes how the program uses the source file name internally; it will not modify the names of your source files.

Manual Folder Picker


Manual Folder Selection

Select or exclude individual folders from the batch.

☑ Auto Expand Branches

```
⊟ ✓ c:\temp
   ⊟ ✗ bobo
         ✓ old
         ✓ OldTyler
         ✓ wgif
     ✓ Diskeeper
     ✗ FrontPageTempDir
   ⊞ ✓ gd
   ⊞ ✓ gimp
     ✗ joe
     ✓ jpeg
   ⊞ ✓ jpeg6b
     ✓ Debug
   ⊞ ✓ JpegLib
```

Source folder : c:\temp

(Advanced / Input Filters / Manual Filters)

When you press Start, if you have the **Read Files In Subfolders** (Folder Options) ThumbNailer will look in the source folder for any subfolders that it may contain. Then it will look into each of those subfolders for any folder they may contain, and so on. The result is that ThumbNailer creates a list of the folders contained within the source folder.

The Manual Folder Picker allows you to view this list of folders and select folders that you wish to exclude. A folder with a green check mark is one that will be processed; a folder with a red "X" is one that will not be processed. Click on the check mark or X to change it to the other.

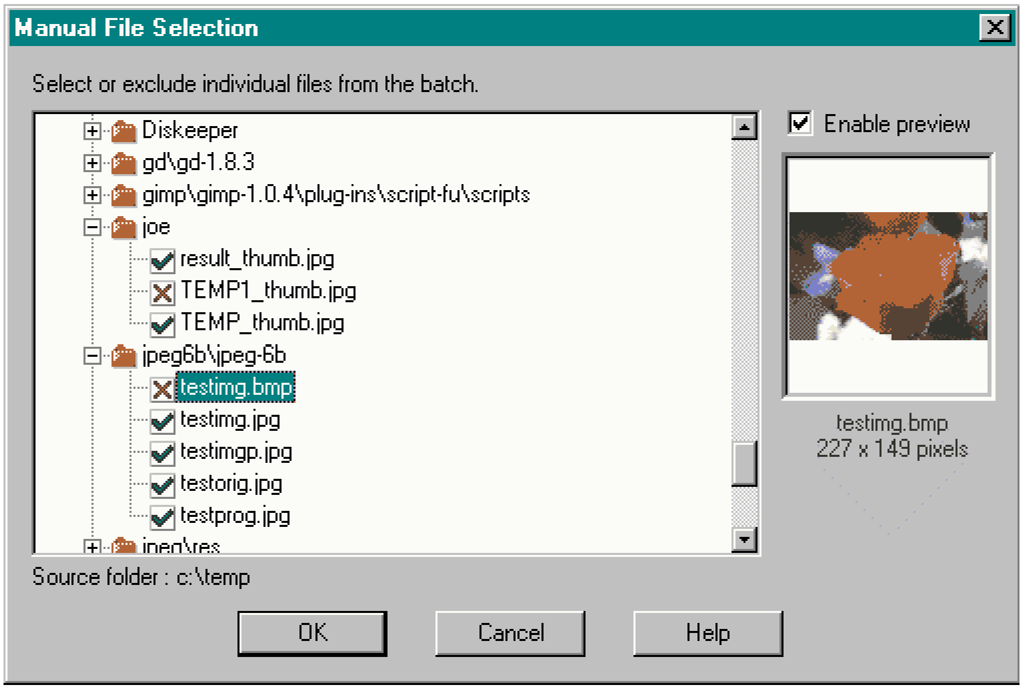
Folders are shown in a "tree" fashion, just like they're shown in Explorer.

When you press OK, ThumbNailer will search only the folders that you have selected.

***Note**:* selecting or de-selecting a folder does not change the settings for any folders contained within that folder. If you want to select or deselect all folders within a folder, you can right-click on the folder and choose the appropriate option from the menu that will appear.

Note: If you have the **Read Files In Subfolders** option checked, this tool isn't very useful – as it only shows the single source folder.

Manual File Picker



Manual File Selection

Select or exclude individual files from the batch.

```
⊞ Diskeeper
⊞ gd\gd-1.8.3
⊞ gimp\gimp-1.0.4\plug-ins\script-fu\scripts
⊟ joe
    ✓ result_thumb.jpg
    ☒ TEMP1_thumb.jpg
    ✓ TEMP_thumb.jpg
⊟ jpeg6b\jpeg-6b
    ☒ testimg.bmp
    ✓ testimg.jpg
    ✓ testimgp.jpg
    ✓ testorig.jpg
    ✓ testprog.jpg
⊞ jpeg\res
```

☑ Enable preview

testimg.bmp
227 x 149 pixels

Source folder : c:\temp

(Advanced / Input Filters / Manual Filters)

When you press Start, ThumbNailer builds a list of all of the folders contained in the source folder. (See Manual Folder Picker). Then ThumbNailer examines each folder within that list, to build a list of the files it will process in the batch.

The Manual File Picker allows you to view this list of files and select files that you wish to exclude.

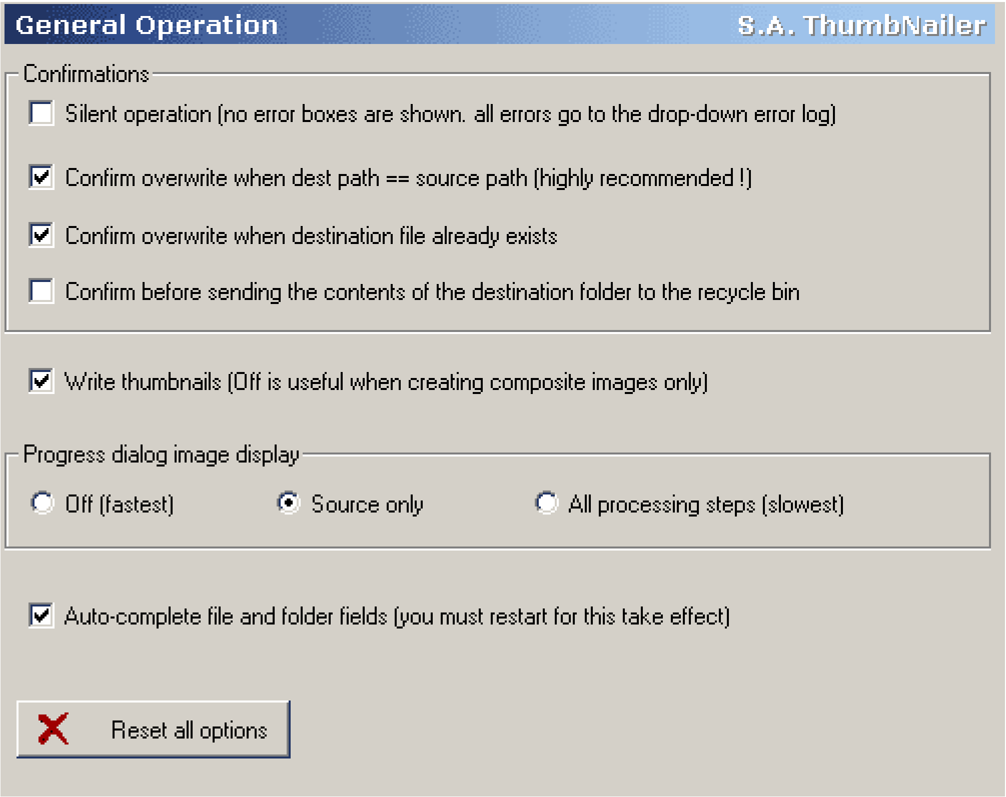
The files are displayed by folder. If there are no files within a folder, even if the folder was shown in the Manual Folder Picker, the folder will not be shown.

A file with a green check mark is one that will be processed; a file with a red “X” is one that will not be processed. Click on the check mark or X to change it to the other.

The small preview window on the right side shows the image that is currently selected in the tree. You can turn this off, if you want, by un-checking the Enable Preview box.

Note: if you want to select or deselect all files within a folder, you can right-click on that folder and choose the appropriate option from the menu that appears.

General Operations



Confirmations

☐ Silent operation (no error boxes are shown. all errors go to the drop-down error log)

☑ Confirm overwrite when dest path == source path (highly recommended !)

☑ Confirm overwrite when destination file already exists

☐ Confirm before sending the contents of the destination folder to the recycle bin

☑ Write thumbnails (Off is useful when creating composite images only)

Progress dialog image display

○ Off (fastest) ● Source only ○ All processing steps (slowest)

☑ Auto-complete file and folder fields (you must restart for this take effect)

(Advanced / General Operation)

Confirmations

Silent Operation
No non-critical message boxes will be launched once Start is pressed. Errors will be reported in the error log at the bottom of the main window. These errors include invalid image files, processing errors, etc..

Confirm overwrite when dest path = source path
If this is on, the program will prompt you if the destination file is the same as the output file. This prevents accidental overwriting of your source files. It is highly recommended that you leave this on. If you turn this off, you may overwrite your original images!

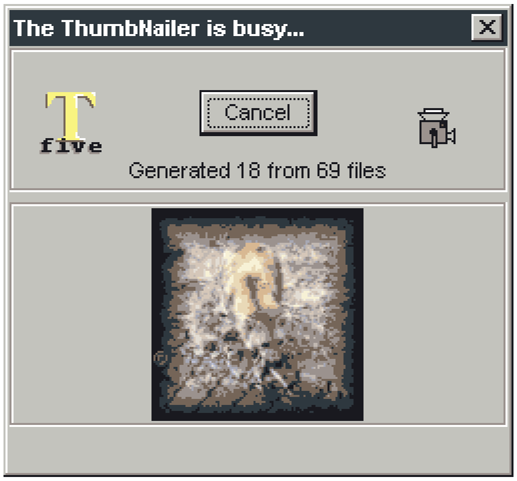
Confirm overwrite when destination file already exists
If this is checked, the program will stop to warn you that the destination file (which is about to be written) already exists in the output folder. If this happens, and you click “OK” at the prompt, ThumbNailer will overwrite the file.

Write Thumbnails
You can choose to have the program not write the final thumbnails. This can be useful if all you want is to create composite images. If you create HTML files with this turned off, however, the program will generate files that assume the thumbnails *have been written*.

Reset all options
All options will be reset to the default values.

Progress Dialog Image Display

You can choose to have the program display the image it’s currently working on. Note that this will slow down the thumbnail creation, sometimes, considerably.



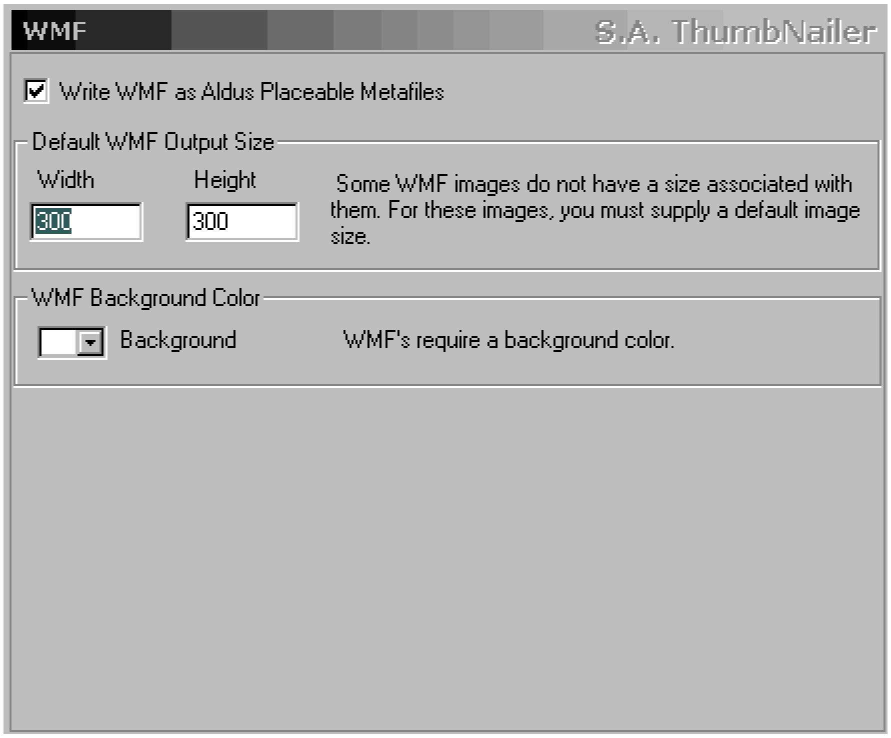
Generated 18 from 69 files

Off (fastest)
Turn image display off.

Source Only
Display the source image only.

All Processing Steps (slowest)
This shows all image processing steps. Some steps are very fast and you may not see them go by. Because this can cause six resize and display operations per file, this is definitely not something you’ll want to use on long jobs.

WMF Options



☑ Write WMF as Aldus Placeable Metafiles

Default WMF Output Size

Width: 300

Height: 300

Some WMF images do not have a size associated with them. For these images, you must supply a default image size.

WMF Background Color

Background

WMF's require a background color.

(Advanced / Format Options / MetaFiles)

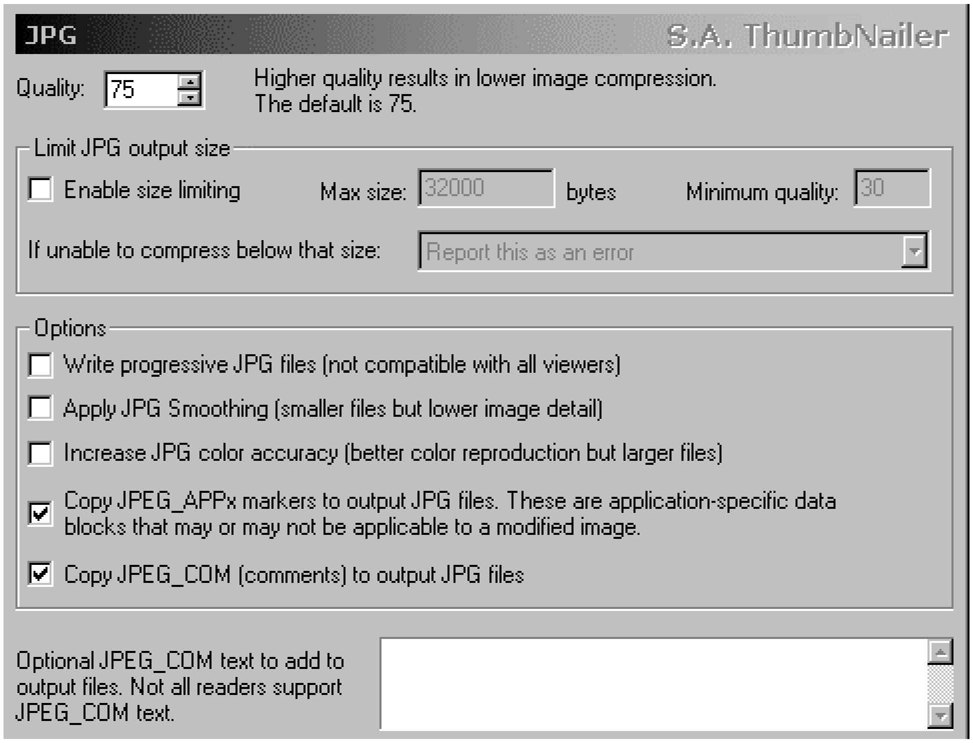
The program supports the input of Windows Metafiles. WMF's are unique among the supported formats in that they are not actually rectangular sets of pixels, but rather are sets of Window graphics drawing instructions. For this reason, the program only supports WMF input.

Write WMF as Aldus Placeable Metafiles
A "placeable" metafile is one that contains size and positioning information. This is one of many flavors of metafiles. Not all readers will read these. Not all readers that do read these will read them correctly, but this is true of all metafile formats.

Default WMF Output Size
Some WMF's do not have a size associated with them : they take the size of the space they are drawn into. For these files, a default size must be assumed, so that they can be used in the Percent, Pixel Area and No Resize methods. This is also the size used to read WMF's for overlay and fade mask application.

WMF Background color.
Because WMF's may have transparent areas, you must choose a color on which they are drawn.

JPG Options



Quality: 75

Higher quality results in lower image compression.
The default is 75.

Limit JPG output size

□ Enable size limiting   Max size: 32000 bytes   Minimum quality: 30

If unable to compress below that size: Report this as an error

Options

□ Write progressive JPG files (not compatible with all viewers)

□ Apply JPG Smoothing (smaller files but lower image detail)

□ Increase JPG color accuracy (better color reproduction but larger files)

☑ Copy JPEG_APPx markers to output JPG files. These are application-specific data blocks that may or may not be applicable to a modified image.

☑ Copy JPEG_COM (comments) to output JPG files

Optional JPEG_COM text to add to output files. Not all readers support JPEG_COM text.

(Advanced / Format Options / JPG)

Quality
Higher quality results in lower image compression. Default is 75.

Limit JPG Output size
Enable size limiting
If this is checked, ThumbNailer will attempt to limit the file size (in bytes, not image dimensions) of the JPG files it writes.

Note: This may cause JPG writes to take up to *ten* times longer than normal, as the program tries to find the optimal JPG quality settings.

Max Size
ThumbNailer will attempt to keep JPG files smaller than this limit.

Minimum quality
This is the minimum quality value ThumbNailer will try. This can be used as a safety measure; when the JPG quality is too low, the resulting image is highly degraded. Setting this value will keep the program from setting the quality too low.

If unable to compress below that size
If ThumbNailer is unable to produce a JPG file that is smaller than the maximum size you've specified, at a quality setting greater than or equal to the minimum quality setting you've specified, it will do one of the following:
Report this as an error –show an error message saying that the file could not be compressed to fit your settings.
Compress to smallest size possible – ThumbNailer will generate the smallest JPG it can.
Use the normal quality setting - ThumbNailer will use the standard JPG quality setting.

Options

Write Progressive JPGs

"Progressive" JPG is a flavor of JPG that most *smart* readers (most web browsers, for example) can display as it is being read (transferred across the web, for example). The image is displayed as a series of successively sharper images. This will not dramatically affect the output image size.

Smoothing

JPG smoothing can reduce file size. The resulting images will appear somewhat blurred, but the effect is slightly less objectionable than what you get by lowering the JPG image quality.

Increase JPG color accuracy

This uses a higher sampling frequency to achieve better color reproduction. As usual, higher quality means larger file sizes.

Copy JPEG_APPx markers

A JPEG_APPx marker is a chunk of data that is put into a JPG file by the application that wrote it. The format and contents of the data are almost always specific to that application. The data may contain timestamps or exposure settings, for example, if the JPG was made with a digital camera. If you check this options, ThumbNailer will copy the data from an input JPG to the output JPG. Note that if the data contains exposure information, copying it to a resized, matted and shadowed image may be misleading to any applications that rely on the data to provide useful information about the image. Caveat emptor.

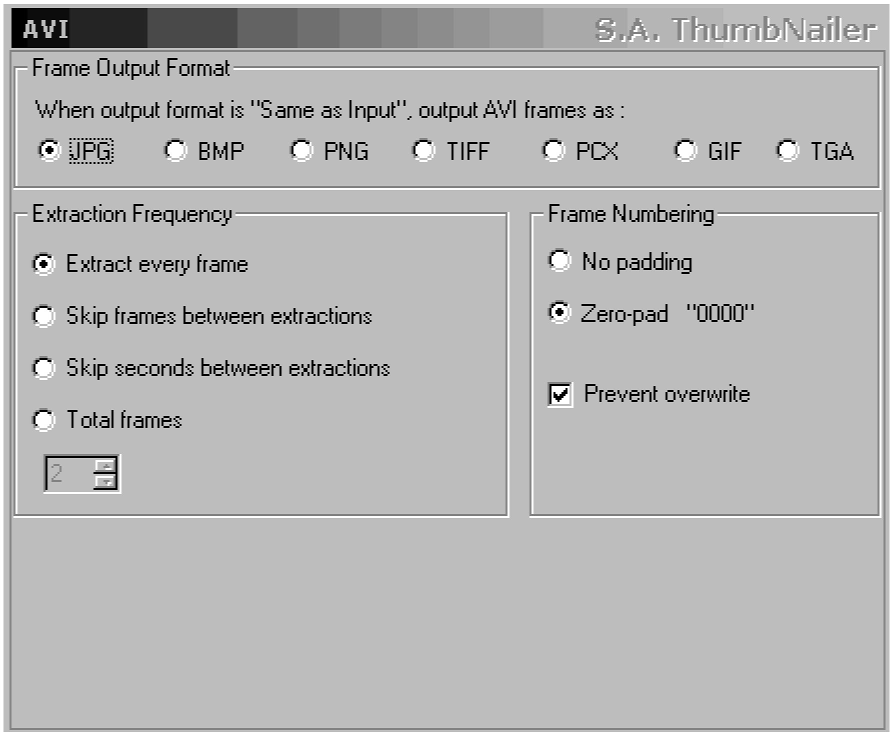
Copy JPEG_COM (comments) to output files

JPG files can contain comment text. If you check this, all comment text from the input files will be written to the output files. Not all JPG readers or writers will recognize or pay attention to this text.

Additional JPEG_COM text

You can specify your own comment text to be written to the output JPGs. This must be plain ASCII text.

AVI Options



Frame Output Format

When output format is "Same as Input", output AVI frames as :

● JPG ○ BMP ○ PNG ○ TIFF ○ PCX ○ GIF ○ TGA

Extraction Frequency

● Extract every frame

○ Skip frames between extractions

○ Skip seconds between extractions

○ Total frames

2

Frame Numbering

○ No padding

● Zero-pad "0000"

☑ Prevent overwrite

(Advanced / Format Options / AVI)

For every frame in an AVI file, the program will output one destination file, consecutively numbered.

Note:
By default, ThumbNailer does not process AVI files. If you want to process AVI files, you must enable this in Advanced / Input Filters / Include Formats. Check the "AVI" option in the list of formats.

Extraction Frequency
Because AVI files can contains hundreds of frames, you may want to extract only a subset of the total frames. These controls allow you to specify how many frames to extract.

S.A. ThumbNailer always extracts frame 0 (the first frame) from the AVI file.

Extract every frame
Extracts all frames.

Skip frames between extractions
Enter the number of frames that you wish the program to skip between frames that it extracts. Ex. If you give a value of 4, the program will extract frame 0, skip 1,2,3 and 4, then it will extract frame 5, etc..

Skip seconds between frames
Enter the number of seconds (whole numbers only) that you wish the program to skip between frames. Ex. If you enter a 3, the program will extract frame 0, then it will extract the frames that occur at times 3 sec, 6 sec, 9 sec, etc..

Total Frames (start with first)
Enter the total number of frames that you wish the program to extract from the AVI. The program will extract this number of frames, evenly-spaced throughout the file, starting with the first frame of the file. If you extract one frame, it will be the first frame; if you extract 3, you will get the first frame, the

frame at 33% and the frame at 66%.

Total Frames (centered)
Enter the total number of frames that you wish the program to extract. The program will extract this number of frames, evenly-spaced throughout the file. Frames will be selected based on the center: if you extract one frame, it will be the middle frame. If you extract 3, you will get frames at 25%, 50% and 75%.

Frame Numbering

All extracted AVI frames are named in this manner:

1. The base name from the AVI is used as the base name for the frame. Ex. the base of Bear.AVI is “Bear”.
1. The renaming options from the File / Dir tab are applied (prefix , suffix, numeric suffix). Ex. “BearThumb”
1. The AVI frame number is added to the end of the name : Ex. BearThumb0002
1. The output file format extension is added. Ex. BearThumb0002.JPG

***Note** : For AVI frame extraction, it is recommended that you do **not** use the Auto-Numeric Suffix renaming style from the File / Dir options tab. The Auto-Numeric Suffix option will cause unpredictable results when used with the numeric suffixes added in step 3, above.*

Zero-pad “0000”
The numbers added to the ends of the base AVI name are padded with at least three zeros. Ex. An AVI with three frames, Bear.AVI, will generate Bear0000.jpg, Bear0001.jpg and Bear0002.jpg. This implies a limit of 10,000 frames per base AVI.

No pad
No zeros are added to the output frame name. Ex. An AVI with three frames, Bear.AVI, will generate Bear0.jpg, Bear1.jpg and Bear2.jpg.

Prevent Overwrite
If this is checked, the program will re-number output frames so that they don’t overwrite any files that may already be in the output folder. This could lead to files with numbers at the end of their filename which are not their actual frame number. Internally, the program uses the Auto-Numeric Suffix generator to prevent output file overwriting.

If Prevent Overwrite is **not** checked, the numbers at the ends of frame filenames will be the same as their frame number. **Any files already in the output folder will be overwritten in case of a name collision.**

AVI Notes

Because of the way AVI frames are returned to the program from the operating system, large (> 1MB) AVI files will take considerably longer to process than smaller AVI files.

This slowdown is due to the way AVI files are composed: most frames are not stored as discrete images, but are stored as a set of compressed changes from key frames. “Key” frames are spaced throughout the video sequence. These are frames that *are* stored as discrete (whole) images.

Playing back video requires that a specified time scale is maintained between frames. Key frames create places for real-time decoders (video players, for example) to jump to if they can’t keep up with the decompression of the frames between key frames.
Ex. If your computer is decoding frames and finds that it is falling behind the clock, it will skip to the next key frame to make up time. A key frame is already a (compressed) whole image, so no further processing is required. This is the cause of the annoying jumping effect that you find when

playing large video sequences on slow PCs.

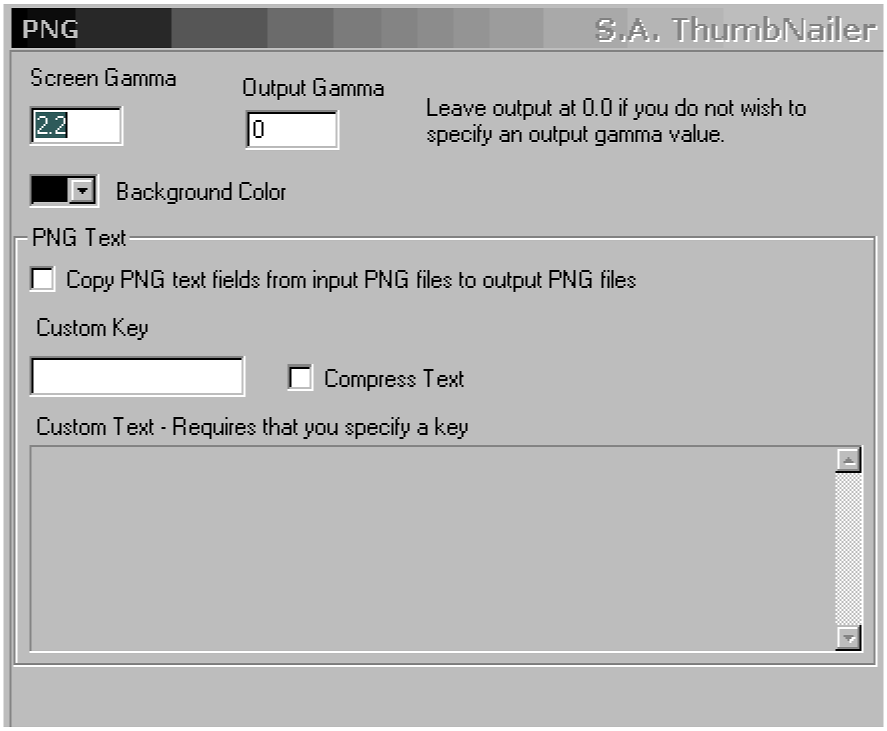
Decoding most frames requires that first the appropriate key frame is located and decompressed, then all of the following frames (which are really just incremental changes to the key frame) are decompressed and applied to the key frame image. Ex. Grabbing the third frame in a sequence requires that the first frame (the key frame) is decompressed. Then, applying the appropriate changes to the key frame creates the second frame. Then, applying the appropriate changes to the second frame creates the third frame.

Key frames take up a lot of space, and this is the reason there may be few of them. But if there are *too* few key frames in an image, decoders will spend a long time creating the in-between frames.

CODECs

Not all AVI files are created equal. There are many different ways to encode the video data in an AVI file and Windows doesn’t support all them right out of the box. For example, many digital cameras create movies that are encoded with what’s called “MJPEG” (Motion JPEG). Even though this is a common encoding type, Windows doesn’t know how to decode it. But, there are third party codecs (a codec is a helper program that can enCOde and DECode data) that can handle this task. Often, the digital camera itself will come with a codec that you can install on your Windows PC, to help with the decoding of the movies it makes. Check with your camera manufacturer, if you find that ThumbNailer can’t open the AVI files your camera makes. Or, contact us, we have a small list of codecs that we’ve tested and can recommend for certain cameras.

PNG Options



Screen Gamma 2.2

Output Gamma 0

Leave output at 0.0 if you do not wish to specify an output gamma value.

Background Color

PNG Text

□ Copy PNG text fields from input PNG files to output PNG files

Custom Key

□ Compress Text

Custom Text - Requires that you specify a key

(Advanced / Format Options / PNG)

Background Color
PNG files can be written with a specified background color. This color is blended with the image using the PNG alpha channel. The color is not specified in all PNG files. Setting this also allows you to specify a color to use in those PNG files for which the background color is not specified.

Screen Gamma
This is the gamma value of your monitor. 2.2 is a good guess for most PC monitors.

Output Gamma
If you want to specify an output gamma for your PNG images, do it here. If you leave the value at 0, no gamma is written to the file. If you want to specify a value, a value of 1 / ScreenGamma is recommended,   unless you know what you're doing.

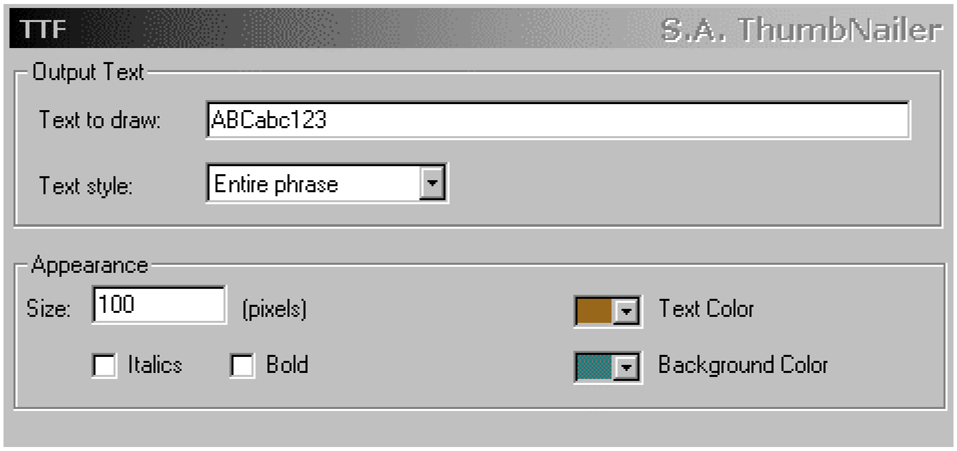
Copy PNG text from input
PNG files can have an arbitrary number of text fields in them. If you want to preserve the text in the input PNG files when writing the output PNG files, choose this.

Key
PNG text fields are composed of a key of 1-79 chars, and an optional text field. These must be printable characters! Note : without a key, you cannot have text.

Text
PNG text can be any size. Note : without a key, you cannot have text.

Compress
PNG text can be compressed in the file.

TTF Options



TTF S.A. ThumbNailer

Output Text

Text to draw: ABCabc123

Text style: Entire phrase

Appearance

Size: 100 (pixels)

☐ Italics ☐ Bold

Text Color

Background Color

(Advanced / Format Options / TTF)

ThumbNailer can generate thumbnails from a True Type font file. The font doesn’t have to be installed on your system; ThumbNailer uses the .TTF file directly.

Note:
By default, ThumbNailer does not process TTF files. If you want to process TTF files, you must enable this in Advanced / Input Filters / Include Formats. Check the “TTF” option in the list of formats.

Output Text

Text to Draw
ThumbNailer creates thumbnails by drawing this text using the current .TTF file.

Text Style
All characters, separately: Each character from the “Text to Draw” text is drawn as an individual thumbnail. This means the number of thumbnails from each .TTF file is equal to the number of characters in the “Text to Draw” text.

Entire phrase: Each thumbnail is created by drawing the entire “Text to Draw” text. Each .TTF file will generate one thumbnail.

Appearance

Size
This is the height of the images that are used for each thumbnail. Generally speaking larger sizes mean greater detail in the drawn text. After the thumbnail is generated, it is still subject to resizing, cropping, etc., the same as any other input image (from a JPG or BMP, for example).

Italics
Draw the text in italics

Bold
Draw the text in bold face.

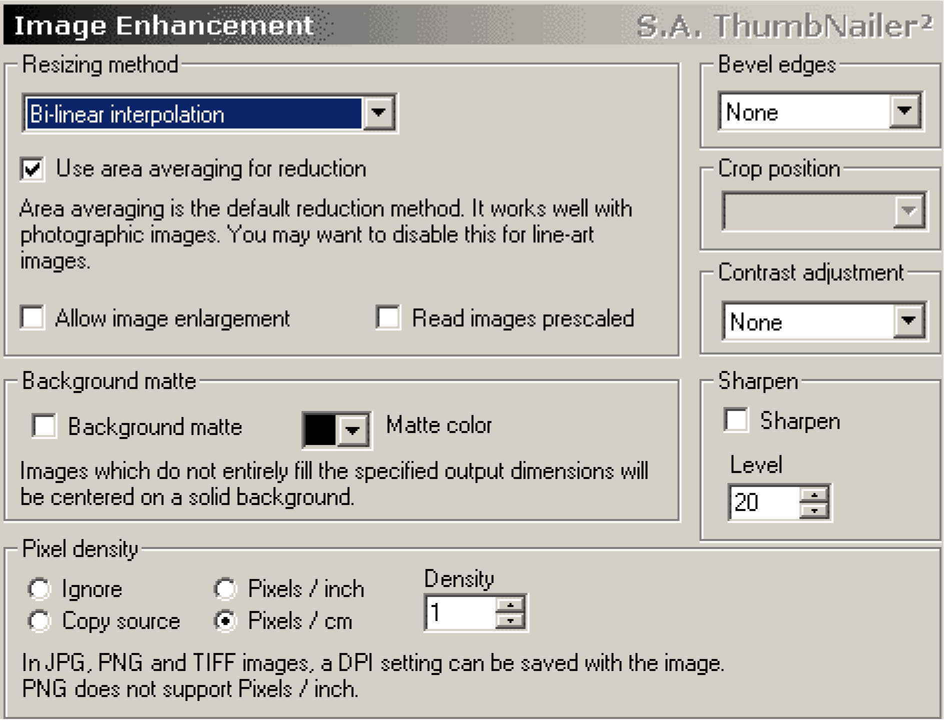
Text Color
This is the color used to draw the text

Font Face
This is the font to use for (most) text elements. You can enter anything you want here – ThumbNailer

does not check it.

Background Color
Text is drawn on top of a solid background. This is the color used for the background.

Image Enhancement



Resizing method

Bi-linear interpolation

☑ Use area averaging for reduction

Area averaging is the default reduction method. It works well with photographic images. You may want to disable this for line-art images.

☐ Allow image enlargement ☐ Read images prescaled

Bevel edges: None

Crop position:

Contrast adjustment: None

Background matte

☐ Background matte ■ Matte color

Images which do not entirely fill the specified output dimensions will be centered on a solid background.

Sharpen

☐ Sharpen

Level: 20

Pixel density

○ Ignore ○ Pixels / inch

○ Copy source ● Pixels / cm

Density: 1

In JPG, PNG and TIFF images, a DPI setting can be saved with the image. PNG does not support Pixels / inch.

(Advanced / Input Enhancement)

Resizing Method

This allows you to choose the scheme that ThumbNailer will use when resizing images.   For most images, most people will not notice the difference in the results of any of the resizing methods. But, ThumbNailer provides these options for cases in which one method just doesn't work well for a particular image type.
This also controls how the thumbnail images are resized for placement on composite images.

Bi-linear Interpolation
This is the default method and will produce good results, quickly.

Bi-cubic Interpolation
Using this method, the results are somewhat sharper in some cases, than bi-linear. It is slow, compared to bi-linear.

Mitchell Filter
Lanczos Filter
Catrom Filter
For most images, these resizing schemes yield superior results when compared to bi-linear and bi-cubic interpolation; but, they are also very slow. Most people will not notice any difference in the results, for most images, from any of these three schemes.

These schemes can be useful when enlarging images: the results will be better than bi-cubic or bi-linear interpolation. However, for reduction, ThumbNailer's "Area averaging" method gives results that are often just as good as these slower, more complex methods.

It's difficult to say when any of these methods will be better than the others. The best advice we can give is: if you don't like the results you're getting with one method, try a different one.

Use Area averaging for reduction
This is the default method for reducing images. For photographic images, this method gives excellent results, quickly. If you un-check this, ThumbNailer will use the method you've chose from the list above. You may want to un-check this when processing line art, or images with fine details.

If the **Allow Enlarge** button is checked, ThumbNailer will expand images to fit the specified dimensions. If this is not checked, input images that are already smaller than the specified Width and Height will not be enlarged but will be written with their original dimensions.

If you are using the Percent, Exact Fit or Pixel Area resize methods, this button will be disabled.

Read images prescaled
Normally, ThumbNailer reads each input image into memory at full size and resolution. For most images, this is not a problem. For large images, though, the amount of memory required for this is simply too large for the typical PC to handle. But if this option is checked, ThumbNailer will read large images at a reduced size. This can speed processing dramatically and can even allow ThumbNailer to process images that may otherwise be too large to hold in memory.

This option has no effect on images with less than 1200x1200 pixels.

Currently, this only affects JPG reading, though support for other formats may be added in the future.

Sharpen

Sharpen
The program will apply a sharpening matrix to the output image.

Sharpen Level
Control the intensity of the sharpening effect. 0 is low, 100 is high.

Pixel Density

This allows you to specify the pixel density (resolution) for the JPG, TIFF or PNG files that the program will write. For simplicity, all images are assumed to have square pixels (density of X = density of Y). This has no real effect on how the images are stored. The value is only used externally. Not all readers care about this value.

Pixel Density
Enter a value.

Ignore
The program will write JPG default values for the pixel density settings.

Copy source
The program will copy the pixel density settings from the source images.

Pixels / Inch
The pixel density value is assumed to be a pixels / inch value. PNG does not support Pixels / Inch - if this is selected, pixel density will be ignored for PNG files.

Pixels / cm
The pixel density value is assumed to be a pixels / centimeter value.

Background Matte

When using the Best Fit, Fixed Width and Fixed Height resize methods, it often happens that the image you are resizing can not be made to exactly fit into the dimensions you have specified. Using a background matte allows you to put a solid border around the actual image large enough to make the resulting image fit your specified output size.

Background Matte
Check this box to enable matting.

Matte Color
This is the color that the matte will be

Contrast

None
No adjustment is applied.

Raise Contrast
A slight contrast raise is applied.

Line Art Contrast
An extreme contrast is applied. This works great for black and white line-art thumbnails. With color images, however, this is extreme.

Lighten Midtones
A slight brightening of the mid-range colors is performed.

Darken Midtones
The program applies a slight darkening of the mid-range colors.

Bevel Edges
You can choose to have the program bevel the edges of the thumbnails. This gives a 3-D look to the images.

None
Don't apply any beveling

Before Matte
Apply bevel before the image is matted.

After Matte
Apply bevel after the image is matted.

Crop Position
This is used in conjunction with the *Fixed Width, Crop Vertical* and *Fixed height, Crop Horizontal* resizing methods. This allows you to choose the areas of the images that you want cropped.

If you have selected *Fixed Width, Crop Vertical* as your resizing method, the following options are available:

Center
The excess image area is cropped equally from the top and bottom of the image.

Top
The excess image area is cropped from the top of the image.

Bottom
The excess image area is cropped from the bottom of the image.

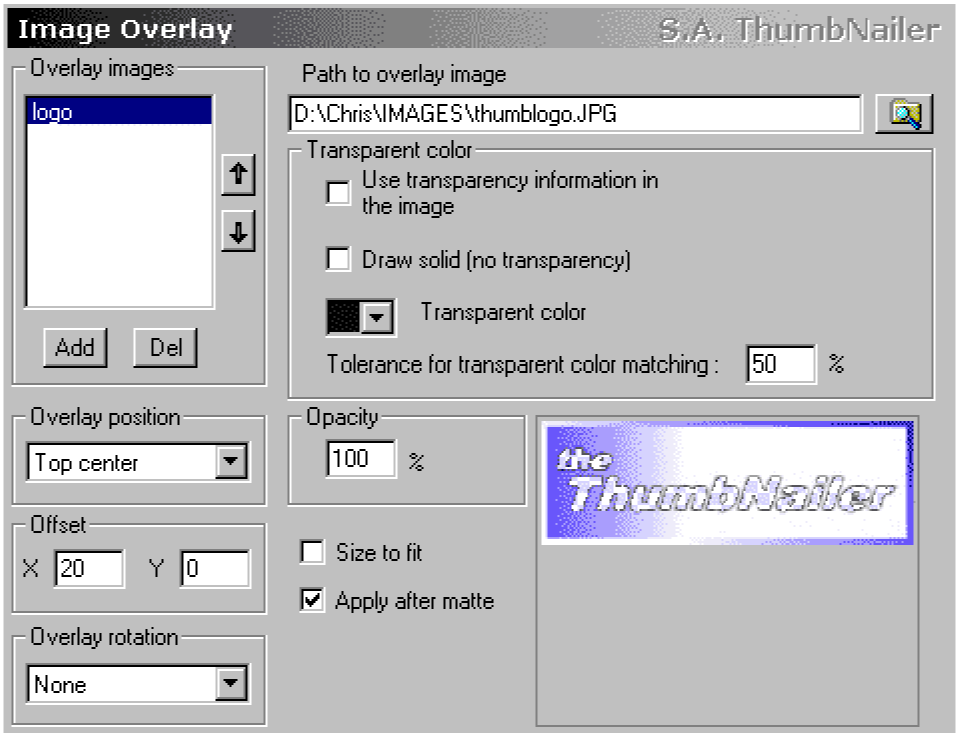
If you have selected *Fixed Height, Crop Horizontal* as your resizing method, the following options are available:

Center
The excess image area is cropped equally from the left and right of the image.

Left
The excess image area is cropped from the left side of the image.

Right
The excess image area is cropped from the right size of the image.

Image Overlay



Overlay images: logo

Path to overlay image: D:\Chris\IMAGES\thumblogo.JPG

Transparent color

□ Use transparency information in the image

□ Draw solid (no transparency)

Transparent color

Tolerance for transparent color matching : 50 %

Overlay position: Top center

Opacity: 100 %

Offset: X 20 Y 0

□ Size to fit

☑ Apply after matte

Overlay rotation: None

(Advanced / Image Overlay)

This allows you to overlay images over all output images. This is handy if you want to stamp your logo, or some informational text (like copyright or company info) over all of your thumbnails.

Overlay Images
This is a list of the images that you want to use as overlays. You can edit the name in the list by double-clicking on the name. Each name in the list has its own set of settings.

Add
Add a new overlay. You will be prompted for a filename.

Del
Delete the currently-selected image from the overlay list. This just removes it from the list, it doesn’t actually delete the file.

Up/Down
Change the position of the overlay in the list. Images are overlayed in this order.

Path to overlay image
Specify the filename of the image you want to use as an overlay.

Overlay Position
Select where the overlay image starts. If you have checked Size To Fit, the overlay image will be resized (without changing its own aspect ratio) to fit into the area you have specified. If you don't have Size To Fit checked, the overlay image will only start at the upper left corner of the area you have specified and may extend past.

Offset
Specify an X and Y offset for the overlay image. This can be used to make fine adjustments on the position of the overlay on the output image.

Overlay Rotation
ThumbNailer can do a 90 degree rotation of the overlay image, if you want.

Use Transparency Information in the Image
If this is checked, the Opacity and transparent color controls will be disabled and all transparency information will come from the input file. Some file types (PNG, TIFF, GIF and TGA) provide the option to save transparency information in the image (also called "alpha" information). This information controls how the image blends with the background when drawn. ThumbNailer can use this transparency information to do the overlay.

Opacity
The overlay image can be transparent. Use this slider to control the percent opacity of the overlay. 100 means the overlay image is opaque (no background shows through) 0 means the overlay is totally transparent (you can't see it).

Transparency
You can choose one color from your overlay image to be totally transparent, regardless of the opacity setting. Areas in the overlay image that contain this color will be 100 % transparent. This is handy if you want to overlay something non-rectangular.
Click the color box to pick a color from the color picker.

Draw Solid
Turn transparent color selection off

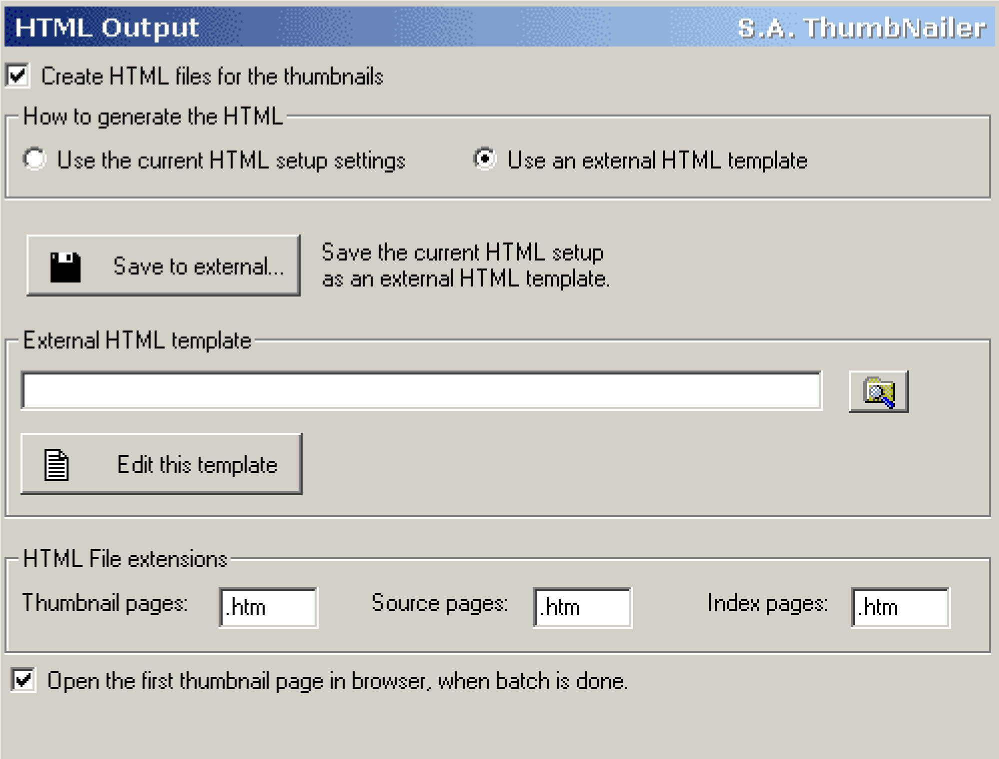
Color matching tolerance
When this value is zero, a single color can be used as a "transparent" color. By increasing this value, you increase the number of colors that are considered transparent. You still pick a single transparent color, but colors "near" to that color also become transparent. This allows you to select wide ranges of color for transparency.

Size To Fit
Check this to force the program to resize the overlay image to fit over every output image. If you don't check this, the overlay image will be drawn at its actual size over all output images - it may run off the edges.

Apply After Matte
By default, overlays are applied before the matter is applied. But you can specify that the overlay be applied after. This may cause the overlay to extend onto the matte.

The Image
When you move the mouse over the image, press the left mouse button, the color of the pixel directly underneath the cursor will become the "transparent color". All pixels with that color will then be considered transparent, and the blue and white-checkered pattern will show through.

HTML Output



☑ Create HTML files for the thumbnails

How to generate the HTML

○ Use the current HTML setup settings ● Use an external HTML template

Save the current HTML setup as an external HTML template.

External HTML template

HTML File extensions

Thumbnail pages: .htm Source pages: .htm Index pages: .htm

☑ Open the first thumbnail page in browser, when batch is done.

(Advanced / HTML Output)

The program can create HTML files based on the input and output files in a batch. HTML files are not created for Single File operations.

The HTML files are placed in the folders with the corresponding output images. If you have multiple output folders (if you are duplicating a source tree, for instance) the program will create HTML files in all folders where there are image files.

Create HTML Files
Check this to enable HTML creation

How to generate the HTML

Use the current HTML setup settings
If you choose this option, ThumbNailer will use the settings that you have chosen in the HTML Setup pages.

Use an external template
If you have created an external HTML Template, you can tell ThumbNailer to use your template instead of using the HTML setup settings.

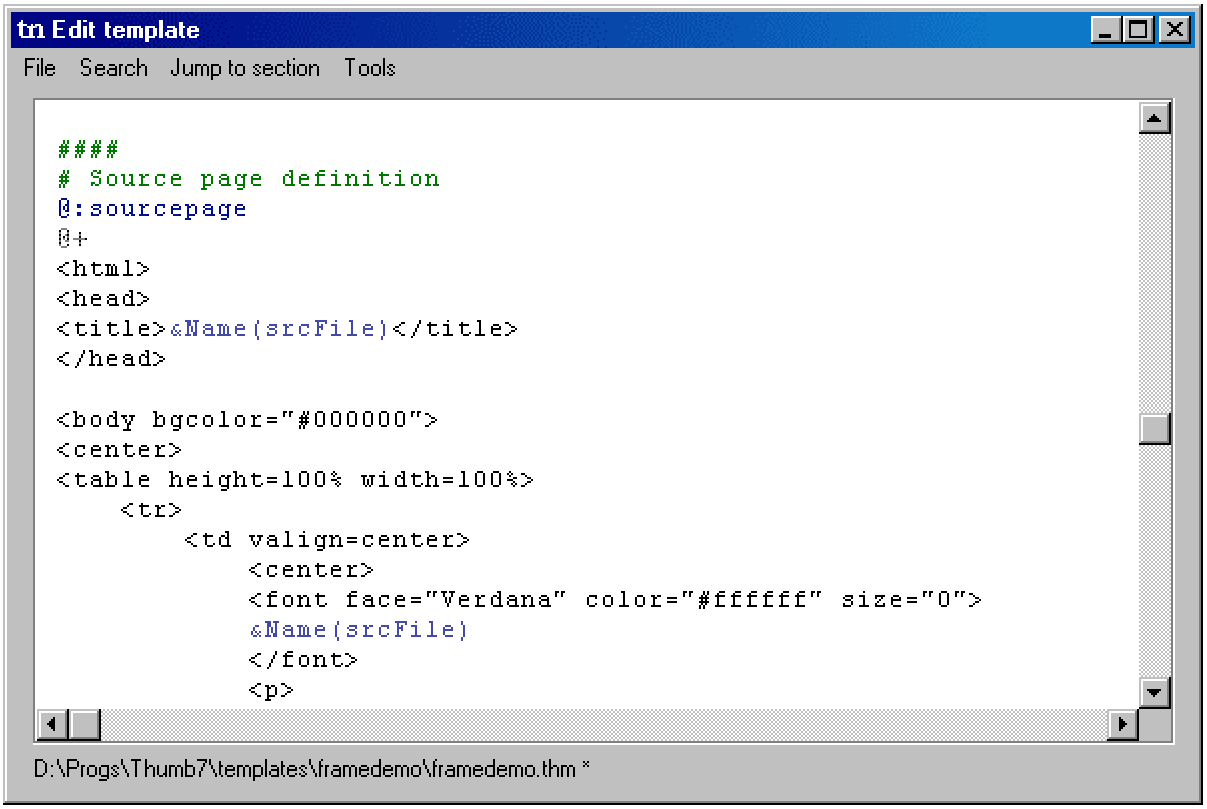
Edit this Template
This will bring up ThumbNailer's HTML Template Editor loaded with the template from the Use External Template field.

Save to external
Clicking this will create a ThumbNailer HTML template based on the current HTML setup. This can be handy if you want to base your template off of the basic ThumbNailer HTML layout.

HTML File Extensions

Enter the text that you wish to use for your HTML files’ file extensions.

HTML Template Editor



```
####
# Source page definition
@:sourcepage
@+
<html>
<head>
<title>&Name(srcFile)</title>
</head>

<body bgcolor="#000000">
<center>
<table height=100% width=100%>
    <tr>
        <td valign=center>
            <center>
            <font face="Verdana" color="#ffffff" size="0">
            &Name(srcFile)
            </font>
            <p>
```

D:\Progs\Thumb7\templates\framedemo\framedemo.thm *

(Advanced / HTML Output / Template Editor)

This is a text editor that you can use to create and edit ThumbNailer HTML templates. While you can use any text editor to create these files, ThumbNailer's editor is specifically designed to work with ThumbNailer templates.

When you edit a template with this editor, different sections of the template will be drawn with different colors.

The ThumbNailer template editor window, once you've started it, will remain open until you close it or shut down ThumbNailer. This way, you can edit templates easily while running batches – edit the template, run a batch, test your changes, edit the template, etc., without having to go through the Advanced windows.

The template editor window will be hidden while a batch is running, but will reappear when the batch is done.

Keyboard shortcuts

- Ctrl-Z – undo
- Ctrl-A – redo
- Ctrl-F – find
- Ctrl-S – save
- F3 – repeat last search

Menus

File

Save

Save to the current template. (Ctrl-S)

Save As…

Save the current template to a new file.

Load

Load a new template, discarding the current one.

Search

Find…

Find the first occurrence of a text string. The search starts at the current cursor position; if the search text is not found before reaching the end of the file, the search will start at the top of the file and continue until the cursor position.

Find Next

Repeat the last search from the current cursor position. (F3)

Replace

Replaces all occurrences of the search text.

Jump to section

Jump to a section definition - if the section exists.

Tools

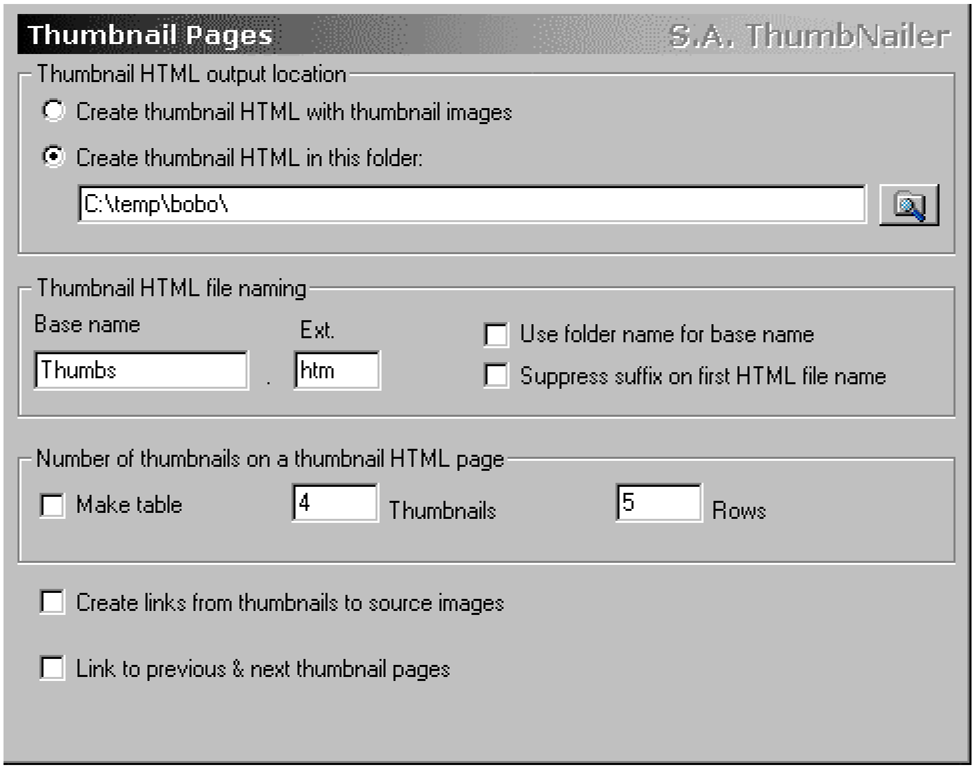
Validate

Perform a minimal format validation on the template. This will not find every error – it only checks that the template meets minimum format requirements.

Set colors

Change the colors that ThumbNailer uses for the different template parts.
Note: The colors of the text have no effect on how the template works.

Thumbnail Pages



Thumbnail HTML output location

○ Create thumbnail HTML with thumbnail images

● Create thumbnail HTML in this folder:

C:\temp\bobo\

Thumbnail HTML file naming

Base name: Thumbs . Ext.: htm

□ Use folder name for base name

□ Suppress suffix on first HTML file name

Number of thumbnails on a thumbnail HTML page

□ Make table 4 Thumbnails 5 Rows

□ Create links from thumbnails to source images

□ Link to previous & next thumbnail pages

(Advanced / HTML Output / Thumbnail Pages)

Thumbnail HTML location

Create thumbnail HTML with thumbnail images
This is the default setting. Thumbnail HTML pages will be created in the same folders as the thumbnail images.

Create thumbnail HTML in this folder
You can choose to have ThumbNailer create the thumbnail HTML pages in any folder you wish.

Note:
For best results, this folder, the root thumbnail image folder, the root source folder and the root source HTML folder should all be on the same drive. If these folders are not all on the same drive, Thumbnailer will be unable to generate relative paths for the image references and hyperlinks – the HTML pages won’t work.

HTML File Naming

HTML Base Filename
Enter the base file name for the HTML files.
Ex.
If you specify "Thumbs" as the base filename, the first HTML file written will be "Thumbs1.htm"; after the maximum number of images have been added to Thumbs1.htm, Thumbs2.htm will be created and filled in, and so on.

Use Folder Name
HTML index files will be named the same as the current lowest-level folder.
Ex. HTML index files generated in c:\temp\bob would have a base name of “bob”.
This has no effect on the source image pages.

Thumbnail arrangement

Make Table

Places the output images in an HTML table. If you don't check this, the images will be arranged (by default) in a single column.

Columns

The number of thumbnail columns in the HTML table.

Rows

The number of thumbnail rows in the HTML table.

Thumbnails (when not creating a table)
The total number of thumbnails on a page

Create Links To Source Images

If this is selected, the program will generate links to the source images. If you click on one of the images in the HTML file, it will cause your browser to show display the source file.

Note : these links are relative links - there is no drive or "file://" tag.

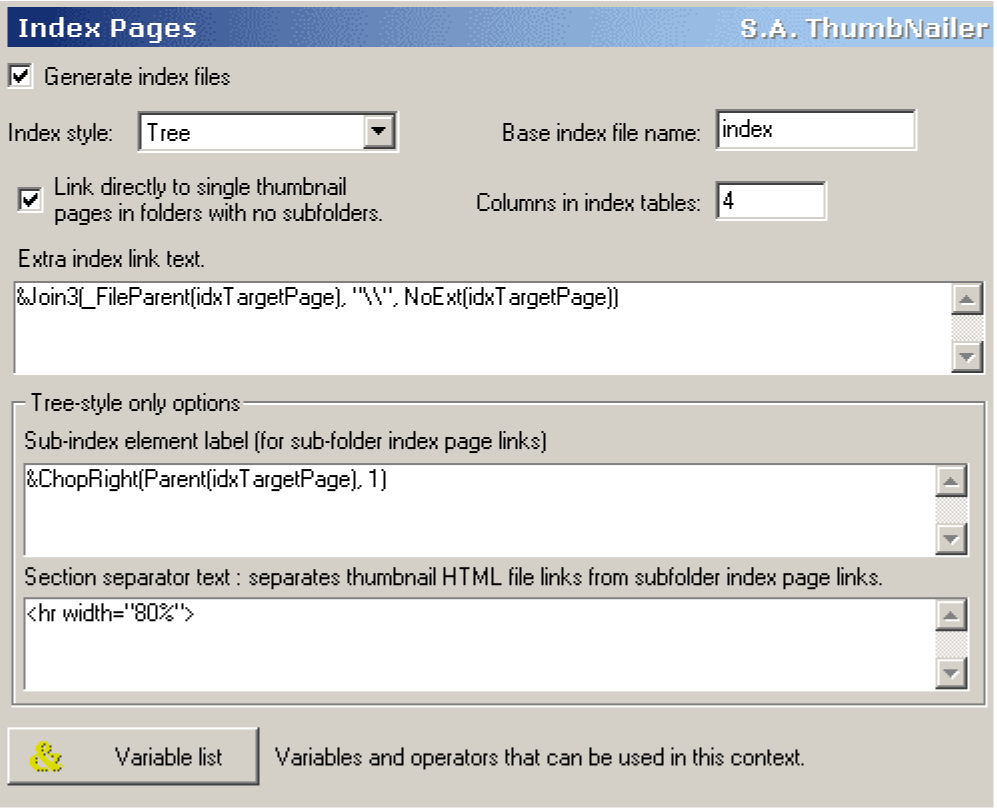
Note : the input and output folders must share a common root directory. If there is no common root, this operation will fail.
Ex.

- If your input folder is C:\temp\trix and your output folder is C:\temp\pix\thumbs, this will succeed because C:\temp is the common root directory.
- If your input folder is C:\temp\pix and your output folder is C:\thumbs\out, this will succeed because C:\ is the common root directory.
- If your input folder is C:\temp\pix and your output folder is D:\thumbs\out, this will fail.

Link to previous and next thumbnail pages

This will create a pair of Previous and Nextlinks to the previous and next ThumbNailer-generated HTML pages.

HTML Index Files



☑ Generate index files

Index style: Tree

Base index file name: index

☑ Link directly to single thumbnail pages in folders with no subfolders.

Columns in index tables: 4

Extra index link text.

&Join3(_FileParent(idxTargetPage), "\\", NoExt(idxTargetPage))

Tree-style only options

Sub-index element label (for sub-folder index page links)

&ChopRight(Parent(idxTargetPage), 1)

Section separator text : separates thumbnail HTML file links from subfolder index page links.

<hr width="80%">

Variables and operators that can be used in this context.

(Advanced / HTML Output / Index Pages)

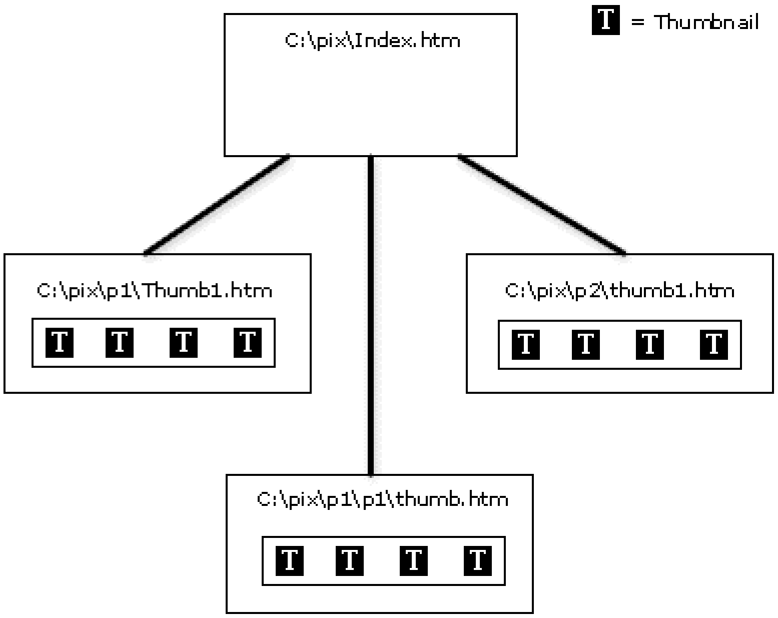
Index files show an index of all thumbnail HTML files created in the batch. Each thumbnail page is shown on the index page with the first image from that thumbnail page.

Generate Index Files

Enable index file generation.

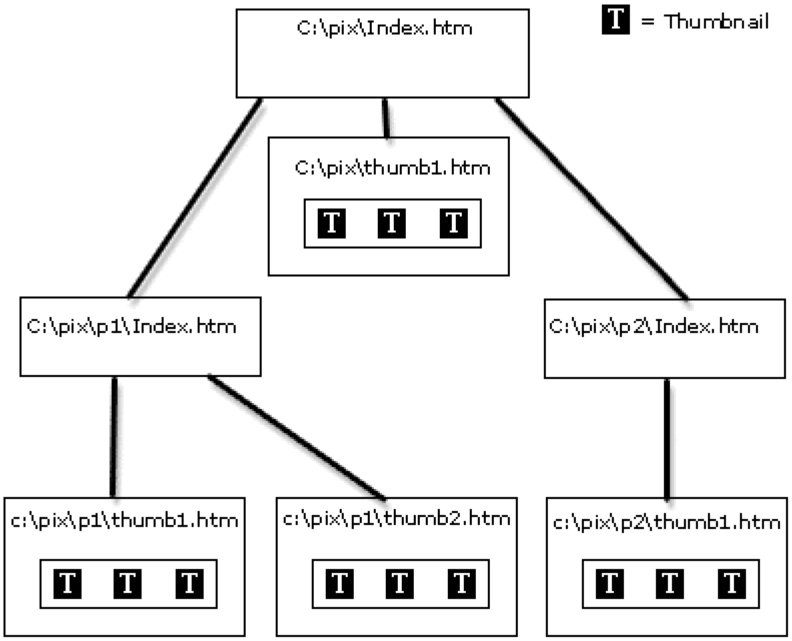
Index style:

Simple Index
One index.htm file is generated in the root destination folder. It has links to all thumbnail HTML pages generated in the batch.



Tree Index

An index.htm file is created in each destination folder. Each of these files contains links to the thumbnail HTML files in its folder and links to each index.htm file in its immediate child folders. Thumbnail links are at the top and index files are at the bottom.
Note: Only folders with thumbnail HTML files in them will receive an index file.



Also see Master Index Page.

Base index file name
This is the name given to each of the index files. The default is “index”.

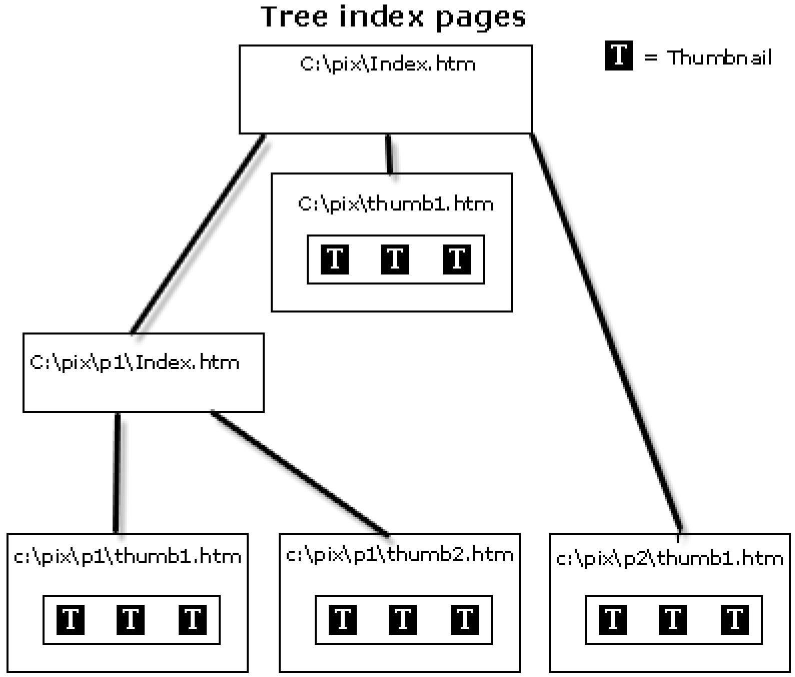
Columns

Choose the number of columns in the extended index page tables. Rows are added as necessary to show all of the thumbnail pages.

Link directly to single thumbnail pages in folders with no subfolders

If a folder contains a single thumbnail page, and has no subfolders below it, link directly to the thumbnail page from the index page in the folder above it, instead of creating an index page. This applies only to Tree-style index pages.

For example, notice how, in the picture above, the C:\pix\p2\Index.htm file links only to a single thumbnail file, which has no subfolders below it. By selecting this option, ThumbNailer will link directly from the C:\pix\Index.htm file to the C:\pix\p2\Thumb1.htm thumbnail file, instead of creating the C:\pix\p2\Index.htm index file.



This makes is browsing the images somewhat easier, since you don't have to click through index files that contain only one link.

Extra Index Text

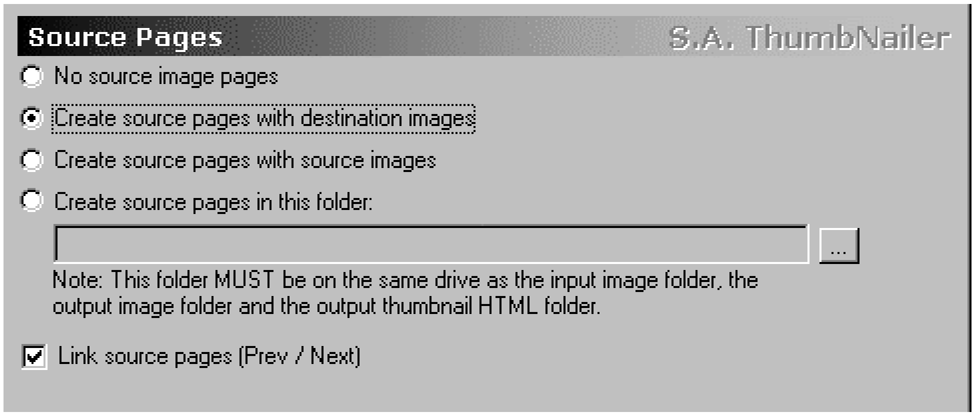
This text goes beneath each thumbnail page link. The default is the base file name (no path, no file extension) of the thumbnail page that is being linked to.

Section Separator Text

When you create "Tree" index files, this text is placed between the thumbnail links and the subfolder index pages.

Tip: *Right click on the control and choose "Bigger" from the menu, to enter your text in a larger window.*

HTML Source Pages



○ No source image pages

● Create source pages with destination images

○ Create source pages with source images

○ Create source pages in this folder:

Note: This folder MUST be on the same drive as the input image folder, the output image folder and the output thumbnail HTML folder.

☑ Link source pages (Prev / Next)

(Advanced / HTML Output / Source Pages)

The program can create HTML pages for individual source images. These are simple pages which contain any user-defined text and the source image. These will be linked-to by the pages created for the output images. The images in these pages will be linked back to the output pages.

Source HTML pages are named after the source image, using an automatic numeric suffix to avoid overwriting existing pages.

Note:
If you are extracting frames from a multi-framed image (like an AVI), and want to be able to link the thumbnails back to the original so that if the user clicks a thumbnail the original movie will open, **don't use source pages**. Your web browser won't be able to show the movie inside a web page. It's better to link to the source file directly – which is what will happen without source pages.

No Source Image Pages
Do not write HTML pages for source images. This is the default.

Source Pages with destination images
Write the pages with the output images. They files will contain relative paths to the source image files.

Source Pages with source images
Write the pages with the input images.

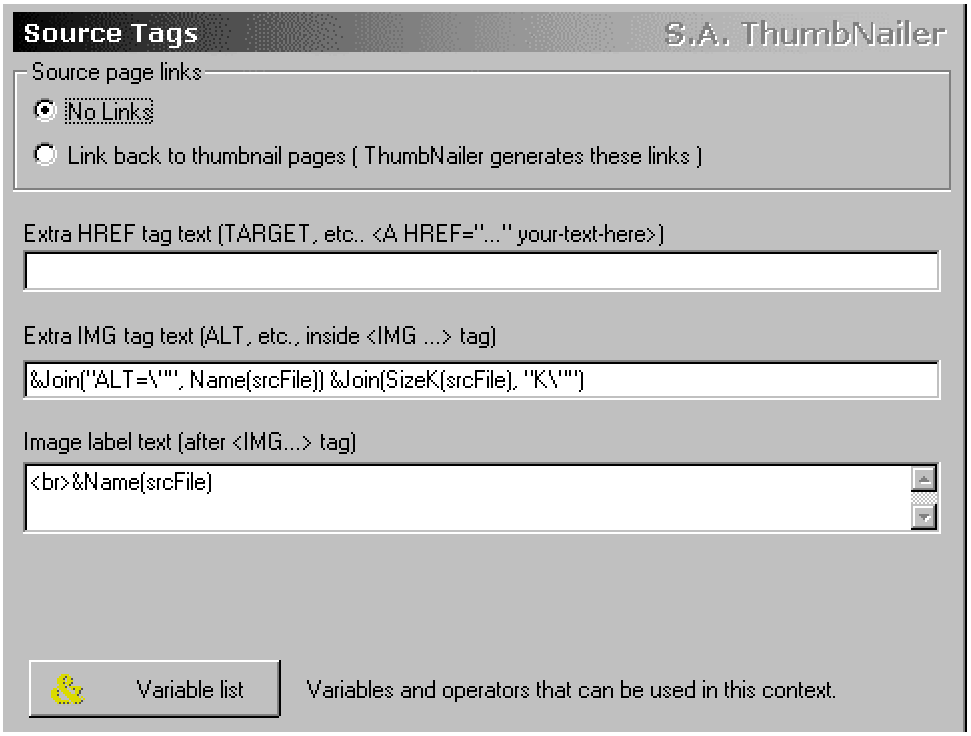
Source pages in this folder
All source image HTML pages will be placed in this folder (no attempt is made to duplicate the source folder tree structure).

Note:
For best results, this folder, the root thumbnail image folder, the root thumbnail HTML folder and the root source HTML folder should all be on the same drive. If these folders are not all on the same drive, Thumbnailer will be unable to generate relative paths for the image references and hyperlinks – the HTML pages won't work.

Link Source Pages (Prev / Next)
Checking this will cause Previous and NextPrevious and Next links to be placed on source image pages. These links will use the Prev / Next text as specified in the HTML Subtext pages. These will also follow the "Link Across Folders" rule.

For more information on Source Page tags, see HTML Source Tag Text.

HTML Source Tag Text



Source Tags

S.A. ThumbNailer

Source page links

● No Links

○ Link back to thumbnail pages (ThumbNailer generates these links)

Extra HREF tag text (TARGET, etc.. <A HREF="..." your-text-here>)

Extra IMG tag text (ALT, etc., inside <IMG ...> tag)

&Join("ALT=\"", Name(srcFile)) &Join(SizeK(srcFile), "K\"")

Image label text (after <IMG...> tag)

&Name(srcFile)

Variables and operators that can be used in this context.

(Advanced / HTML Output / Source Pages / Source Tags)

See variables for information on using variables.

Source Page Links

The program can place the source images in <A HREF ="xxx"><img src="xxx></a> tags for you.

No Links
The images are not places in HREF tags.

Link Back to Thumbnail Pages
The program will place relative links from the source images back to the thumbnail pages. This is the usual behavior.

Extra HREF Tag Text
If you supply text here, it will be written as <a href ="xxx" *your text*>.

IMG Tag Text
This is extra text that you can have placed in the IMG tags for the source images. It may be any text you wish. <img src="file.jpg" width=10 height =10 *yourTextGoesHere* >. The default text for source image pages is:

&Join("ALT=\"", &Name(srcFile)) &Join(SizeK(srcFile), "K\"")

If the current source image is "picture.jpg" and is 19.2Kb, this will add text similar to the following:

ALT="picture.jpg 19.2K"

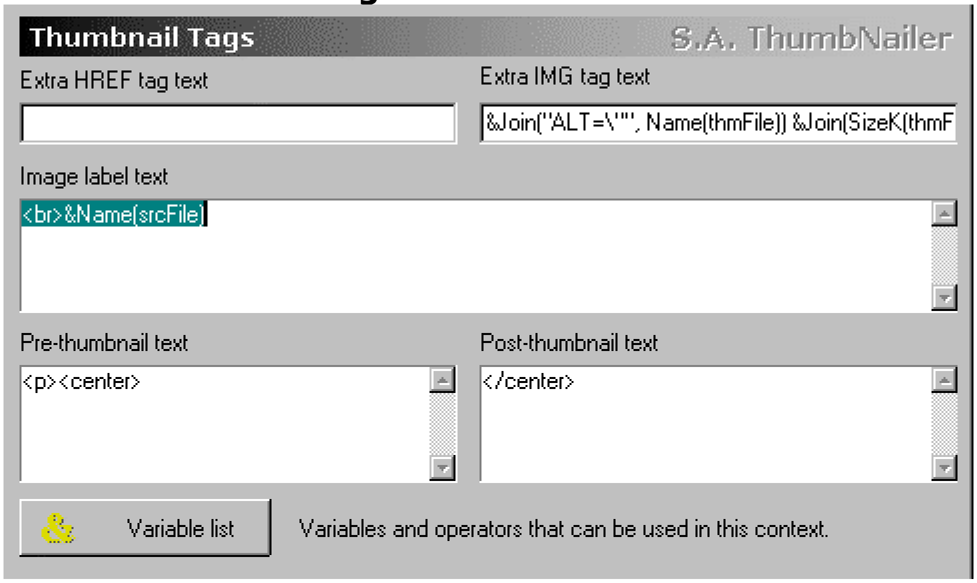
Image Label Text
This is the text label that goes beneath the source image. The default is a line break followed by the

name of the current source image: <br>&Name(srcFile).

Tip: *Right click on the control and choose "Bigger" from the menu, to enter your text in a larger window.*

Variables List
This will display a list of the variables which are available for these text fields.

HTML Thumbnail Tags



Extra HREF tag text:

Extra IMG tag text: &Join("ALT=\"", Name(thmFile)) &Join(SizeK(thmF

Image label text:
&Name(srcFile)

Pre-thumbnail text: <p><center>

Post-thumbnail text: </center>

See variables for information on using variables.

(Advanced / HTML Output / Thumbnail Pages / Thumbnail Tags)

Extra HREF Tag Text

This is extra text that you can have placed in the HREF tags for the thumbnail images. If you supply text here, it will be written as <a href ="xxx" *your text*>.

IMG Tag Text

This is extra text that you can have placed in the IMG tags for the thumbnail images. It may be any text you wish. <img src="file.jpg" width=10 height =10 *yourTextGoesHere* >. The default extra text for thumbnail images is:

&Join("ALT=\"", &Name(thmFile)) &Join(SizeK(thmFile), "K\"")

If the current thumbnail image is "picture.jpg" and is 19.2Kb, this will add text similar to the following:

ALT="picture.jpg 19.2K"

Image Label Text

This is the text label that goes beneath the thumbnail image. The default is a line break plus the name of the current source image : <br>&Name(srcFile).

Tip: *Right click on the control and choose "Bigger" from the menu, to enter your text in a larger window.*

Pre-thumbnail text

This text is placed immediately before each thumbnail image.

Tip: *Right click on the control and choose "Bigger" from the menu, to enter your text in a larger window.*

Post-thumbnail text

This text is placed immediately after each thumbnail image.

Tip: *Right click on the control and choose "Bigger" from the menu, to enter your text in a larger window.*

Variables List

This will display a list of the variables which are available for these text fields.

Profiles

A “Profile” is a collection of all of your ThumbNailer settings. ThumbNailer lets you save these settings to a profile, for later re-use.

In version 8, you can load or save the HTML setup portion of the profile separately.

Load Profile

Find and load one of your previously-saved profiles.

Note:
If you are loading only the HTML part of a profile, ThumbNailer maintains the current profile name (as displayed at the bottom of the main TN window) and only modifies the HTML settings. Ex. If you are working with a profile named “MyProfile” and load the HTML setup parts of a profile named “ProfileTwo”, ThumbNailer will read those HTML settings, but you’ll still be working with “MyProfile”.

Save Current

Save the current settings. This will save your settings into the profile you last loaded.

Save Current As..

Save the current settings to a new file.

Note:
When you save a profile using Save Current As…, ThumbNailer changes the name of the current profile (as displayed at the bottom of the main TN window). If you save the HTML setting only, the name is not changed.

***Note* :** All option settings will be saved to the last loaded or saved profile on exit. Ex. If you save a profile called “MyProfile” , change the output height then exit, the new output height will be saved to “MyProfile”. Likewise, if you load a profile, “Profile2”, change the output height, then exit, the new height will be saved in “Profile2”.

Order of Operation

These are the steps the program uses to create each output image.

1. Read the overlay image (once per batch)
2. Rotate and color quantize the overlay image (once per batch)
3. Read the source image (or next AVI, GIF or TIFF frame)
4. Apply fade (generate pre-defined mask or read and resize mask image file, once per image)
5. Calculate the output image size based on input size and resize method.
6. Resize the source image (in memory)
7. Apply overlay image
8. Sharpen
1. Enhance
1. Text overlay
11. Matte
12. Apply after-matte overlays
13. Quantize to 8-bit or grayscale
14. Write output image

DOS Operation

The program can be used from a Windows Command Prompt.

No output occurs in the DOS box.

By default, the program uses the options that were last set from Windows mode.

Note: It is required that you use a space between all parameters. Also, If a single parameter contains a space, you must put the parameter in quotes.

The key to this mode is the `/h` (for “hide”) parameter. It must immediately follow the executable name.

DOS ERRORLEVEL exits

If there are no errors, the program exits with a value of -1.

If there is a parameter syntax error, the program will exit with a value of 1.

If there is a processing error in a single-file operation, or in the initialization stage of a directory mode operation, the program will exit with a value of 2. Note that a file processing error in directory mode will not cause an error level to be set and the program will attempt to process all files as specified.

Single File Mode

```
c:> start /wait Thumb8.exe /h /f input_file output_file <other params>
```

Both input_file and output file are required.

Directory Mode

```
c:> start /wait Thumb8.exe /h /d input_folder output_folder <other params>
```

Both input_folder and output_folder are required.

List Mode

```
c:> start /wait Thumb8.exe /h /L input_file_list output_folder <other params>
```

Where `input_file_list` is the path to a text file which contains one file name per line. Each file listed will be processed in single file mode. Note that single file mode excludes such things as HTML and Composite generation

Both input_folder and output_folder are required.

Profile loading

By default, ThumbNailer will load the last-used profile settings first; then it will apply your command line switches.

Use Previously-saved Profile

```
/p <profile name>
```

Load and use a previously saved settings profile. The profile must exist, or system default values will be used instead. If specify other parameters on the command line, they will override those specified in the profile.

Command line parameters

All other parameters are optional and must come **after** the input and output folder specifications.
Ex: `c:> start /wait Thumb8.exe /h /d input output <other params>`

Reset

`/x`

Reset all options to program defaults

Enable Message Box Reports

`/!`

This will allow you to see the normal error reports from DOS mode.

Note: Unlike other DOS-mode switches, this switch can be used when starting in Windows mode.

Enhancements

No Enhancement

`/e0`

Raise Contrast

`/ec`

Lower Midtones

`/ed`

Raise Midtones

`/ei`

Extreme Contrast

`/ex`

Sharpen Image

`/v #`

# is required and represents the percent of sharpness applied.

Resize Method

`/r #`

# is one of :

0	Best Fit
1	Fixed Width
2	Fixed Height
3	Exact Size
4	Pixel Area
5	Percent
6	No Resize
7	Crop X
8	Crop Y
9	Auto Crop

Allow image enlargement

`/r+`

Allow images to be enlarged (default).

`/r-`

Prohibit image enlargement.

Output Size

`/s # #`

The first # is the width; the second is the height; both are required. If you are using the Percent resize method, the first number is the percentage, and the second number is required, but ignored. If you are using the Pixel Area resize method, the first number is the area; the second is ignored but required.

Output Format

```
/o #
```

# is one of :

0	24-bit Same as Input
1	8-bit Same as Input
2	8-bit grayscale, Same as Input
3	JPG
4	JPG grayscale
5	BMP 24-bit
6	BMP 8-bit
7	BMP 8-bit grayscale
8	PNG 24-bit
9	PNG 8-bit
10	PNG 8-bit grayscale
11	TIFF 24-bit
12	TIFF 8-bit
13	TIFF 8-bit grayscale
13	GIF 8 bit
13	GIF 8 bit grayscale
13	PCX 24-bit
17	PCX 8-bit
18	PCX 8-bit grayscale
19	TGA 24-bit
20	TGA 8 bit
21	TGA grayscale
22	Raw 24-bit
23	Raw 8-bit
24	Raw grayscale
25	WMF
26	EMF
27	BMP 4 bif
28	GIF 4-bit
29	PSD
30	PSD 8-bit
31	PSD 8-bit grayscale
32	WBMP
33	JPEG-2000
34	TIFF 1-bit, black and white
35	PNG 1-bit, black and white
36	BMP 1-bit, black and white

Date Filter

Not applicable in single file mode.

Date Filter

```
/t+ <dd/mm/yyyy>
```

Include only files modified after the date specified in mm/dd/yyyy

```
/t- <dd/mm/yyyy>
```

Include only files modified before the date specified in mm/dd/yyyy

```
/t!
```

Include only files modified after the last ThumbNailer run.

Output Naming Style

These switches are mutually exclusive; only one of them may be used at a time.

Overwrite

`/no`
Use the input name for the output name. ThumbNailer will still prompt before overwriting an existing file. To really overwrite, use the "`/n!`" switch instead.

Prefix
`/np <text>`
<text> is the prefix and is required.

Suffix
`/ns <text>`
<text> is the suffix and is required

Prefix plus autonumeric
`/n+ <text>`
<text> is the prefix and is required.

This allows you to create a list of thumbnails with names that are created by adding a zero-padded number to your text prefix text. The files will be named like this :
File00000.jpg
File00001.jpg
File00002.jpg
…and so on.

Numeric
`/nn`
Auto-numeric suffix.

Overwrite (really)
`/n!`
Use the input name for the output name. Overwrite the output file, without prompting. Be careful, this will overwrite any existing file. The only exception is that it will not overwrite the source image file.

Overwrite (no, really!)
`/nxxx`
Use the input name for the output name. Overwrite the output file, without prompting. **Be very careful** with this option. ThumbNailer will overwrite the output file, without warning; it will overwrite the source file or any other file without warning. **Use at your own risk**.

Output Name Case

Not applicable in single file mode.

`/cn`
Use the same case as the input filename

`/cu`
Force all characters in the output name to upper case

`/cl`
Force all characters in the output name to lower case

Image Overlay

Note: If you are using a profile that already has an overlay, these settings will modify the first overlay in the list.

`/oo <path>`
Specify an overlay image. All other overlay parameters will be set to default values.

`/op <pos> <resize> <rotation>`
Specify overlay image position.

<pos> is a position flag. The possible values are:
1 - no overlay
2- top, center
3   bottom, center
4 - center, left
5 - center, right
6 - top, left
7 - top, right
8 - bottom, left
9 - bottom, right
10 - center, center
11 - top edge
12 - bottom edge
13 - left edge
14 - right edge

<resize> is a true/false flag : 0=FALSE 1=TRUE. If TRUE, the overlay image will be sized to fit the position specified.

<rotation> specifies the overlay image rotation. The possible values are :
0 - none
1 - 90 degrees clock-wise
2 - 180 degrees
3 - 270 degrees clock-wise

```
/ot #
```
Specify the overlay image transparency. 1..100.

```
/ob #
```
Specify the transparent color in the overlay image. This is a "hex color", such as used in HTML pages. Ex. 888888 is medium gray.

Matte

```
/ma #
```
Specify the matte color for the image. This is a "hex color", such as those used in HTML pages. Ex. ffff00 is yellow.

Text Overlay

```
/tt <text>
```
Enable text overlay. <text> is the text to overlay.

```
/tf <font name> # #
```
Specify a text overlay font. <font name> is the font name. The first # is the font size, in points. The second # is the font color as a Hex Color.

```
/tp # #
```
Specify text overlay position.
The first value is the horizontal position. The possible values are :
0 - left
1 - center
2 - right

The second value is the vertical position. The possible values are :
0 - top
1 - center
2 - bottom

```
/ts # - Obsolete.
```

JPG Params

`/jp`

Enable progressive JPG saves.

`/jq #`

Set JPG quality level. 1..100.

Sample

```
c:> start /wait Thumb8.exe /h /d c:\images\big c:\images\small /no /r 0 /s 100 100 /o 3
```

This will cause the program to read all images from c:\images\big, resize them using the Best Fit method to 100x100 pixels and output them as JPGs to c:\images\small.

Note the spacing between switches and parameters. This is important!

NT / Win2000 / XP Users, note :

The program uses the system Registry to store some licensing information. This information is stored in the HKEY_CURRENT_USER hive. If the program cannot read the Registry, it will behave if it is not registered. This situation can occur when you are running from a non-user process (ie. ASP pages)

One solution is to run the program from a process that has access to HKEY_CURRENT_USER.

Another solution is to copy the registration info from:

HKEY_CURRENT_USER\SOFTWARE\SmallerAnimals\ThumbNailer\ProgramInfo

to

HKEY_LOCAL_MACHINE\SOFTWARE\SmallerAnimals\ThumbNailer\ProgramInfo

…and provide the "/hm" key on the command line. This will instruct ThumbNailer to read its registration information from HKEY_LOCAL_MACHINE.

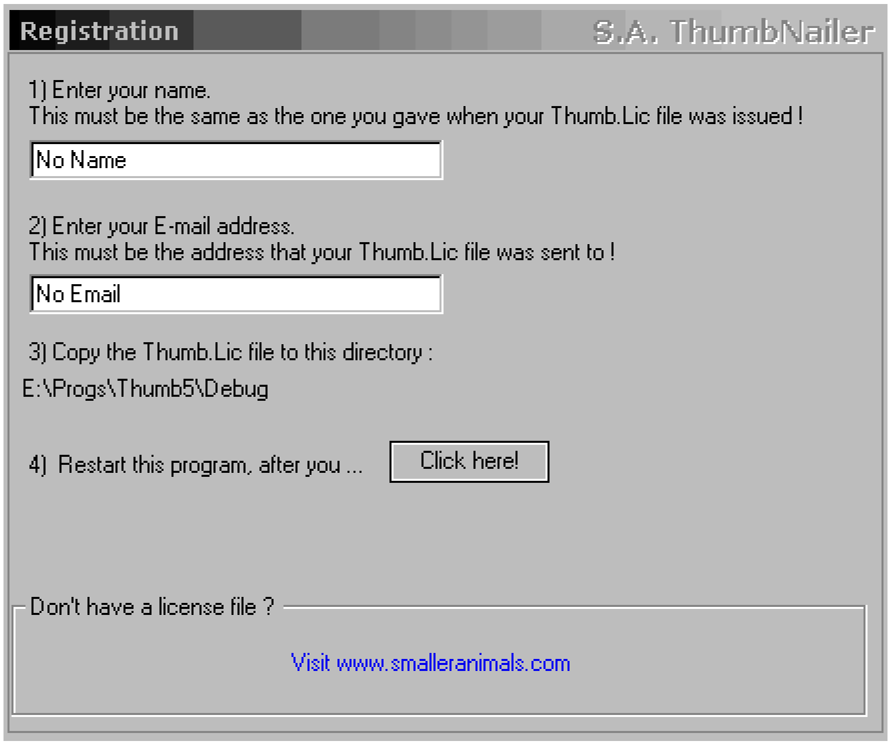
/hm

Read registration info from HKEY_LOCAL_MACHINE

/hu

Read registration info from HKEY_CURRENT_USER

Registration



1) Enter your name.
This must be the same as the one you gave when your Thumb.Lic file was issued !

No Name

2) Enter your E-mail address.
This must be the address that your Thumb.Lic file was sent to !

No Email

3) Copy the Thumb.Lic file to this directory :

E:\Progs\Thumb5\Debug

4) Restart this program, after you ...

Don't have a license file ?

Visit www.smalleranimals.com

This is where you enter the registration information that you will be sent when you register this program. After you have registered, you will no longer see this screen.

Unregistered Program Limitations
If you choose not to register the program, the following restrictions will be enforced:

1. Each HTML page that ThumbNailer creates will have some text on it that states that it was created with an unregistered copy.
1. You will be able to process a maximum of 50 files at a time. If you try to exceed this limit, only the first 50 files will be processed, and you will receive a warning message.
1. After 14 days, the Start button will be disabled and you will no longer be able to process images.

To register, please visit **http://www.smalleranimals.com**

License_TN.lic
This file will be sent to you upon receipt of your registration. You must place this file into the same directory that you run Thumb8.Exe from. The folder name is displayed for you in step 3 on the Registration page. Optionally, you can place this file in your Windows directory.

The license file contains information used by the program to confirm registration. Among other things, it contains the name and email address of the person who registered the program. This information is used to help track down software pirates.

If you choose to give out your license file, be aware that you are giving out a file with your name in it.

Users of previous versions

The License_TN.lic file will only work with ThumbNailer version 8.0 and higher.

If you are using ThumbNailer version 8 on a computer that currently has an earlier version installed, ThumbNailer will attempt to contact the Smaller Animals Software, Inc website to upgrade your license automatically. This automatic upgrade requires that you are using the original license that you were issued. If you have received a new license (a replacement of a lost license, for example), you may need to contact Smaller Animals Software, Inc to receive a manual license replacement.

Log Window

The small page icon at the bottom left of the main program window controls the display of the log window. When errors are encountered, they are automatically sent to this window. If the window is closed when the errors occur, the error log open/close icon will flash red until you open the log window or start a new operation.

Misc Info

Feature Creep

If you have an idea for a feature which would make the program more useful to yourself (and others), please send e-mail to smallest@SmallerAnimals.com. Most of what is in the program is a result of people just asking “can you do this?” Many things are actually simple to do, but just haven’t been brought up.

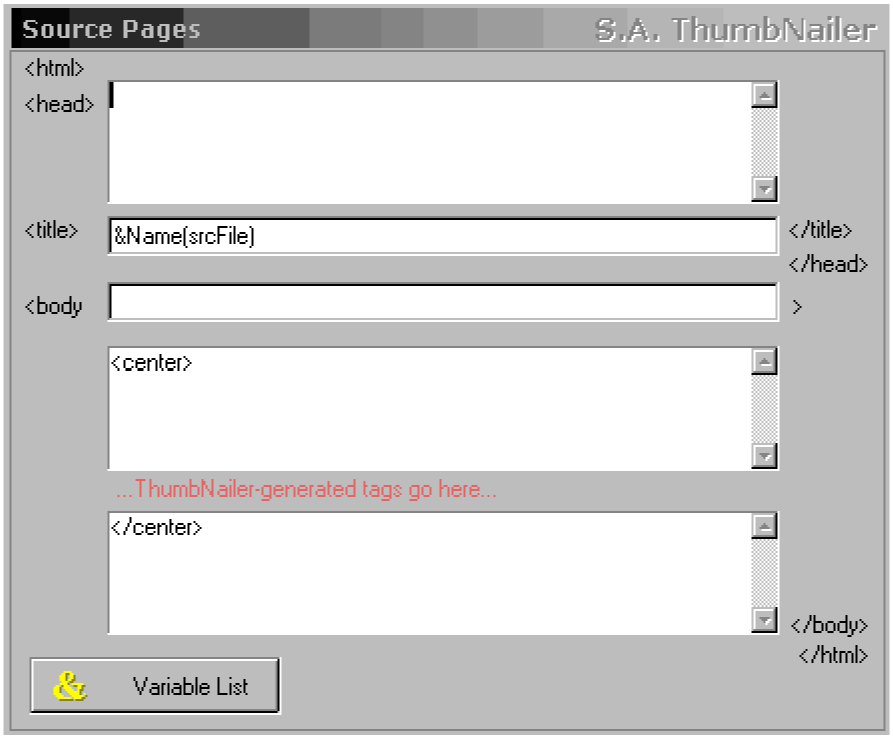
FAQ

There is a Smaller Animals ThumbNailer FAQ. See http://www.smalleranimals.com/thumb.htm for the location.

Discussion Board

Visit the Smaller Animals ThumbNailer Discussion Board. You can get there from:
http://www.smalleranimals.com/thumb.htm

HTML Subtext Pages



&Name(srcFile)

<center>

</center>

(Advanced / HTML Output / Thumbnail Pages / Thumbnail Page Text)
(Advanced / HTML Output / Source Pages / Source Text)
(Advanced / HTML Output / Index Pages / Index Text)

Tag Text Pages
You can edit the format of the text that goes into the HREF and IMG tags and the text that is displayed beneath images for both thumbnail and source image pages. See HTML Source Tag Text and HTML Thumbnail Tag Text**Previous / Next Labels Page**
You can edit the format of the labels given to the previous / next links on thumbnail pages. Previous / Next Labels
General Subtext
This dialog shows the general layout of a typical HTML file. The program allows you to add text to these areas :

- the <head> section *
- the <title> section
- the <body> tag
- between the <body> tag and where the program will place its stuff *
- between the program-generated stuff and the </body> tag *

**** Tip:** Right click on the control and choose "Bigger" from the menu, to enter your text in a larger window.*

0 You can edit this text for each of the three different HTML page categories that the program generates :

- Source - Pages generated to show the source images
- Thumbnails - Pages generated to show the output thumbnails
- Index - Index pages that link to the Thumbnail pages

File Import
If you want to include text from a file, enter a "%" followed by the filename, no spaces. The program will replace the "%filename" with the text from the file when it goes to write the HTML pages. Files imported in this way are only read once per batch. Files imported in this way may not be specified using variables. If you need more complex file import options, see the Import operator in the variables topic.

Note: *This operator is only available in the fields on the Index Page Text, Source Page Text and ThumbNail Page Text dialogs! It will NOT work in other text fields.*

Variables List
This will display a list of the variables which are available for these text fields.

Head / Title
Note that the Title section is enclosed in the Head section. Even if you don't specify text in the Title section, the program will write the <title></title> tags as shown.

Variables / Operators

You can cause the program to add specific information to text fields by using the pre-defined variables. Not all variables are available on all pages, because the information they represent is not applicable to all pages.

You can test your expressions in the expression tester
Note : The expressions and variables used in ThumbNailer v7.0 and above are *not* compatible with those used in previous versions. The expression syntax is similar but fundamentally different in a number of ways.

List of Variables

This is a list of all variables used by ThumbNailer's expression evaluator. Most of these are available in specific contexts and not available in others.

srcFile	Source image
thmFile	Thumbnail image
srcDir	Source image folder
destDir	Destination image folder
thmPage	Thumbnail page
srcPage Source	HTML page
prevPage	Previous source/thumbnail page
nextPage	Next source/thumbnail page
nextFile	Next source file
prevFile	Previous source file
idxPage	Index page
parentPage	Parent index page
idxTargetPage	Target index page
idxTargetThm	First thumbnail on the target thumbnail page
idxTargetSrc	First source page referenced by idxTargetThm
idxTargetCur	Number of the current thumbnail page in the index page list
cmpFile	Composite file
thmImageCount	Total number of thumbnail images created in this batch
thmImageCur	Index of the current thumbnail image (see thmImageCount)
thmPageCount	Total number of thumbnail pages in the current batch
thmPageCur	Index of the current thumbnail page (see thmPageCount)
thmPageList	List of all thumbnail HTML pages in this batch

These variables are available everywhere:

tnVers	ThumbNailer version information
noChar	An empty text string ("")

Operators

Besides the variables, there are text operators that can be used to extract and format information. All operators are available on all pages.

All of the operator names are case insensitive, unless otherwise specified. Ie. "Upper" is the same as "uppeR"

The Rules for Operators and Variables

- A parameter to an operator can be a variable, text, the name of a file, or, the result from another operator.

- Operators can be freely nested. Ex.

&Upper(NoExt(srcFile))

- In all cases, a “&” character is required to signal the beginning of an Operator/Variable expression. Only one “&” per expression :

  &Upper(NoExt(srcFile))

  not

  &Upper(&NoExt(srcFile))

- An open parenthesis “(“ is required after all operators.
- To use literal text in an expression, it must be placed in quotes: &Upper(“This is some text”)
- If the expression itself is inside quotes, it will be treated as literal text and will not be evaluated: “&Upper(srcFile)” = “&Upper(srcFile)”. If you need the output of the expression to be inside quotes, use the “Q” operator: &Q(Upper(srcFile)).
- To use a quotation mark inside quoted text, you must precede the quotation mark with a single backslash. In the text output, the backslash will not be present. Ex. &Upper(“Here is a quote \” inside some text”) = Here is a quote “ inside some text
- To use a backslash inside quoted text, you must precede the backslash with another backslash. In the text output, only a single backslash will be shown. Ex. &Upper(“Here is a backslash \\ inside some text”) = Here is a backslash \ inside some text
- To force a new line inside quoted text, use “\n”.

  Ex. &Lower(“One Line\nTwo Lines”) =
  one line
  two lines

- When using the Repl or Find operator to replace or find newline characters (“\n”), you should be aware that some applications will use the “\r\n” combination as a newline, instead of just a single “\n” character.
- The Operator/Variable expression ends at the last required closing parentheses after the “&”. Ex.

  Blah-&Upper(“c:\\temp\\pic.jpg”)-blah = Blah-C:\TEMP\PIC.JPG-blah

- Some operators expect their parameter to be the path to a file. Ex.

  &SizeK(Date(srcFile))

  This expression will generate error text because the SizeK operator will not be able to find a filename in the result from the Date operator: “Dec 8, 1998” is not a filename.

- The comparison operators (Eql, Not, Neql, Less, Grtr) output “1” for a true condition and “0” for a false condition.
- The first parameter of the If(x, A, B) operator is the “condition”. The program evaluates all parameters, then tests the first parameter text to see if it contains all digits (0 through 9 , “-“, and the “.”, for fractional values). If the text is all digits, it is treated as a number. If this number is non-zero, the condition is “true”; if the number is zero, the condition is “false”. Otherwise, if the text is not all digits, a non-empty text is considered to be a “true” condition, and an empty condition text is a “false” condition. In either case, a “true” condition will cause the text A to be output and a “false” condition will cause text B to be output.

- A **list** is a special variable type. Unlike all other variables, which hold a single item of text, a list holds a list of text items. ThumbNailer provides an operator to get an item from a list (LItem) and an operator to get the size of the list (LSize). Lists elements are indexed, starting at index “1”. To retrieve an item from a list, you must use the item’s index number: LItem(list, indexNumber). If you try to access a list using an index greater than the number of elements in the list, ThumbNailer will output an error message.

- If you use a list variable in an operator that doesn’t use lists, the value of the variable will be the name of the variable: &Upper(thmPageList) = “THMPAGELIST”.

- &Import(file) will read the text in a file into the current expression. Then, the program will evaluate this text for variables/operators. In cases where the Imported file contains Import statements, there is the chance that loops can be established. For example, if the file Input.Txt contains the line :

  &Import(“input.txt”)

  the program will read Input.txt, and evaluate it. Evaluating the text will cause ThumbNailer to re-read input.txt and evaluate it, etc. To prevent infinite loops, the program only allows ten levels of nested Import statements.

List of Operators

The following operators work with text.

Upper(text)
Convert text to Upper Case &Upper(“fred”) = FRED

Lower(text)
Convert text to Lower Case

Join(text1, text2)
Concatenate text1 and text2 &Join(“Red”, “Blue”) = RedBlue

Join3(text1, text2, text3)
Concatenate text1, text2 and text3 &Join3(“Red”, “Blue”, “Green”) = RedBlueGreen

JoinX(N, text1, text2...)
Concatenate an arbitrary number of text items. The first parameter is the number of text items to join.

&JoinX(4, “1” “Red”, “Blue”, “Green”) = 1RedBlueGreen

Len(text)
Find the length of “text” &Len(“stupid”) = 6

Left(text, count)
Extract the leftmost “count” characters from the text. &Left(“big long text”, 3) = big

Right(text, count)
Extract the rightmost “count” characters from the text.

ChopLeft (text, count)
Remove the leftmost “count” characters from the text.

ChopRight(text, count)
Remove the rightmost “count” characters from the text.

Mid(text, count, start)
Extract the “count” characters from the text, starting at “start”.

Repl(text, t1, t2)
Replace all occurrences of “t1” in text with “t2”

URLPath(param)
Formats param in URL format by replacing non-alpha-numeric characters with server-friendly escapes. This ignores the following characters: “ \ . ~ / _ - : “.

URL(param)
Like URLPath, but operates on all non-alpha-numeric characters. This shouldn’t be used with paths, since it will convert characters that are used in directory paths (ex. “/”, “:” and “\”)

Find(t1, t2)
Find the first occurrence of text “t2” in text “t1”; returns the position, starting at zero. &Find(“Some text”, “me”) = 2

Rfind(t1, t2)
Find the last occurrence of text “t2” in text “t1”; returns the position, starting at zero.

LMatch(t1, t2)
Find the position of first character where “t1” and “t2” differ (starting with 0). “-1” if no match.
&LMatch(“House”, “Horse”) = 2

Q(text)
Place the text in quotation marks. &Q(“Text”) = “Text”

T(text)
Output the text. &T(“hi”) = hi

The following operators only work on Lists.

LItem(list, index)
Return the list item from the list “list” at index “index”. The first element of the list has index 1.

&LItem(thmPageList, 2) = d:\thumbs\thumbs2.htm

Returns the second item in the thumbnail pages list.

LSize(list)
Return the number of items in the list.

The following operators do not require any input.

SysDate()
Returns the current system date as “Jan 31 1921”

SysTime()
Returns the current system time as “12:45:02”

Other

Month(MonthNumber)

Returns the name of the month in the local language &Month(3) = "March"

The following operators require the parameter to be the name of an actual file, or folder. I.e., they do not operate on simple text strings, but require a real live file.

Date(file)
File Modification Date of file

Time(file)
File Modification Time of file

Fexist(file)
Test for the existence of the file. Returns "1" is the file exists, else "0".

Text(file)
PNG or JPG Comment Text from file

Exif(file, tagID)
Read EXIF data from a JPG file. More Info

ExifI(file, tagID)
Read EXIF data from a JPG file. Some fields are interpreted. More Info

Iptc(file, tagID)
Read IPTC data from a JPG file. More Info

Size(file)
File Size, in Bytes of file

SizeK(file)
File Size, in Kilobytes of file

XDim(file)
Image Width of image in file

YDim(file)
Image Height of image in file

Dims(file)
Image Dimensions of image in file, as "100 x 100"

Import(file)
Read the text of the file into the current expression and evaluate it.

FileOut(outputFile, outputText, append)
Write the outputText to outputFile. If append is 0, the file is cleared before writing, else the text is appended to the end of the file. This operator is unique in that it returns no text, except when a file write error occurs.

&FileOut(c:\ text.txt, srcFile, 0) <- writes the name of the current source image to the file "c:\ text.txt". Erases the current contents of the file, if any.

Lookup(file, key, n)
Returns the nth field from the first line where the first field on that line matches "key". "File" is a tab-delimited file (fields in the file are separated by a *single* tab). "n" starts at 0, which is the key field. This operator can be used to perform simple database-style lookups. Text files are read the first time they are referenced and cached for the duration of the batch. Imported text is evaluated as an

expression.

The following operators operate only on text. They do not require real files.

Parent(folder)
Lowest folder in the path &Parent(“c:\bob\fred”) = fred\
(try _FileParent for the parent folder of a file)

Name(file)
File Name, no Path &Name(“c:\bob\me.jpg”) = me.jpg

Path(file)
Path, no Filename, of file &Path(“c:\bob\me.jpg”) = c:\bob\

Ext(file)
File Extension of file &Ext(“c:\bob\me.jpg”) = jpg

NoExt(file)
File Name, no Extension, of file &NoExt(“c:\bob\me.jpg”) = me

RelPath, (Folder A, Folder B)
Relative Path From Folder "B" to Folder "A".

RelFile, (File A, File B)
Relative Path From File "B" to File "A", Includes "To" File

Add(x, y)
Add x and y &Add(15, 5) = 20

AddX(N, x, y)
Add an arbitrary number of items. The first parameter is the number of items to add.
&Add(3, 15, 5, 100) = 120

Sub(x, y)
Subtract y from x &Sub(15, 5) = 10

Mul(x, y)
Multiply x and y &Mul(15, 2.2) = 33

Div(x, y)
Divide x by y &Div(15, 5) = 3

Rand(x, y)
Generates a random number between x and y.

Advanced operators

If(x, A, B)
If x is does not equal zero or is non-blank, output A, else B &If(stuff, A, B) = A &If(0, A, B) = B

Or(x, y)
If one of x or y does not equal zero or is non-blank, output “1”, else “0”

OrX(N, x, y)
If any one of the values is not equal to 0 or is non-empty, output “1”, else "0". The first parameter is the number of items to test.
&OrX(3, 1, 2, “text”) = 1
&OrX(2, 0, 0) = 0

And(x, y)
If both x and y do not equal zero and are non-blank, output “1”, else “0”

AndX(N, x, y)
If all of the values are not equal to 0 and non-empty, output "1", else "0".
&AndX(3, 1, 2, “text”) = 1
&AndX(3, “text”, 0, 2) = 0

Eql(x, y)
If x equals y, output “1” else “0” &Eql(“fred”, “steve”) = 0

NEql(x, y)
If x equals y, output “0” else “1” &NEql(“fred”, “steve”) = 1

Not(x)
If x equals “0” or is empty, output “1”, else output “0”.

Less(x, y)
If x is less than y, output “1”, else “0” &Less(“Abc”, “Def”) = 1

Grtr(x, y)
If x is greater than y, output “1”, else “0” &Grtr(“Abc”, “Def”) = 0

LEql(x, y)
If x is less than or equal to y, output “1”, else “0”

GEql(x, y)
If x is greater than or equal to y, output “1”, else “0”

VarSet(“var”, val)
Create a temporary user-defined variable. This variable can hold any text (“val”) you wish. After it has been set, you can use it in the same ways you use the built-in variables (like srcFile, thmPage, etc.)

Examples:
&VarSet(“myVariable”, Join3(“I ”, “like ” “cheese”)) (no output)
&Upper(myVariable) = “I LIKE CHEESE”
&Lower(myVariable) = “i like cheese”

VarSet Note: The variable name must be in quotes.
VarSet Note: The variable you create is only valid inside a single template section.
VarSet Note: It is recommended that you only use this operator when using external templates.

Built-in Macros

The following macros are built-in. You can use them as you would any other operator.

_Max(__value1, __value2) &_Max(3,4) = 4
Output the maximum of the two values.

_Min(__value1, __value2) &_Min(3,4) = 3
Output the minimum of the two values.

_FileParent(__filePath) &_FileParent(“d:/pix/image.jpg”) = pix
Output the parent folder of the file

_LinkFiles(__from, __to, __text)
Create a link from from the __from file to the __to file, using __text as the link text. This must only be

used when both files are on the same drive; this is because the macro generates relative paths, and that is only possible to do when files share a common root folder.

&_LinkFiles(thmPage, srcPage, "Source!") = <a href="source.htm">Source!</a>

_ThumbLink(__dir, __offset)
Create a link from the *current* thumbnail page to another thumbnail page. If __dir = 0, the link is to a "previous" page (a page that comes before the current thumbnail page). If __dir is 1, the link is to a page that comes after the current thumbnail page. The __offset specifies the number of pages from the current page. The following links to the previous thumbnail page:

&_ThumbLink(0,1) = <a href="thumbs2.htm">thumbs2</a>

The following links to the page before previous thumbnail page:

&_ThumbLink(0,2) = <a href="thumbs1.htm">thumbs1</a>

The following links to the thumbnail page that comes 3 after the current:

&_ThumbLink(1,3) = <a href="thumbs4.htm">thumbs4</a>

The link text is always the base filename (no extension) of the linked page.
If the offset and the dir specify a page that doesn't exist (ex. If the offset is 10 but there are only 5 pages), this will output "" (blank text of zero length).

Samples

Assume : srcFile = c:\temp\pix\brothers\steve.jpg

&Name(srcFile)	= steve.jpg
&Q("hello")	= "hello"
&T("hello")	= hello
&Q(srcFile)	= "c:\temp\pix\brothers\steve.jpg"
&T(srcFile)	= c:\temp\pix\brothers\steve.jpg
&Path(srcFile)	= c:\temp\pix\brothers\
&Parent(Path(srcFile))	= brothers\
File : &Name(srcFile) is &SizeK(srcFile) KB.	= File : steve.jpg is 71 KB.
&T(thmImageCur) of &T(thmImageCount)	= 44 of 100 – ie. This is the 44th thumbnail out of 100 total
Page &T(idxTargetCur) of &T(thmPageCount)	= Page 5 of 24
&Join(Path(srcFile), "textfile.txt")	= c:\temp\pix\brothers\textfile.txt
&Join3("abc", "def", "ghi")	= abcdefghi
&Left(srcFile, 5)	= c:\te
&Right(srcFile, 5)	= e.jpg

&Mid(srcFile, 5, 6) = mp\pix

&Left(srcFile, Div(Len(srcFile), 2)) = c:\temp\pix\bro - ie, the first half of the text

&ChopRight(“abcdefg”, 3) = abc

&ChopLeft(“abcdefg”, 3) = defg

&If(Eql(NoExt(srcFile) , “steve”), “Cool”, “Lame”) = Cool
This translates to: if the name, without the file extension, of the source file equals “steve”, output “Cool”, else output “Lame”.

&Import(Concat(Path(srcFile), “textfile.txt”)) reads c:\temp\pix\brothers\textfile.txt into the current expression, then evaluates it.

&If(Or(Eql(NoExt(srcFile) , “steve”), Eql(NoExt(srcFile), “bob”), “Cool”, “Lame”) = Cool .
This translates to: if the name, without the file extension, of the source file equals “steve”, or, the name, without the file extension, of the source file equals “bob” output “Cool”, else output “Lame”.

Differences between v6 expressions and v7,8 expressions

There are significant differences between the version 7 (and v8) operator/variable expressions and previous versions. These differences make the two systems strictly incompatible.

Literal text
Literal text is text between quotation marks that is used as a parameter to an operator. The quotes allow the expression evaluator (the part of ThumbNailer that turns expressions into text) to easily distinguish between plain text and the other kinds of parameters (numbers, operators and variables). &Len("This is text") : "This is text" is literal text.

Quotation marks are required around any parameter that is not an operator, a number, or a variable.

Escape characters
Because of "literal" text, ThumbNailer uses the quotation mark as a special character in expressions, we need a way to use a quotation mark inside literal text. ThumbNailer uses the backslash "\" as an *escape* character. The backslash tells the expression evaluator to ignore any special meaning of the following character.

Ex.: &Upper("I said \"Hello\".")

Each backslash in the expression above tells the evaluator to treat the quotation mark that follows it not as a special text-enclosing quotation mark but as a plain old quotation mark.

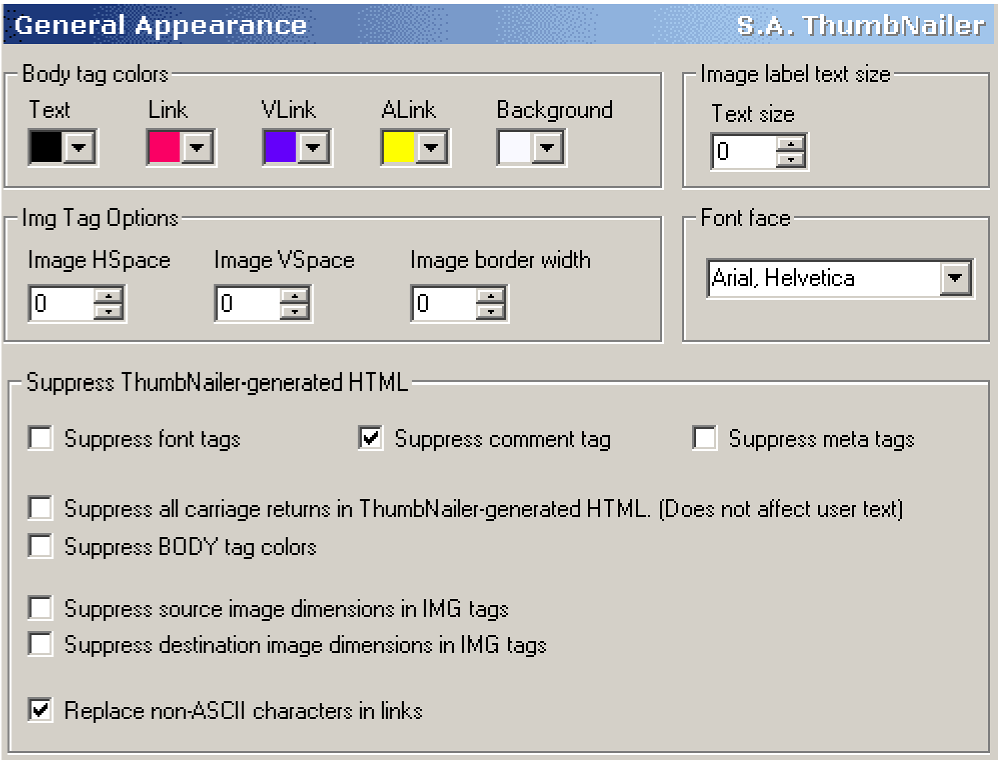
And, as a consequence, the backslash has now become a special character, too. If you want to use a backslash in quoted text you need to precede it with a backslash, too: &Upper("c:\\temp\\junk\\file.txt").

The \n sequence can be used to create a new line within quoted text:

```
&Upper("One Line\nTwo Lines") =
ONE LINE
TWO LINES
```

New operators and variables
ThumbNailer 7 introduced a number of new variables and operators and removes some of the old ones. Ex.
The "Concat" operator is now the "Join" operator. "Join" is a more familiar word than "concatenate".

Appearance



General Appearance — S.A. ThumbNailer

Body tag colors

Text	Link	VLink	ALink	Background
⬛	🟥	🟪	🟨	⬜

Image label text size

Text size: 0

Img Tag Options

Image HSpace: 0

Image VSpace: 0

Image border width: 0

Font face: Arial, Helvetica

Suppress ThumbNailer-generated HTML

☐ Suppress font tags ☑ Suppress comment tag ☐ Suppress meta tags

☐ Suppress all carriage returns in ThumbNailer-generated HTML. (Does not affect user text)

☐ Suppress BODY tag colors

☐ Suppress source image dimensions in IMG tags

☐ Suppress destination image dimensions in IMG tags

☑ Replace non-ASCII characters in links

(Advanced / HTML Output / General Appearance)

This dialog allows you to select various display parameters for the HTML files that the program generates.

Body Tag Options

These options control what is entered in the <BODY> Tag.

Text Color
Use this to set the text color

Background Color
Use this to set the background color

Link Color
Use this to set the link color

VLink Color
Use this to set the visited-link color

ALink Color
Use this to set the active-link color

Image Label Text Size

Text Size
This allows you to set the text size in the standard HTML range of -2 to +2. This applies only to the image labels.

Img Tag Options

These options control what is entered in the <IMG> tags.

Image Border Width
Set the image border width

Image Vspace, Image Hspace
Set the Image Vspace and Hspace fields

Font Face
Set the Font face used in all HTML text.

Suppress ThumbNailer-generated tags

Suppress Font Tags
If you check this, the program will not add <font> tags to the image labels.

Suppress Comment Tags
If this is not checked, the program will insert HTML comments into the thumbnail and source pages to mark the sections. This can be useful when fine-tuning your page layouts.

Suppress Meta Tags
Normally, ThumbNailer will put META tags in the HTML that credit itself with creating the page. You an disable this.

Suppress all carriage returns in ThumbNailer-generated HTML.
Some browsers (IE 5, for example) do not handle carriage-returns properly in all cases. Turning this off will cause ThumbNailer to generate HTML files without any carriage returns at all.

Suppress BODY Tag Colors
If you check this, the program will not add color specifications into the <BODY> tag.

Suppress Source Image Dimensions in IMG Tags
If you check this, the program will not add width and height specifications to source image IMG tags.

Suppress Destination Image Dimensions in IMG Tags
If you check this, the program will not add width and height specifications to destination image IMG tags.

Replace non-ASCII characters in links
Replace characters which are forbidden in URLs with URL-friendly escape codes.

Thumbnail Table Appearance

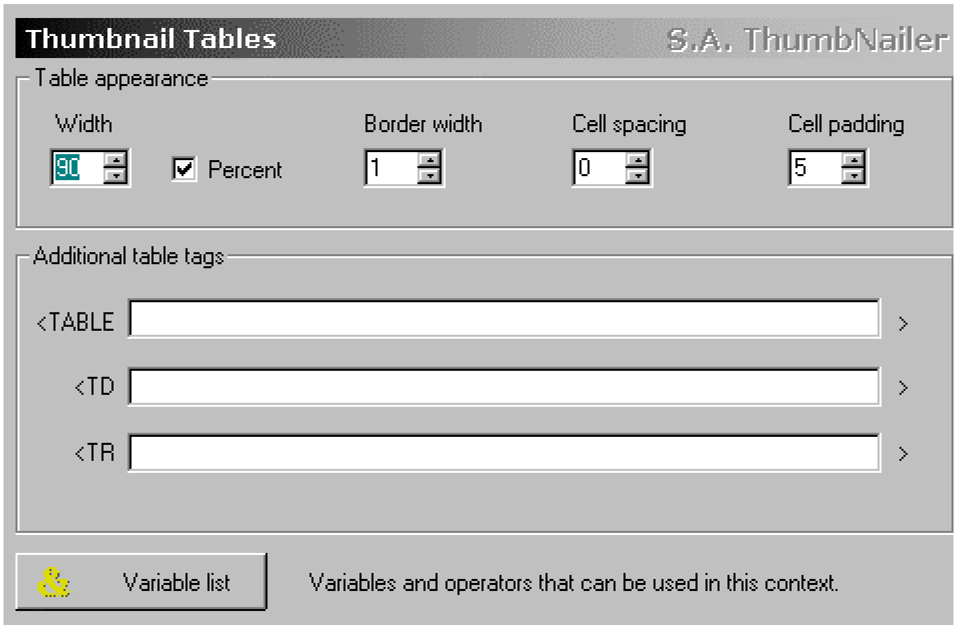


Table appearance

Width 90 ☑ Percent

Border width 1

Cell spacing 0

Cell padding 5

Additional table tags

<TABLE >

<TD >

<TR >

Variables and operators that can be used in this context.

(Advanced / HTML Output / Thumbnail Pages / Thumbnail Tables)

Table Options

These controls allow you to set various options for the tables that enclose the thumbnails.

Width
Set the percentage/size of the browser window that this table will take up.

Percent
If you check this, the *Width* number will be followed by a percent sign, "%". This makes the table a percent of the browser window width. If this is not checked, the table width will be *Width* pixels.

Border Width
Set the width of the borders for tables.

Cell Spacing
Set the spacing between table cells.

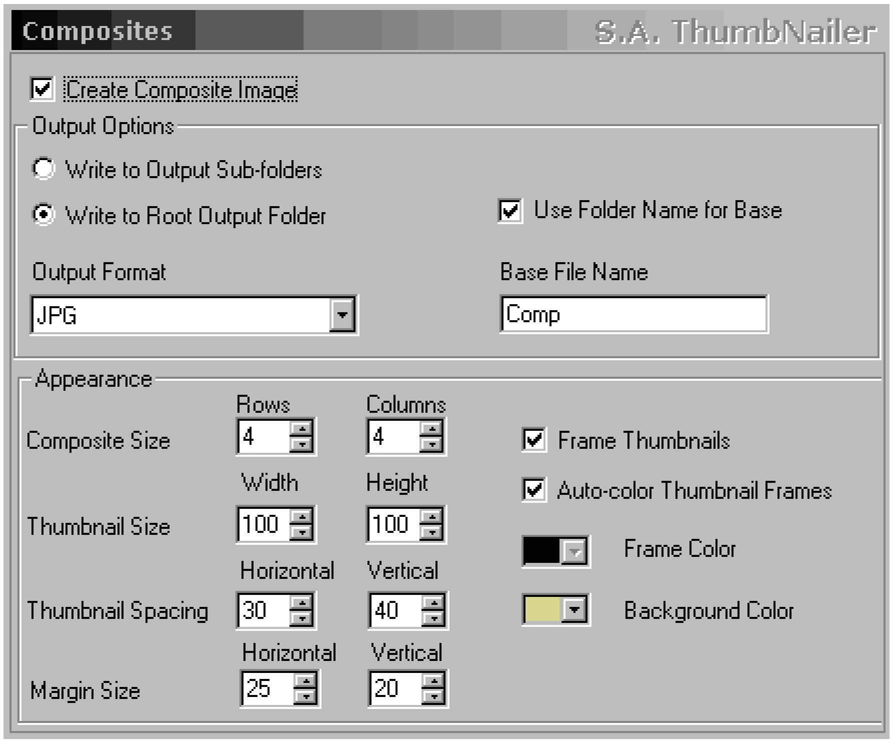
Cell Padding
Set the spacing for the edges of the cells.

Additional TABLE Text
This is text that goes inside the <TABLE> tag. It is added before the Width, Border Width, Cell Spacing and Cell Padding tags.

Additional TD Text
This is text that goes inside the <TD> tag. It is inserted every time a new cell is started.

Additional TR Text
This is text that goes inside the <TR> tag. It is inserted every time a new row is started.

Composite



☑ Create Composite Image

Output Options

○ Write to Output Sub-folders

● Write to Root Output Folder

☑ Use Folder Name for Base

Output Format: JPG

Base File Name: Comp

Appearance

	Rows	Columns
Composite Size	4	4
	Width	Height
Thumbnail Size	100	100
	Horizontal	Vertical
Thumbnail Spacing	30	40
	Horizontal	Vertical
Margin Size	25	20

☑ Frame Thumbnails

☑ Auto-color Thumbnail Frames

■ Frame Color

▨ Background Color

(Advanced / Composite Images)

The program will create composite, contact-sheet, images of all output images in a batch. Output images are arranged on a grid in the composite image. It is sometimes useful to turn off Thumbnail creation when creating composites (see Operation)

Create Composite Image
If this is checked, the program will create composite images.

Output Options

Write to Output Subfolders
The composite images will be written to the subfolders in which the output images reside.

Write to Root Output Folder
The composite images will be written to the destination root directory.

Base File Name
Set the base name from which all composite image names will be created. New files names are created by adding a numeric suffix to the base name. The numeric suffix resets to 0 for each new output folder.

Use Folder Name for Base
Instead of using the Base File Name for the file name, this option causes the program to use the current folder name.

Output Format
Choose the file format to for the composite images.

Appearance

Composite Size
How many thumbnail images will make up this composite?

Rows
Specify the number of rows in the composite images

Columns
Specify the number of columns in the composite images

Thumbnail Size
Choose the width and height of the individual thumbnails for this composite image. Thumbnails will be sized to fit into these dimensions. This size, along with the number of columns and rows, and the spacing between the thumbnails, determines the final size of the composite image.

Thumbnail Spacing
Set the number of pixels between thumbnails.

Note:
The vertical spacing is in addition to the space required for any label text you have specified. This is different from previous versions.

Margins
This controls the distance from the edge of the composite image to the start of the thumbnails.

Frame Thumbnails
Check this if you want your composite thumbnails to appear in beveled boxes.

Auto-color Thumbnail Frames
Allow the program to choose the colors for the thumbnail frames.

Thumbnail Frame Color
Choose the thumbnail frame color, if not using Auto-color.

Background Color
Choose the background color for the composite images

Text Options
Launch the Composite Text Options dialog.

How to make the best possible contact sheets:

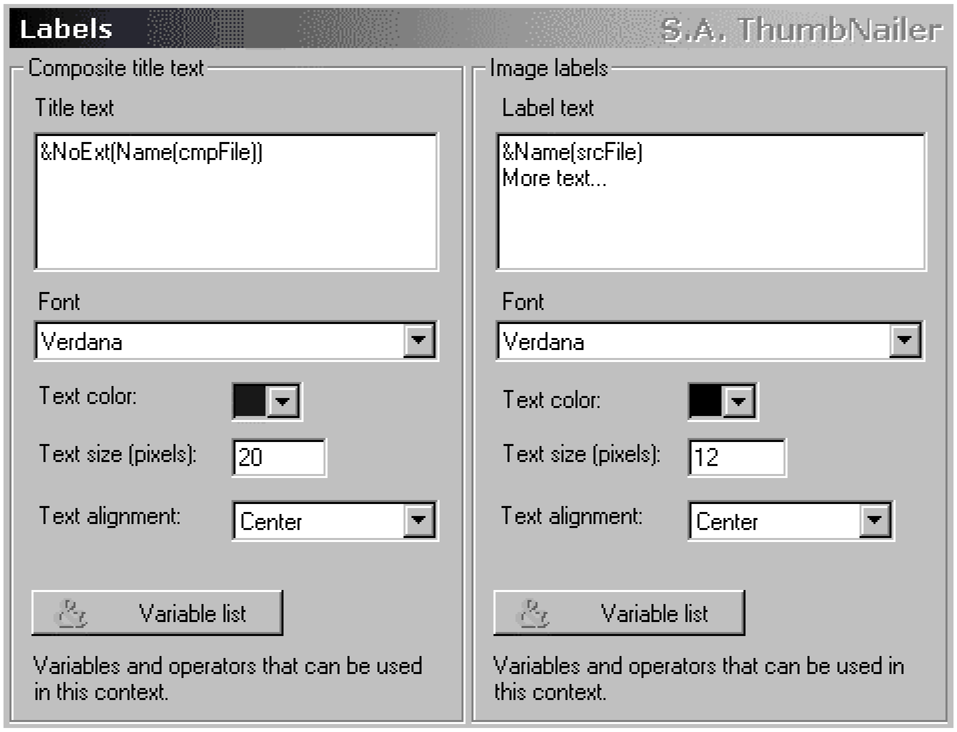
For the best composite results:

1. Turn off thumbnail creation (see above).
2. Set the image resizing method resizing method to Lanczos, Catrom or Mitchell.
3. Set the main resize method to “No Resize”.

The reasons are:

1. The images on the composite images are generated from the thumbnail images, after all effects and resizing has been applied. So, if your input image is 800x800 and you are generating 50x50 thumbnails, the composite image will be resized from the 50x50 thumbnail, not the 800x800 original. And, it’s always best to resize from the original. So, if you choose “No Resize”, the composite image will be generated from the original image.
2. The resizing methods listed above give the best results. They are many times slower than the other resizing methods, but the results are far superior.

Composite Text



Composite title text

Title text

```
&NoExt(Name(cmpFile))
```

Font: Verdana

Text color: ■

Text size (pixels): 20

Text alignment: Center

Variables and operators that can be used in this context.

Image labels

Label text

```
&Name(srcFile)
More text...
```

Font: Verdana

Text color: ■

Text size (pixels): 12

Text alignment: Center

Variables and operators that can be used in this context.

(Advanced / Composite Images / Labels)

Composite Title Text

The program will add a title to the composite pages.

Title Text
The title text to appear on all composite images.

Font
Choose from the fonts installed on your PC. This is for the composite titles.

Text Color
Choose the color that the text is output as. This is for the composite titles.

Text Size
Specify the height of the text, in pixels. This is for the composite titles.

Text Alignment
Choose from Left, enter or Right justified.

Variables List
This will display a list of the variables that are available for these text fields.

Image Labels

Lable Text
This is the text that appears below each thumbnail image on the composite. Vertical spacing between thumbnails will be adjusted to accommodate multiple lines of text. The text will be wrapped to fit in a box no wider than the thumbnail (or thumbnail frame).

Font Name
Choose from the fonts installed on your PC. This is for the image titles.

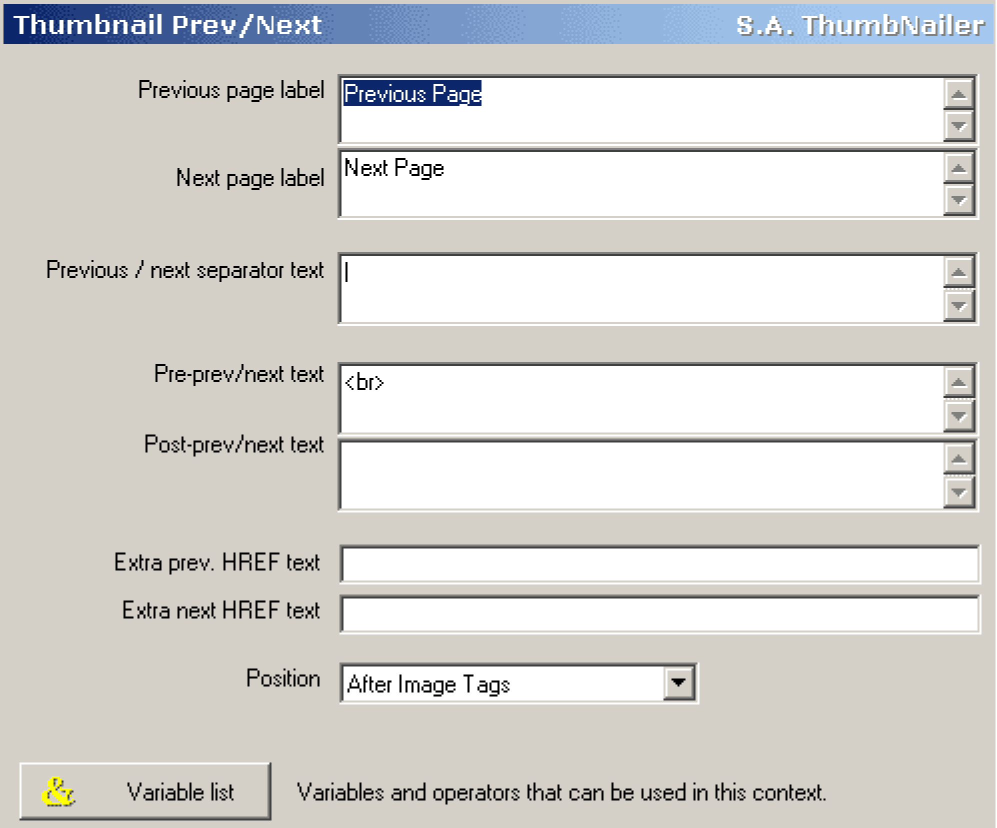
Text Color
Choose the color that the text is output as. This is for the image titles.

Text Size
Specify the height of the text, in pixels. This is for the image titles.

Text Alignment
Choose from Left, enter or Right justified.

Variables List
This will display a list of the variables that are available for these text fields.

Thumbnail Previous / Next Tags



Previous page label: Previous Page

Next page label: Next Page

Previous / next separator text: |

Pre-prev/next text:

Post-prev/next text:

Extra prev. HREF text:

Extra next HREF text:

Position: After Image Tags

Variables and operators that can be used in this context.

(Advanced / HTML Output / Thumbnail Pages / Thumbnail Prev/Next)

Previous Page Label
Choose any text you like. *

Next Page Label
Choose any text you like. *

Previous / Next Separator Text
By default, the program puts a vertical bar “|” between the text for the Previous and Next links. But, you can choose any text you like. *

Pre-Prev/Next Text
This will be placed immediately before each Previous / Next text on the thumbnail pages. *

Pre-Prev/Next Text
This will be placed immediately following each Previous / Next text on the thumbnail pages. *

Extra Prev HREF Text
This text will be placed inside the HREF for the “previous” link, after the URL.

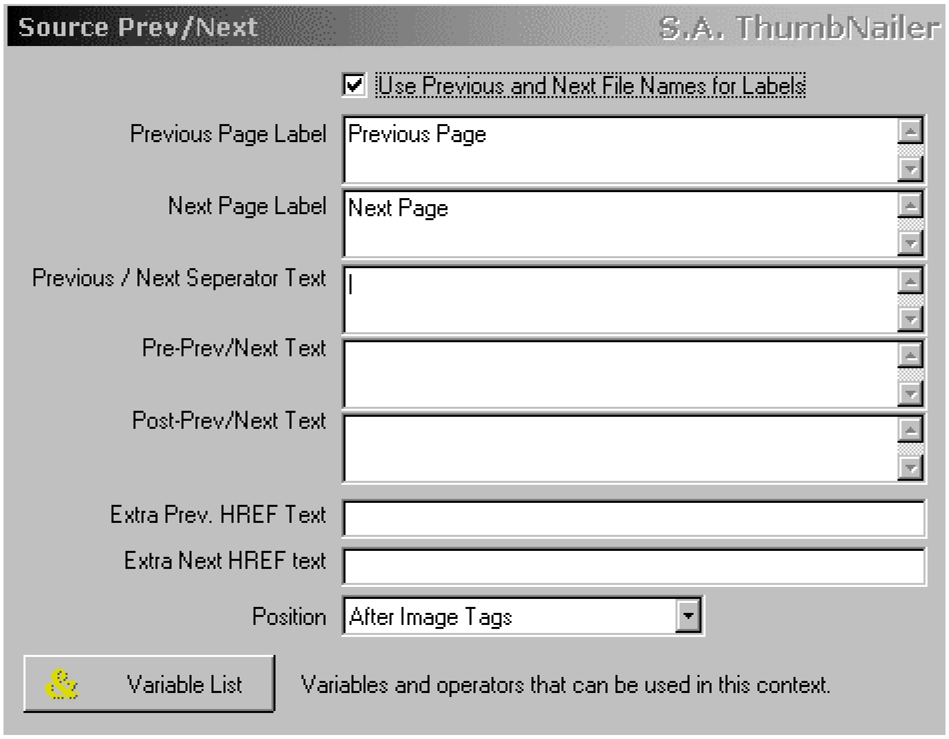
Extra Next HREF Text
This text will be placed inside the HREF for the “next” link, after the URL.

Position
Choose the location of the Previous / Next tags within the thumbnail pages.

Variables List
This will display a list of the variables which are available for these text fields.

**** Tip:*** *Right click on the control and choose "Bigger" from the menu, to enter your text in a larger window.*

Source Previous / Next Tags



☑ Use Previous and Next File Names for Labels

Previous Page Label: Previous Page

Next Page Label: Next Page

Previous / Next Seperator Text: |

Pre-Prev/Next Text:

Post-Prev/Next Text:

Extra Prev. HREF Text:

Extra Next HREF text:

Position: After Image Tags

Variables and operators that can be used in this context.

(Advanced / HTML Output / Source Pages / Source Prev/Next)

Previous Page Label
Enter any text you like. *

Next Page Label
Enter any text you like. *

Previous / Next Separator Text
By default, the program puts a vertical bar "|" between the text for the Previous and Next links. But, you can choose any text you like. *

Pre-Prev/Next Text
This will be placed immediately before each Previous / Next text on the source pages. *

Pre-Prev/Next Text
This will be placed immediately following each Previous / Next text on the source pages. *

Extra Prev HREF Text
This text will be placed inside the HREF for the "previous" link, after the URL.

Extra Next HREF Text
This text will be placed inside the HREF for the "next" link, after the URL.

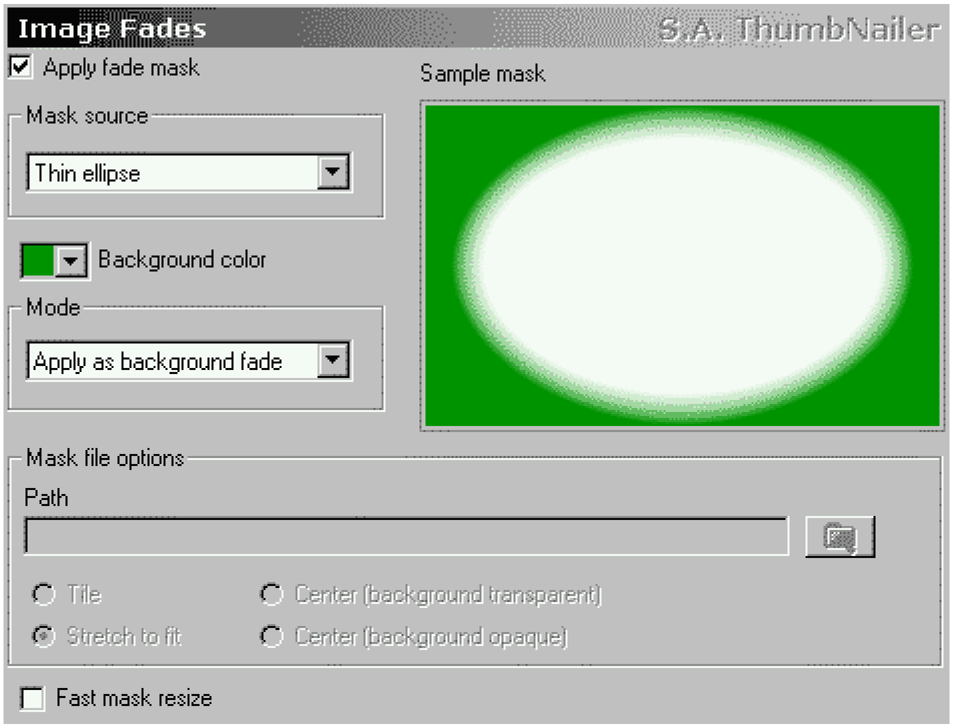
Position
Choose the location of the Previous / Next tags within the source pages.

Variables List
This will display a list of the variables which are available for these text fields.

***** ***Tip:*** *Right click on the control and choose "Bigger" from the menu, to enter your text in a larger window.*

Image Fades

(Advanced / Image Fades)



☑ Apply fade mask

Mask source: Thin ellipse

🟩 Background color

Mode: Apply as background fade

Path:

○ Tile ○ Center (background transparent)

● Stretch to fit ○ Center (background opaque)

□ Fast mask resize

The program can automatically apply fades to the edges of images. The basic idea is that one image is used as a mask over another image. If you use the predefined fade types (either of the elliptical or either of the rectangular fades), the program will construct the mask image for you.

But, you are also able to use your own images as mask sources. This allows you a great deal of control over the effect.

Example: If you use the predefined Thick Rectangle fade, your source image will start with black edges then slowly fade in to the full source brightness towards the center of the image.

Apply Fade
Apply a fade to all output images.

Mask Source
Thin Rectangle, Thick Rectangle, Thin Ellipse, Thick Ellipse
You can choose from these pre-defined shapes.

From a File
If you want to use a fade mask that you create on your own, check this.

Background Color
Choose a color that the image will fade into.

Mode
Apply as background fade
This is the traditional ThumbNailer fade mode. The fade is applied as a mask that fades from the image color to the background color.

Apply as transparency mask
If you have selected an output image type that supports an alpha (transparency) channel, the mask will be used to control the image transparency. Currently, ThumbNailer supports alpha for 32-bit PNG and 32-bit TIFF images. Internet Explorer (as of version 5.5) will render 32-bit PNG images with alpha channels by drawing them on a solid background (you can set this background color in ThumbNailer's Advanced / Format Options / PNG screen), this is, in effect, the same as if you had used the "Apply as background fade" option. Adobe Photoshop, however, will correctly import these PNG with the alpha channel truly controlling transparency: you can overlay these images onto other images and the alpha channel will control how the images blend together. Likewise, Photoshop will read these TIFFs with proper transparency.   We have not tested other image programs.

Mask File Options

File
Choose the file you wish to use as the mask. For best results, this should be a grayscale image.

Stretch to Fit
The mask image will be resized to fit each output image.

Tile
The mask image will be tiled over each output image.

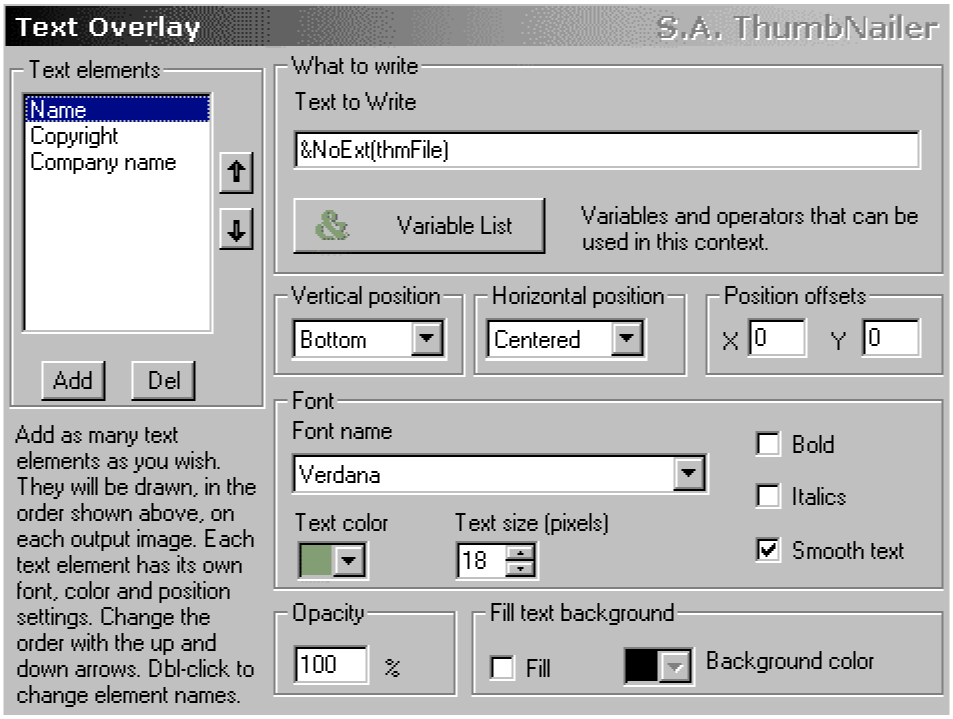
Center, Background Transparent
The mask image will be centered on the output images and any area not covered by the mask image will be shown normally.

Center, Background Opaque
The mask image will be centered on the output images and any area not covered by the mask image will be totally masked.

Notes on using your own masks
All masks are converted to grayscale before being applied. To avoid surprises, create your masks as grayscale images.

A white pixel in a mask means the source image shows through at 100% brightness. A black pixel in a mask hides the source pixel completely.

Text Overlay



Text elements: Name, Copyright, Company name

Text to Write: &NoExt(thmFile)

Variables and operators that can be used in this context.

Vertical position: Bottom

Horizontal position: Centered

Position offsets: X 0 Y 0

Font name: Verdana

☐ Bold

☐ Italics

☑ Smooth text

Text color

Text size (pixels): 18

Opacity: 100 %

☐ Fill

Background color

Add as many text elements as you wish. They will be drawn, in the order shown above, on each output image. Each text element has its own font, color and position settings. Change the order with the up and down arrows. Dbl-click to change element names.

(Advanced / Text Overlay)

This feature causes the program to draw text onto all output images. You can use it to draw copyright info, captions, time/date stamps, etc. on your thumbnail images.

Text Elements
This is a list of text elements that you have created. Each element is a piece of text that will be drawn on each output image. Each element is independent of all other elements. They are drawn in the order that they appear in this list. ThumbNailer can draw an unlimited number of text elements on each output image.

Up/Down
To control when each text element is drawn, you can move it up and down in the list. Select the item you wish to move and press the up or down arrow to move it accordingly. Elements near the top of the list are drawn before elements below.

Add
Add a new text element to the bottom of the list.   When you add a new element, its name will be “New Text”. To change the name of a text element, double click on the name in the list and type a new name.

Del
Remove the currently-selected text element from the list.

What to write
Text to Write
Draw this text on the image. This text will appear on all output images. Note that the space for this text is very limited, so be careful when specifying text. See variables for information on using operators and variables. You can test your expressions in the expression tester

Variables List

This will display a list of the variables that are available for this text field.

Vertical Position / Horizontal Position

Choose where you want the text to go. If text does not fit, it will run off the edge.

Font Name

Choose from the fonts installed on your PC.

Text Color

Choose the color the text.

Text Size

Specify the height of the text, in pixels.

Smoothing

Draw the text with edges smoothed.

Bold, Italics

Draw text as bold and/or italics.

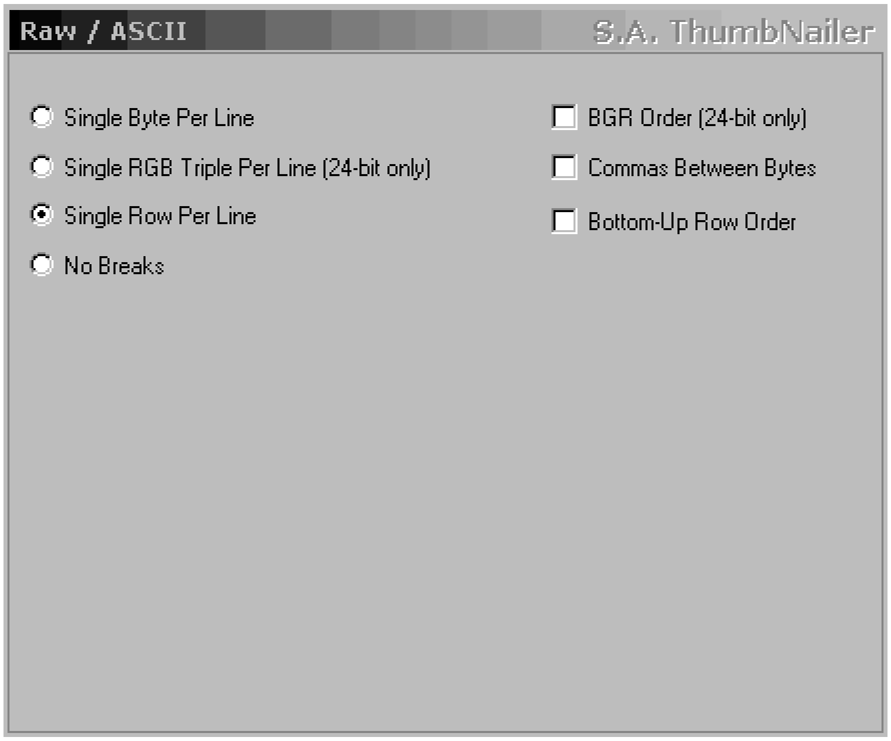
Opacity

This controls the opacity/transparency of the text. A value of 100 will produce fully opaque (solid) text. A value of 0 will produce completely transparent (and therefore invisible) text.

Background

The program can fill the area behind the text, to make reading easier; choose a color.

Raw



○ Single Byte Per Line

○ Single RGB Triple Per Line (24-bit only)

● Single Row Per Line

○ No Breaks

□ BGR Order (24-bit only)

□ Commas Between Bytes

□ Bottom-Up Row Order

The program can output images to text files as raw ASCII values, formatted according to the styles you set.

Single Byte Per Line
One output Byte is written per line.

Single RGB Triple Per Line
The three values that make up an RGB triple are written, one per line :

123, 44, 221

Single Row Per Line
An entire row of pixels are written on a single line

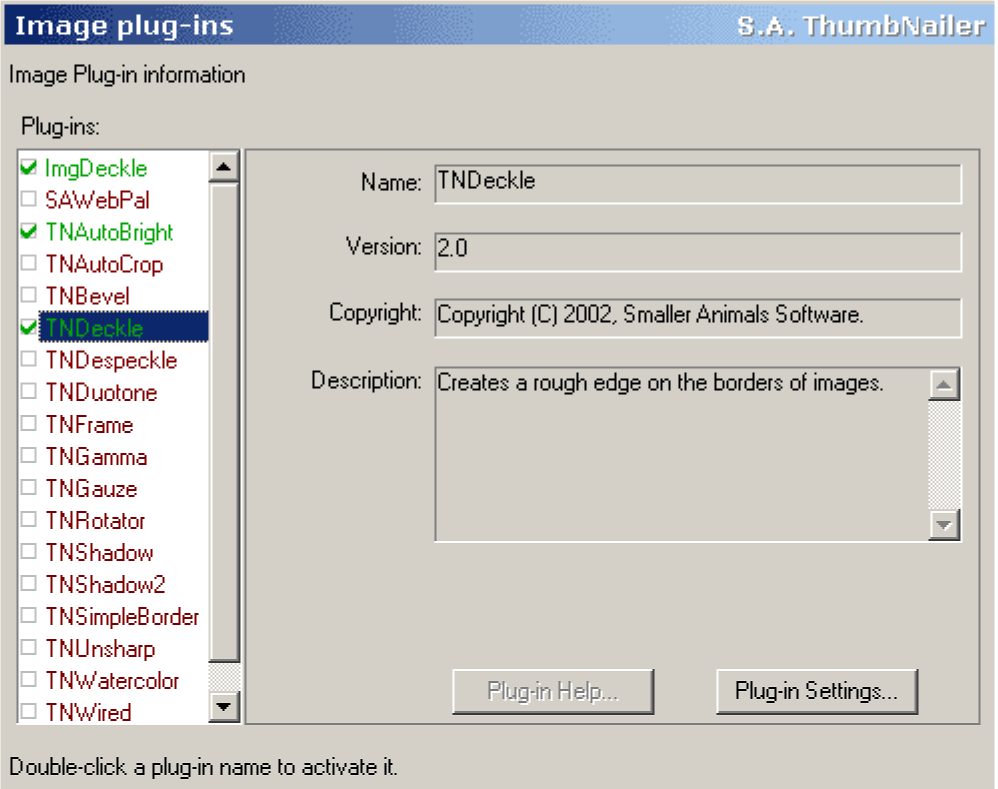
No Breaks
The values are written without line breaks

BGR Order
RGB Triples are written in BGR order

Commas Between Bytes
A comma (“,”) is placed after each number, except at the end of a line. Note that this has no effect if there is only one byte per line.

Bottom-up Row Order
Rows are written in reverse order (bottom image row is the first output row).

Plug-ins



- ☑ ImgDeckle
- □ SAWebPal
- ☑ TNAutoBright
- □ TNAutoCrop
- □ TNBevel
- ☑ TNDeckle
- □ TNDespeckle
- □ TNDuotone
- □ TNFrame
- □ TNGamma
- □ TNGauze
- □ TNRotator
- □ TNShadow
- □ TNShadow2
- □ TNSimpleBorder
- □ TNUnsharp
- □ TNWatercolor
- □ TNWired

Name: TNDeckle

Version: 2.0

Copyright: Copyright (C) 2002, Smaller Animals Software.

Description: Creates a rough edge on the borders of images.

(Advanced / Plug-ins)

Plug-ins allow you to extend ThumbNailer's built-in capabilities by adding special image effects or adding support for more file formats. Plug-ins come in three basic varieties: image processing, file import and multi-frame import.

See the ThumbNailer home page for the list of plug-ins that are currently available.

Note: As of v7, ThumbNailer no longer supports HTML plug-ins.

Image Processing Plug-ins
The program has many image enhancement functions built-in: sharpening, contrast, LUT, overlay, etc.. These functions are useful, but they are hard-coded into the program. If you don't like the built-in sharpening function, what can you do? You could try to get the sharpening function in the program changed (not likely); you could import all of your images, one-by-one, into another image processing program, like PhotoShop (no fun); or you could use a S.A. ThumbNailer Image Processing plug-in that someone else has written that does a better job (in your opinion, of course) at sharpening.

The program allows IP (Image Processing) plug-ins to modify the current image at many places in the main processing loop. The possibilities are (almost) infinite. Also, because the IP plug-in has access to the pixels in image, it could potentially save the image in a format that the program does not even support.

For programmers, see the Smaller Animals ThumbNailer home page for plug-in tutorials and sample code, in both Visual Basic and Visual C++.

Image Import Plug-ins
These plug-ins allow ThumbNailer to read additional image file formats.

This plug-in type was introduced in v7.0.

These plug-ins provide ThumbNailer with one or more file extensions to search for. When a batch is started, ThumbNailer searches for files that match the file extensions on the Input Filter page, then it searches for files with the file extensions used by the selected import plug-ins. When it reads an image file, ThumbNailer first tries to read it using its built-in format support; if it is unable to read the file, the plug-ins are given a chance to read it.

Multi-framed Image Import Plug-ins
These plug-ins allow ThumbNailer to read image files that contain more than one image per file: movies, and video clips, for example. ThumbNailer has built-in support for some multi-framed image types (AVI, GIF and TIFF), but these plug-ins can extend that to include other file formats, such as MPG, WMV, ASF, etc..

Like the Image import plug-ins (above), These plug-ins provide ThumbNailer with one or more file extensions to search for. When a batch is started, ThumbNailer searches for files that match the file extensions on the Input Filter page, then it searches for files with the file extensions used by the selected import plug-ins. When it reads an image file, ThumbNailer first tries to read it using its built-in format support; if it is unable to read the file, the plug-ins are given a chance to read it.

This plug-in type was introduced in v8.0.

Plugins Folder
The plug-ins listed are those that are located in the Plugins folder (this must be a sub-folder of the folder where Thumb8.exe lives. You must create it, when you install your first plug-in). Only a plug-in that correctly responds to the program's attempts to initialize it will work.

When you get a new plug-in, you should place it in your Plugins folder. Then, you must *register* it. VB users should be familiar with registering OLE/COM components. For everyone else, just use the RegSvr32 button on the Plug-ins dialog.

Note : ThumbNailer plug-ins must reside on the same machine that Thumb8.exe is on. If the plug-in is on a different machine, the result will be a crash of some kind, probably on the machine that is hosting the plug-in, possibly on the Thumb8.exe machine, as well.

Plug-ins lists
This is a list of plug-ins in the Plugins folder. You can select multiple plug-ins. Double-click a plug-in name to select or deselect it. Selected plug-ins will have a check mark next to them in the list box.

You can arrange the order that the plug-ins will be executed in by dragging the plug-in names in the lists. Only plug-ins that are selected (ie. Plug-ins with a check next to their name) will be executed.

Settings
If a plug-in supports a settings dialog, this button will be enabled.

Help
If a plug-in supports help, this button will be enabled.

Re-Scan
This will cause the program to scan the Plugins folder again.

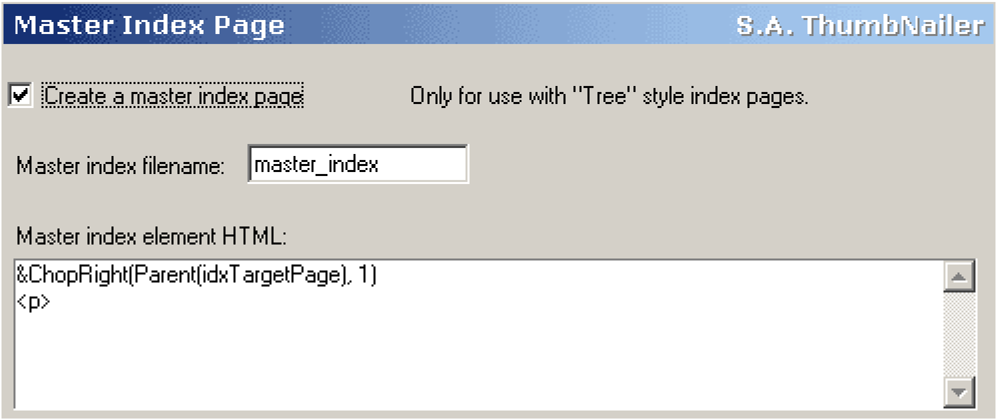
RegSvr32
Plug-ins, like all other OLE Automation components, must be registered on your computer. The usual way to do this is with a program called RegSvr32. This button allows you to select a plug-in DLL for registration.

AutoReg
Scan for plugins in the Plugins folder and attempt to run RegSvr32 on each of them. You will have to

confirm the success or failure of each registration attempt.

Developers:
Tutorials and samples for creating your own plug-ins are available on the Smaller Animals ThumbNailer web site.

Master Index Page



☑ Create a master index page   Only for use with "Tree" style index pages.

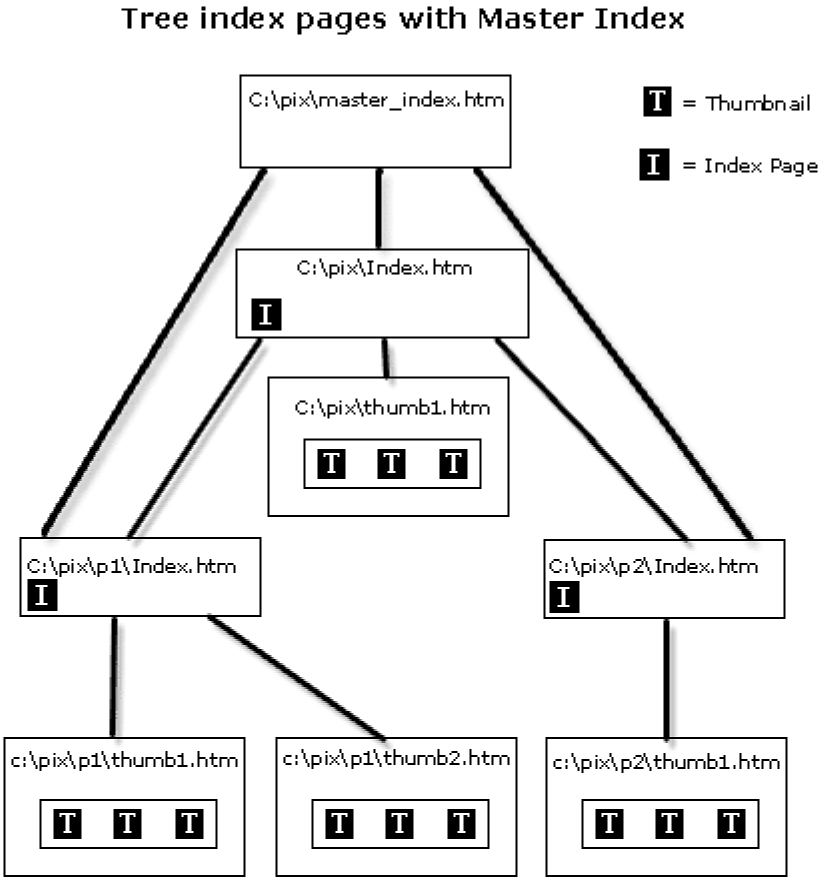
Master index filename: master_index

Master index element HTML:

```
&ChopRight(Parent(idxTargetPage), 1)
<p>
```

(Advanced / HTML Output / Index Pages / Master Index Page)

When creating index pages in the "tree" style (where there is an index page in each sub-folder listing all of the thumbnail pages in that folder, as well as linking to all sub-folder index pages), you can have ThumbNailer create a single page which contains links to all of those sub-folder index pages.



This can only be used with the "tree" style index pages, because, in the "simple" index page style, there is only one index file total anyway.

By default, each link on the master index page contains the thumbnail of the first image in the folder and the name of that folder as a label. The label can be changed below. To change the rest of the page, you can use an HTML Template.

Master Index Filename
This is the base filename for the master index page. The index HTML file file extensionwill be added automatically.

Master index element label
This is used to create the label for each index file link. The default is:

```
&ChopRight(Parent(idxTargetPage), 1)
<p>
```

That gives the name of the folder in which the target index file resides.

Diagnostic Mode

After an abnormal termination, ie., a crash, the program will prompt you to enter diagnostic mode the next time it starts up. In diagnostic mode, the program will generate a detailed log of its internal operations. This log may be of use to Smaller Animals Software in determining the cause and solution to the problem. The program will tell you the location of the log file, but it will always be in the same folder as Thumb6.exe and be named "ThumbLog.txt". You may examine this file, if you like, but its contents will likely be meaningless to you.

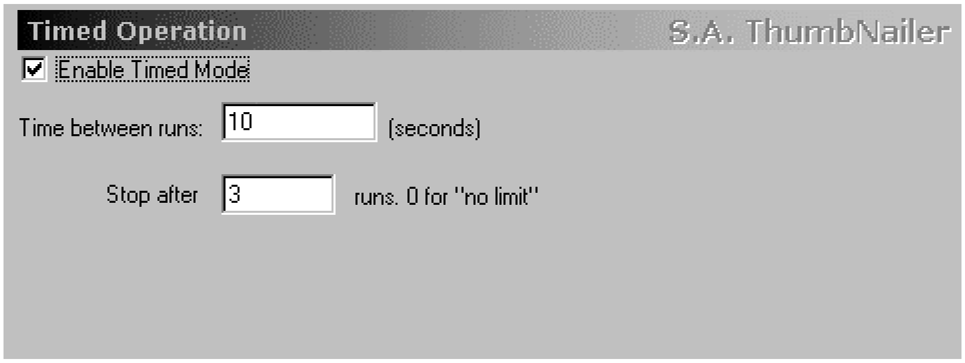
When the program prompts you to enter diagnostic mode, you can choose "Cancel" or "OK". If you choose Cancel, the program will operate as usual. But, if you choose "OK", the program will immediately start writing entries into its log file. In order to reproduce the problem, you should use the program exactly as you did before the crash. Then, if the program crashes, you should immediately email the log file to Smaller Animals Software : smallest@smalleranimals.com. If the program does not crash, do not send the log file.

The program will run slightly slower in diagnostic mode.

If you have to prevent the program from prompting you to enter diagnostic mode, you must delete the file "ThumbLock.lck" from the directory in which Thumb6.exe runs. The program creates this file on startup and deletes this file during a normal shutdown. If the program sees this file on startup, it assumes that there was an abnormal shutdown the last time it ran.

Disabling
To disable this lock file checking, run the program with the "/nolock" command line switch.

Timed Mode



☑ Enable Timed Mode

Time between runs: 10 (seconds)

Stop after 3 runs. 0 for "no limit"

(Advanced / Timed Mode)

In “Timed Mode”, ThumbNailer will automatically process a folder based on a timer. You set the frequency to rescan (in seconds) and ThumbNailer will do the rest.

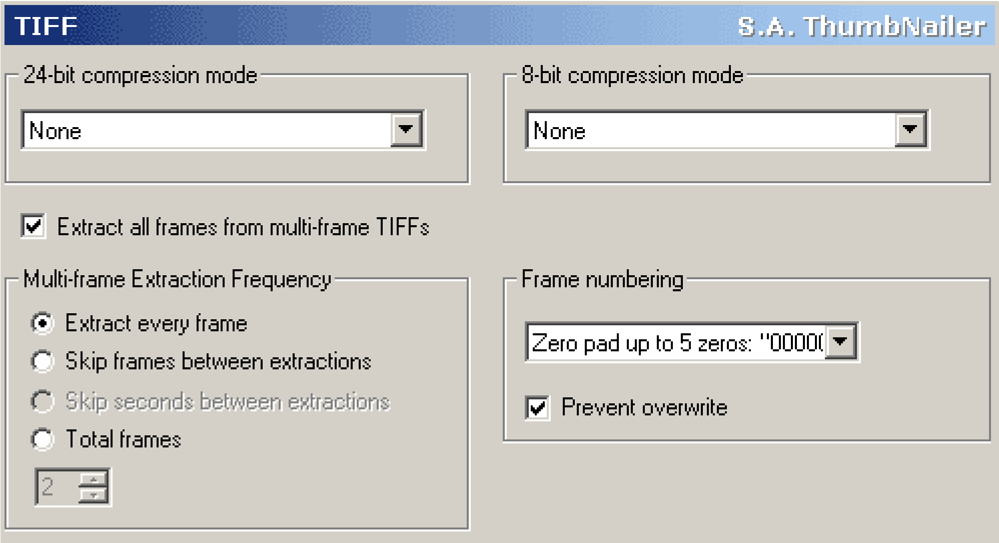
Note: This is a new and very advanced feature. It is recommended that you only use it if you have a definite use for such functionality.

Enable Timed Mode
If you check this, ThumbNailer will run in Timed Mode. The ThumbNailer Start button will start the timer, and turn into a “Stop” button, so you can interrupt the timer.

Time Between Runs
This is the timer frequency. You shouldn’t use very short times (less than 5 seconds). Very short times may mean that ThumbNailer won’t have time to finish a run before the timer goes off to start the next one.

Stop After xxxx Runs
This allows you to limit the number of time ThumbNailer will run by itself. If you leave this at zero, it has no effect.

TIFF Options



24-bit compression mode

None

8-bit compression mode

None

☑ Extract all frames from multi-frame TIFFs

Multi-frame Extraction Frequency

● Extract every frame
○ Skip frames between extractions
○ Skip seconds between extractions
○ Total frames

2

Frame numbering

Zero pad up to 5 zeros: "0000(

☑ Prevent overwrite

(Advanced / Format Options / TIFF)

24-bit Compression Mode
None - This is the default. No compression is applied.

JPEG - The JPEG-in-TIFF compression scheme is used. This provides good compression for photographic images.

PackBits - This is a simple run-length encoding scheme. JPEG or Deflate are much better compression choices. PackBits is only provided for completeness.

Deflate - This is the same compression scheme that is used in PNG files. It provides excellent compression for images with large areas of solid color.

LZW – Use the LZW compression method.

8-bit Compression Mode
Same as above. JPEG cannot be used on 8-bit images.

Extract all frames from multi-frame TIFFs
TIFF files can contain multiple images per file. If you check this box, the program will extract all images in multi-page TIFFs.

Extraction Frequency
These controls allow you to specify how many frames to extract from multi-page TIFFs.

S.A. ThumbNailer always extracts frame 0 from the TIFF file.

Extract every frame
Extracts all frames.

Skip frames between extractions
Enter the number of frames that you wish to skip between frames that it extracts. Ex. If you give a value of 4, the program will extract frame 0, skip 1,2,3 and 4, then it will extract frame 5, etc..

Skip seconds between frames
TIFF files do not use a concept of time for the multiple frames. This option is unavailable.

Total Frames (start with first)
Enter the total number of frames that you wish the program to extract from the TIFF. The program will extract this number of frames, evenly-spaced throughout the file, starting with the first frame of the file. If you extract one frame, it will be the first frame; if you extract 3, you will get the first frame, the frame at 33% and the frame at 66%.

Total Frames (centered)
Enter the total number of frames that you wish the program to extract. The program will extract this number of frames, evenly-spaced throughout the file. Frames will be selected based on the center: if you extract one frame, it will be the middle frame. If you extract 3, you will get frames at 25%, 50% and 75%.

Frame Numbering

All extracted TIFF frames are named in this manner :

1. The base name from the TIFF is used as the base name for the frame. Ex. the base of Bear.TIFF is “Bear”.
1. The renaming options from the File / Dir tab are applied (prefix , suffix, numeric suffix). Ex. “BearThumb”
1. The TIFF frame number is added to the end of the name : Ex. BearThumb0002
1. The output file format extension is added. Ex. BearThumb0002.JPG

***Note** : For TIFF frame extraction, it is recommended that you do **not** use the Auto-Numeric Suffix renaming style from the File / Dir options tab. The Auto-Numeric Suffix option will cause unpredictable results when used with the numeric suffixes added in step 3, above.*

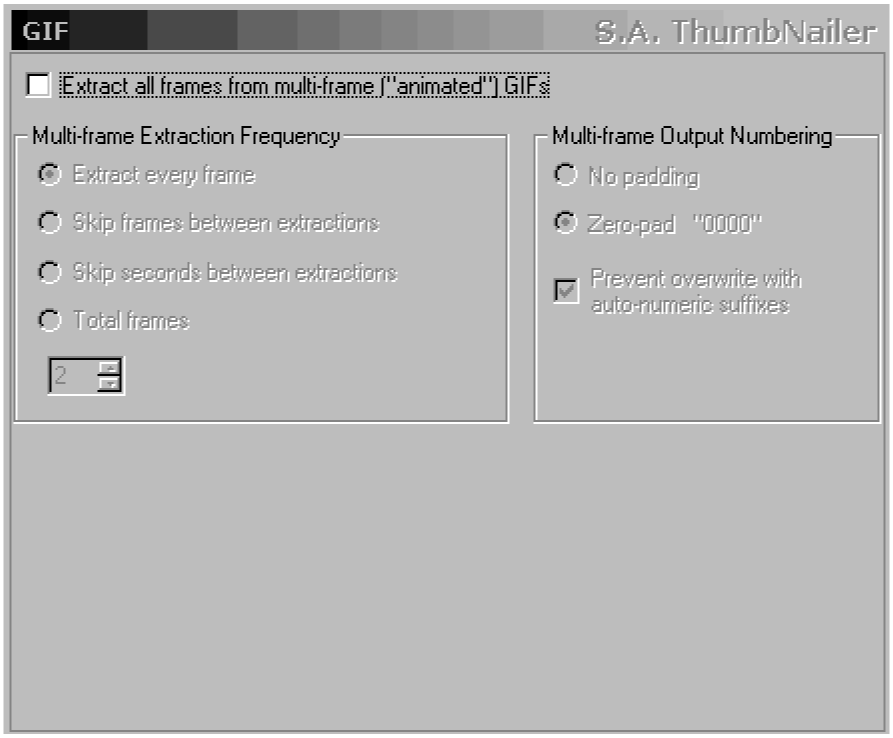
Zero-pad “0000”
The numbers added to the ends of the base TIFF name are padded with at least three zeros. Ex. An TIFF with three frames, Bear.TIFF, will generate Bear0000.jpg, Bear0001.jpg and Bear0002.jpg. This implies a limit of 10,000 frames per base TIFF.

No pad
No zeros are added to the output frame name. Ex. A TIFF with three frames, Bear.TIFF, will generate Bear0.jpg, Bear1.jpg and Bear2.jpg.

Prevent Overwrite
If this is checked, the program will re-number output frames so that they don’t overwrite any files that may already be in the output folder. This could lead to files with numbers at the end of their filename which are not their actual frame number. Internally, the program uses the Auto-Numeric Suffix generator to prevent output file overwriting.

If Prevent Overwrite is **not** checked, the numbers at the ends of frame filenames will be the same as their frame number. **Any files already in the output folder will be overwritten in case of a name collision.**

GIF Options



□ Extract all frames from multi-frame ("animated") GIFs

Multi-frame Extraction Frequency

- ● Extract every frame
- ○ Skip frames between extractions
- ○ Skip seconds between extractions
- ○ Total frames
- 2

Multi-frame Output Numbering

- ○ No padding
- ● Zero-pad "0000"
- ☑ Prevent overwrite with auto-numeric suffixes

(Advanced / Format Options / GIF)

Extract all frames from multi-frame GIFs
GIF files can contain multiple images per file. If you check this box, the program will extract all images in multi-frame GIFs.

Extraction Frequency
These controls allow you to specify how many frames to extract from multi-page GIFs.

S.A. ThumbNailer always extracts frame 0 from the GIF file.

Extract every frame
Extracts all frames.

Skip frames between extractions
Enter the number of frames that you wish the program to skip between frames that it extracts. Ex. If you give a value of 4, the program will extract frame 0, skip 1,2,3 and 4, then it will extract frame 5, etc..

Skip seconds between frames
Not all animated GIF files use a concept of time for the multiple frames. This option is unavailable.

Total Frames (start with first)
Enter the total number of frames that you wish the program to extract from the GIF. The program will extract this number of frames, evenly-spaced throughout the GIF, starting with the first frame of the file. If you extract one frame, it will be the first frame; if you extract 3, you will get the first frame, the frame at 33% and the frame at 66%.

Total Frames (centered)
Enter the total number of frames that you wish the program to extract from the GIF. The program will

extract this number of frames, evenly-spaced throughout the GIF. Frames will be selected based on the center: if you extract one frame, it will be the middle frame. If you extract 3, you will get frames at 25%, 50% and 75%.

Frame Numbering

All extracted GIF frames are named in this manner:

1. The base name from the GIF is used as the base name for the frame. Ex. the base of Bear.GIF is "Bear".
2. The renaming options from the File / Dir tab are applied (prefix , suffix, numeric suffix). Ex. "BearThumb"
3. The GIF frame number is added to the end of the name : Ex. BearThumb0002
4. The output file format extension is added. Ex. BearThumb0002.JPG

Note *: For GIF frame extraction, it is recommended that you do* ***not*** *use the Auto-Numeric Suffix renaming style from the File / Dir options tab. The Auto-Numeric Suffix option will cause unpredictable results when used with the numeric suffixes added in step 3, above.*

Zero-pad "0000"

The numbers added to the ends of the base GIF name are padded with at least three zeros. Ex. An GIF with three frames, Bear.GIF, will generate Bear0000.jpg, Bear0001.jpg and Bear0002.jpg. This implies a limit of 10,000 frames per base GIF.

No pad

No zeros are added to the output frame name. Ex. A GIF with three frames, Bear.GIF, will generate Bear0.jpg, Bear1.jpg and Bear2.jpg.

Prevent Overwrite

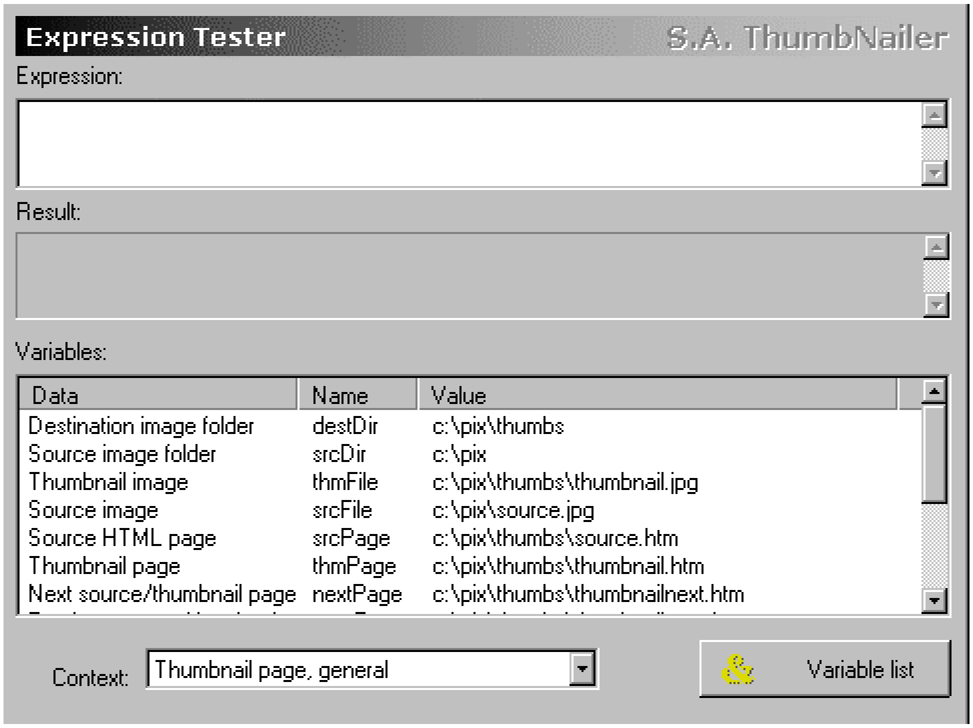
If this is checked, the program will re-number output frames so that they don't overwrite any files that may already be in the output folder. This could lead to files with numbers at the end of their filename which are not their actual frame number. Internally, ThumbNailer uses the Auto-Numeric Suffix generator to prevent output file overwriting.

If Prevent Overwrite is **not** checked, the numbers at the ends of frame filenames will be the same as their frame number. **Any files already in the output folder will be overwritten in case of a name collision.**

Prevent Overwrite
If this is checked, the program will re-number output frames so that they don’t overwrite any files that may already be in the output folder. This could lead to files with numbers at the end of their filename which are not their actual frame number. Internally, the program uses the Auto-Numeric Suffix generator to prevent output file overwriting.

If Prevent Overwrite is **not** checked, the numbers at the ends of frame filenames will be the same as their frame number. **Any files already in the output folder will be overwritten in case of a name collision.**

Expression Tester



Data	Name	Value
Destination image folder	destDir	c:\pix\thumbs
Source image folder	srcDir	c:\pix
Thumbnail image	thmFile	c:\pix\thumbs\thumbnail.jpg
Source image	srcFile	c:\pix\source.jpg
Source HTML page	srcPage	c:\pix\thumbs\source.htm
Thumbnail page	thmPage	c:\pix\thumbs\thumbnail.htm
Next source/thumbnail page	nextPage	c:\pix\thumbs\thumbnailnext.htm

Context: Thumbnail page, general

(Advanced / Expression Tester)

To help in creating expressions for HTML, composites and text overlays, the Expression Tester page will allow you to create expressions and see the results immediately: without having to run a batch of images. Just enter your expression, and ThumbNailer will print the results as you type. See variables for more information about variables, operators and expressions.

The expression tester window, once you've started it, will remain open until you close it or shut down ThumbNailer. This way, you can test expressions easily while working with other windows (like the template editor or the various HTML fields).

Expression
Enter your expression text here.

Tip: *Right click on the control and choose "Bigger" from the menu, to enter your text in a larger window.*

Result
This is the result from the evaluation of your expression. It is updated as you change the expression text, the context or any of the variables.

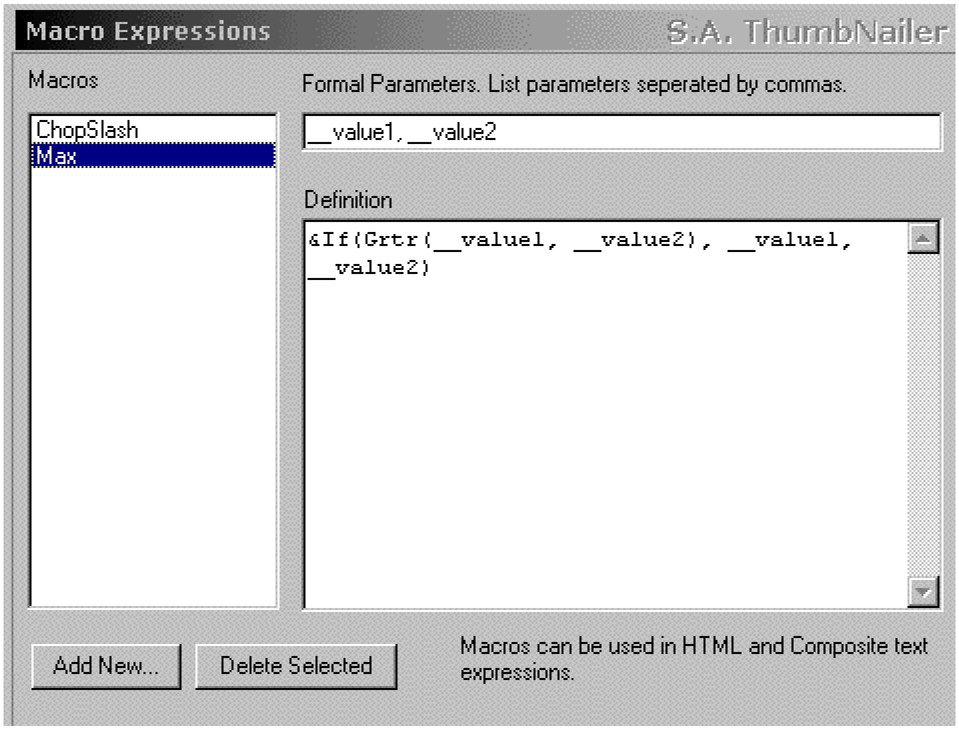
Variables
ThumbNailer provides you with default values for all of the pieces of data that go into expressions. You can change these values if you wish, to see how your expression reacts with different data. Just double click on the value you want to change and enter the new value. Your expression will be re-evaluated immediately.

Context
You can also choose the context in which the expression is evaluated: certain variables are only available in certain contexts. Ex. the "thmFile" variable isn't available on certain places on thumbnail pages, and not at all on index pages. This way, you can verify that your expression works in the

desired context.

Variable List
If you press this button, ThumbNailer will show you a list of variables that are applicable to the context you have selected, along with a list of all operators.

Macro Expressions



Macro Expressions — S.A. ThumbNailer

Macros

- ChopSlash
- Max

Formal Parameters. List parameters seperated by commas.

__value1, __value2

Definition

```
&If(Grtr(__value1, __value2), __value1,
__value2)
```

Macros can be used in HTML and Composite text expressions.

(Advanced / Expressions / Macro Expressions)

One of the more advanced features of the ThumbNailer HTML and composite expression evaluator is the ability to use user-defined “macro expressions”. The immediate benefit of using macro expressions in your HTML expressions is to save typing. At their very simplest, a macro is just a way of representing a complex expression with a short name.

Why use macros?
If you want to list the source name of each source image underneath the corresponding thumbnail image on the thumbnail HTML pages (Advanced / HTML Output / Thumbnail Tags / Thumbnail Tags), you can use the default expression that ThumbNailer provides: `&Name(srcFile)`.

But, if your file names are sometimes very long and you need to limit the length of the text printed to control the appearance of your HTML documents, you can use an expression like `&Left(Name(srcFile), 15)`. This will only print the first 15 characters of the source file name. That’s a quick and easy way to do it. However, the results might not be very pretty – long names are simply chopped off after 15 characters, and there is no indication that the real name is longer than the printed name. A better solution would be to add “...” to the end of the first 15 characters of the name, if the name is greater than 15 characters long.

An expression to do this is:

```
&If(Grtr(Len(Name(srcFile)), 15) , Join(Left(Name(srcFile), 15), “...”) ,
Name(srcFile))
```

This says “If the length of the file name, no path, of the source file is greater than 15, output the first 15 characters of the file name, no path, of the source file, followed by “...”. If the length is not greater than 15, output the file name, no path, of the source filename.”

The obvious problem with an expression like this is its length. If you only have to use it in one place, then you can probably just type it in and forget about it. But. if you want to do the same thing in multiple HTML sections, you will have to copy and paste this expression everywhere you need it. This is probably acceptable, in some cases, too; but expressions like these tend to take up a lot of the limited space available on the HTML text pages.

Even worse: what if you want to do this text limiting with other file names, the thumbnail file name, for example. You would have to copy and paste this expression, then carefully replace each occurrence of “srcFile” with “thmFile”. Things will start to get very ugly, if you have many of these huge expressions in your text fields, each modifying a different file name.

The solution to this problem, of course, is macro expressions.

Create a macro
On the Advanced / HTML Output / Macros page, add a new macro called “_ChopPath1”:

Formal Parameter List: `@text@`
Definition:

```
(If(Grtr(Len(Name(@text@)), 15) , Join(Left(Name(@text@), 15), “...”) , Name(@text@))
```

Notes:

- We’ve named our macro ChopText1. The leading “_” is to help lessen the chance that our macro’s name will be the same as an existing operator, or a future operator. There will never be a pre-defined operator with a name beginning with “_”.

- We’ve called our formal parameter “`@text@`” to help minimize the chance that our parameter name will be the same as any of the pre-defined operator names. Someday, there could be an operator called “text”, but there will never be one called “`@text@`”. You should take similar precautions when writing your own macros.

Then, in the Image Label Text field ((Advanced / HTML Output / Thumbnail Pages / Thumbnail Tags / Image Label Text), your expression is simply `<br>&_ChopPath1(srcFile)`.

How does it work?
Each time this particular macro expression is evaluated, ThumbNailer will replace each occurrence of the formal parameter “`@text@`” in the macro with the user parameter “srcFile”.

So, this

```
(If(Grtr(Len(Name(@text@)), 15) , Join(Left(Name(@text@), 15), “...”) , Name(@text@))
```

...becomes this :

```
(If(Grtr(Len(Name(srcFile)), 15) , Join(Left(Name(srcFile), 15), “...”) ,
Name(srcFile))
```

Then, this expression is evaluated.

Notes:

- You can use ChopPath1 anywhere you want to limit the length of the text that is printed for a file name. `&_ChopPath1(thmFile)` , `&_ChopPath1(htmFile)` , `&_ChopPath1(“c:\\path\\file.ext”)` , etc..

- A macro behaves just like a new operator. You can use macros inside other operators: `&Upper(_ChopPath1(srcFile))` and vice versa: `&_ChopPath1(Upper(srcFile))`.

One Step Further...
Create a new macro, _ChopText2:

Formal Parameter List: `@text@, @len@`
Definition:

```
(If(Grtr(Len(Name(@text@)), @len@), Join(Left(Name(@text@), @len@),
“...”),Name(@text@))
```

With _ChopText2, you can control the maximum length of the text that is printed before the “…”, like this :
`&_ChopText2(srcFile, 10)`. This will print the first 10 characters of the source image file name before adding “…”. Likewise, `&_ChopText2(srcFile, 5)` will print only the first five characters before chopping.

One more…
Create a new macro, _ChopText3:

Formal Parameter List: `@text@, @len@, @ext@`
Definition:

```
(If(Grtr(Len(Name(@text@)), @len@), Join(Left(Name(@text@), @len@), @ext@),
Name(@text@))
```

Now, you can control the text that is used to signal that the text has been chopped.
`&_ChopText3(srcFile, 5, “...”)` is equivalent to `&_ChopText2(srcFile, 5)`. But, `&_ChopText3(srcFile, 5, “*”)` will use “*” instead of “…”, at the end of the chopped text.

More notes:

- Macro expressions behave like operators. You can nest them inside other macros or inside pre-defined operators.
- When choosing a name for your macro, be sure to avoid using any of the pre-defined operator names. Starting your macro names with “_” is a good idea.
- When choosing names for formal parameters in a macro expression, be sure to use names that will not conflict with current or future operators or macros. “@param@” and “##param” are all good choices for naming styles.
- Macro definitions must begin with “&“ followed by an operator or macro name then “(“. Macro definitions must end with “)”. Ex. &Upper(@p1)
- Parameters in formal parameter lists must be separated by a comma.
- There is no upper or lower limit to the number of formal parameters allowed.

You can test your macros in the expression tester

Also, see Macros in Templates

HTML Templates

ThumbNailer's built-in HTML setup options give you a lot of control over the look of you final HTML pages; you can control colors, fonts, label text, headers, footers, etc.. But, the program still determines the overall layout; for example, all thumbnail pages follow this basic layout:

HTML header stuff
User-defined header
Thumbnails
User-defined footer

While this is adequate for many purposes, it doesn't give you total control over the layout. ThumbNailer HTML templates, on the other hand, allow you to control the exact placement of nearly everything on the HTML page.

ThumbNailer HTML templates work by substitution; basically, you create a skeleton HTML page with sections marked off in which ThumbNailer later adds image data.

An example
The simplest of all ThumbNailer templates is the source page template. Here's a generic source page example:

```
# source page definition
@:sourcepage
@+
<html>
<body>
<center>
    <a href=&Q(RelFile(thmPage, srcPage))>
    <img src=&Q(RelFile(srcFile, srcPage)) width=&Q(XDim(srcFile))
height=&Q(YDim(srcFile))>
    </a>

    <br>

<font face="Arial">
    &Name(srcFile)
    </font>

    <font face="Arial">
    @@sourceprevpage @@sourcepnsep @@sourcenextpage
    </font>
</center>
</body>
</html>
@-
```

Let's examine this template line-by-line.

```
# source page definition
```

The first line is a comment. ThumbNailer ignores any line, outside of a section definition, that starts with a "#" (I'll explain what a "section" is below). You can add any text you want to a comment line.

The second line starts the "sourcepage" section.

```
@:sourcepage
```

A “section” is a block of text that ThumbNailer treats as a single unit. Most section can contain other sections. When ThumbNailer builds a page, it combines all of the different sections that it needs forthat page. The names and purposes of the sections are fixed, but the contents are up to you. ThumbNailer supports a fixed number of sections, some are required and some are optional.

The “sourcepage” section is what Thumbnailer starts with when it needs to create a source HTML page.

The next line starts the section text:

```
@+
```

This tells ThumbNailer that everything that follows this line belongs to the section was just started (in this case, the “sourcepage” section).

The rest of the text, down to the “@-“ is a combination of HTML and ThumbNailer expressions – everything, that is, except this line:

```
@@sourceprevpage @@sourcepnsep @@sourcenextpage
```

This line handles the Previous / Next tags. It references three other sections: “sourceprevpage”, “sourcepnsep” and “sourcenextpage”. The “@@sourceprevpage” text is a substitution point – ThumbNailer will replace this text with the text from the “sourceprevpage” section. Likewise, “@@sourcepnsep” and “@@sourcenextpage” will be replaced by the text from those sections.

We’ll define these other sections as follows:

```
# Source page source page link
@:sourceprevpage
@+
<a href=&Q(RelFile(prevPage, srcPage))>Previous Page</a>
@-

# Source page previous / next separator
@:sourcepnsep
@+
|
@-

# Source page next page
@:sourcenextpage
@+
<a href=&Q(RelFile(nextPage, srcPage))>Next Page</a>
@-
```

Just like the sourcepage, each section starts with a “@:section name” line, then a “@+” to start the actual section text, and a “@-“ to end it. And, just like the sourcepage section, these sections are combinations of plain HTML and ThumbNailer expressions.

How does it work?
When ThumbNailer needs to build a source page, it follows these steps:

1. Get a copy of the source page template: the sourcepage, sourceprevpage, sourcepnsep and sourcenextpage sections.
2. Run each section through the expression evaluator. At this point, the variables, “srcFile”, “thmFile”, etc., contain information about the current images and HTML pages; so when the expressions are evaluated, the template goes from a generic HTML page to a page that deals with specific files.
3. If there is a “previous” file, place the sourceprevpage section at the appropriate location(s) in

the sourcepage section.

4. If there is a previous *and* a "next" file, place the "sourcepnsep" section. (you only need a separator if there is both a previous and a next).
5. If there is a next file, place the "sorucenextpage" section.
6. Write the whole thing to the output HTML file.

There are templates that deal with thumbnail and index pages. While these other templates are somewhat more complicated than the source page template (more substitution points), the basic idea is the same.

Section names are fixed – you can't make up your own sections. ThumbNailer will report an error if it encounters a section that it doesn't recognize.

Here is the list of template sections, what they do, the variables you can use in each and the substitution points each section supports.

To actually create a template, you define the appearance of all of the required sections one after the other, in the same text file. Then, tell ThumbNailer to use that file from the main HTML Generation page.

To get a feel for how to create your own templates, or to start a new one, you can export the results from the built-in HTML setup options to an external template file. In fact, this is exactly what ThumbNailer does when you don't use an external template – it takes your options from the HTML Setup pages and generates a temporary template to use. When you save that template to an external file, you will see *exactly* what ThumbNailer uses.

thumbnailgroup

A simple @@thumbnailpage section might look like this:

```
@:thumbnailpage
@+
<html>
<head><title>Thumbnails</title></head>
<body bgcolor="#ffffc8" link="#fa0064" alink="#ffff00" vlink="#6400fa" text="#000000">

<center>

<table>
<tr><td>@@thumbnail</td><td>@@thumbnail</td></tr>
<tr><td>@@thumbnail</td><td>@@thumbnail</td></tr>
</table>
<hr>
<p>
@@thumbprevpage@@thumbpnsep@@thumbnextpage
</center>
</body>
</html>
@-
```

That's just four thumbnails in a 2x2 table, a horizontal line, and then the previous/next page stuff. What happens if you run out of thumbnails, after only filling two of those four @@thumbnail points? You get a table with two empty cells and two thumbnails – a whole blank row. Now imagine a larger table (5x5 maybe). Having that many empty table cells can really mess up a nice page layout. It would be nice if there was a way to get rid of the extra table rows.

So, to help with this problem, ThumbNailer v8 introduces a new section called "@@thumbnailgroup".

A @@thumbnailgroup section goes on the @@thumbnailpage section and contains one or more

@@thumbnail sections.

```
@:thumbnailpage
@+
<html>
<head><title>Thumbnails</title></head>
<body bgcolor="#ffffc8" link="#fa0064" alink="#ffff00" vlink="#6400fa" text="#000000">
<center>

<table>
@@thumbnailgroup
@@thumbnailgroup
</table>

<hr>
<p>
@@thumbprevpage@@thumbpnsep@@thumbnextpage
</center>
</body>
</html>
@-
```

And the @@thumbnailgroup section might look like this:

```
@:thumbnailgroup
@+
<tr><td>@@thumbnail</td><td>@@thumbnail</td></tr>
@-
```

What does this accomplish? Well, now, if you run out of thumbnails after the first two, ThumbNailer will have filled out the entire first thumbnailgroup section and placed it in the output file, and it will just erase the next thumbnailgroup substitution point. The HTML file will only have a single row of thumbnails., and no blank cells.

Index pages and their groups

Index pages contain groups, too: idxchildgroup, idxsubidxgroup, masteridxgroup. These are similar to the thumbnailgroup as described above, but with one important difference: as soon as ThumbNailer fills one of these index page groups, it puts it in the template and starts another one immediately following the one it just finished. This way, these pages can be arbitrarily long – it just keeps adding new groups until it runs out of images to place. This is different from the thumbnailpage’s thumbnailgroup where there are a fixed number of them on the page, and when they fill up, a new page starts.

Template Section Definitions

Section Name	Description	Variables Supported	Substitution Points
Thumbnail Pages			
thumbnail	A single thumbnail element.	srcFile, thmFile, srcDir, destDir, thmPage, srcPage	None
altthumbnail	Optional alternate thumbnail section. This can be used anywhere a "thumbnail" section can. It provides a way to have a second thumbnail look – for variety.	srcFile, thmFile, srcDir, destDir, thmPage, srcPage	None
defaultthumbnail	Used instead of the "thumbnail" section to complete a thumbnail page when there are no more thumbnails left in the folder or batch.	srcDir, destDir, thmPage, prevPage, nextPage	None
thumbnailpage	The thumbnail page. Contains any number of "@@thumbnail" substitution points and any number of the prev/next substitution points. When all of the "@@thumbnail" points have been replaced, a new thumbnail file is started.	srcDir, destDir, thmPage, prevPage, nextPage	@@thumbnail @@thumbprevpage @@thumbpnsep @@thumbnextpage @@thumbnailgroup
thumbprevpage	Used to do a link to the previous thumbnail page. Is set to blank text when there is no previous page.	srcDir, destDir, thmPage, prevPage, nextPage	None
thumbpnsep	Used to separate the previous and next sections when both are present. Set to blank when one or the other prev/next file doesn't exist.	srcDir, destDir, thmPage, prevPage, nextPage	None
thumbnextpage	Used to do a link to the next thumbnail page. Is set to blank text when there is no next page.	srcDir, destDir, thmPage, prevPage, nextPage	None
thumbnailgroup **	Contains one or more @@thumbnail points. It can be used to group thumbnails, or to minimize the number of "defaultthumbnail" sections that your page will have. Any thumbnailgroup points not filled by thumbnails will be removed from the page. If your template uses this section, your @@thumbnailpage section must not have any @@thumbnail points.	srcDir, destDir, thmPage, prevPage, nextPage	@@thumbnail @@thumbprevpage @@thumbpnsep @@thumbnextpage @@thumbnailgroup

Source Pages

sourcepage	The source HTML page. Contains any number of the source prev/next substitution points. One of these is generated for the source file from each thumbnail.	srcFile, thmFile, srcDir, destDir, thmPage, srcPage, prevPage, nextPage, prevFile, nextFile	@@sourceprevpage @@sourcepnsep @@sourcenextpage
sourceprevpage	Used to do a link to the previous source page. Is set to blank text when there is no previous page.	srcFile , thmFile, srcDir, destDir, thmPage, srcPage, prevPage, nextPage, prevFile, nextFile	None
sourceprevpage	Used to do a link to the next source page. Is set to blank text when there is no next page.	srcFile, thmFile, srcDir, destDir, thmPage, srcPage, prevPage, nextPage, prevFile, nextFile	None
sourcepnsep	Used to separate the previous and next sections on the source page when both are present. Set to blank when one or the other prev/next file doesn’t exist.	srcFile, thmFile, srcDir, destDir, thmPage, srcPage, prevPage, nextPage, prevFile, nextFile	None

Index Pages

indexpage	The index page. If you’re doing a “single” index page, this should contain one “@@idxchildgroup” substitution point and neither of the other two. If you’re doing a “tree” style index, this should contain each of the three substitution points. ThumbNailer will add completed @@idxchildgroup and @@idxsubidxgroup sections until all of the available index/thumbnail page references have been placed.	srcDir, destDir, IdxPage, parentPage*	@@idxchildgroup @@idxsubidxgroup @@idxsectionseparator
idxchildgroup	Contains one or more “@@idxchildelement” substitution points. ThumbNailer adds as many these sections as is necessary to account for all of the thumbnail HTML pages referenced by the current index page.	srcDir, destDir, idxPage, parentPage*, idxTargetPage, idxTargetThm, idxTargetSrc	@@idxchildelement
idxchildelement	References a single thumbnail page. When completed, this section is placed in a @@idxchildgroup section.	srcDir, destDir, idxPage, parentPage*, idxTargetPage, idxTargetThm, idxTargetCur, idxTargetSrc	None
defaultidxchildelement	Completes the @@idxchildgroup by replacing @@idxchildelement points when there are no more thumbnail page references for the current index page.	srcDir, destDir, idxPage, parentPage*	None

idxsubidxgroup*	Contains one or more “@@idxsubidxgroup ” substitution points. ThumbNailer adds as many these sections as is necessary to account for all of the index HTML pages referenced by the current index page.	srcDir, destDir, idxPage, parentPage, idxTargetPage, idxTargetThm, idxTargetSrc	@@idxsubidxelement
idxsubidxelement*	References a single index page. When completed, this section is placed in a @@idxchildgroup section.	srcDir, destDir, idxPage, parentPage, idxTargetPage, idxTargetThm, idxTargetSrc	None
defaultidxsubidxelement*	Completes the @@ idxsubidxgroup by replacing @@ idxsubidxelement points when there are no more thumbnail page references for the current index page.	srcDir, destDir, idxPage, parentPage, idxTargetPage, idxTargetThm, idxTargetSrc	None
idxsectionseparator*	Separates the @@idxchildgroup and the @@idxsubidxgroup sections.	srcDir, destDir, idxPage, parentPage	None
masterIdxPage* **	Defines the HTML file for the Master index of all “tree” index files.	srcDir, destDir, idxPage	@@masteridxgroup
masteridxgroup * **	Contains one or more @@masteridxelement points. ThumbNailer adds as many masteridxgroup sections as is necessary to account for all of the tree index HTML pages created.	srcDir, destDir, idxPage	@@ masteridxelement
masteridxelement * **	References a single tree index page. When completed, this section is placed in a @@masteridxgroup section.	srcDir, destDir, idxPage, idxTargetPage, idxTargetThm, idxTargetSrc	None
General			
Macros	This is where you can define macros to use on the template. Macros defined here can only be used on this template. No HTML output is generated by this section. See Macros in Templates .	N/A	N/A
Readme	A comment section. Any text in here is displayed to users when they validate the template.	N/A	N/A
Initialize	This section is evaluated before all others. Any variables defined here (with VarSet) will be available in any other HTML page section. No HTML output is generated by this section.	srcDir, destDir	N/A

* Denotes sections and variables that are only applicable when you’re using the “tree” index style. These sections are not used and will be blanked-out when using the “single” index style.

** Available in ThumbNailer Version 8 only.

EXIF Data Tags

ThumbNailer can read the EXIF data stored in JPG images that were taken with a digital camera. This data can be placed on HTML pages (or composites) with the “EXIF” operator. This operator takes two inputs; the first is the filename of the image, the second is the tag number. The tag number tells ThumbNailer which EXIF data to read and interpret.

Examples:
`&Exif(srcFile, 272)` -> “FinePix1400Zoom” (Camera Model)
`&Exif(srcFile, 306)` -> “2000:12:11 18:49:48” (DateTime)

Note: If the tag ID you requested was not in the file (most tags are optional), ThumbNailer will output “…” instead.

Below is a list of some EXIF data tag IDs. The list is not complete (there are over 65,000 tag IDs available), but it does list most of the common tags. Some descriptions are just a “?”, because no description was available. Not all tags will be meaningful or useful, but are shown here for completeness.

This table is based on: From http://www.butaman.ne.jp/~tsuruzoh/Computer/Digicams/exif-e.html

Tag ID	Description.
2	InteroperabilityVersion : Records the interoperability version.
254	NewSubfileType : ?
255	SubfileType : ?
256	ImageWidth : Shows size of thumbnail image.
257	ImageLength
258	BitsPerSample : When image is not compressed, this value shows the number of bits per component for each pixel. Usually this value is '8,8,8'
259	Compression : Image compression method.
262	PhotometricInterpretation : color space of the image data components.
270	ImageDescription : text
271	Make : Camera manufacturer.
272	Model : Camera model number.
273	StripOffsets : When image not compressed, this value shows offset to image data. In some case image data is striped and this value is plural.
274	Orientation : The orientation of the camera relative to the scene.
277	SamplesPerPixel : When image is not compressied, this value shows the number of components stored for each pixel. For color images, this value is '3'.
278	RowsPerStrip : When image is not compressed and is stored as strips, this value shows how many rows were stored in each strip. If image is not stripped, this value is the same as ImageLength
279	StripByteCounts : When image is not compressed and is stored as strips, this value shows how many bytes were used for each strip. If image is not stripped, this is the whole data size of image.
282	XResolution : Display/Print resolution of image. Default value is 1/72inch.
283	YResolution
284	PlanarConfiguration : When image is not compressed (YCbCr), this value shows the byte alignment of YCbCr data. If value is '1', Y/Cb/Cr value is interleaved, contiguous. If value is '2', image data is planar.
296	ResolutionUnit : N/A, Inch, cm
301	TransferFunction : ?
305	Software : Shows the firmware (internal software of camera) version number.
306	DateTime : Date/Time of image was last modified. Data format is "YYYY:MM:DD HH:MM:SS"+0x00, total 20 characters. If clock has not been set or if the camera doesn't have clock, the field may be filled with spaces. Usually, this has the same value as DateTimeOriginal.

315 Artist : ?

317 Predictor : ?

318 WhitePoint : Defines chromaticity of white point of the image. If the image uses CIE Standard Illumination D65 (known as international standard of 'daylight'), the values are '3127/10000,3290/10000'.

319 PrimaryChromaticities : Defines chromaticity of the primaries of the image. If the image uses CCIR Recommendation 709 primaries, values are '640/1000,330/1000,300/1000,600/1000,150/1000,0/1000'.

322 TileWidth : ?

323 TileLength : ?

324 TileOffsets : ?

325 TileByteCounts : ?

330 SubIFDs : ?

347 JPEGTables : ?

513 JpegIFOffset : When image format is JPEG, this value shows offset to JPEG data.

514 JpegIFByteCount : When image format is JPEG, this value shows data size of JPEG image.

529 YCbCrCoefficients : When image format is YCbCr, this value shows constants used to translate it to RGB format, usually, '0.299/0.587/0.114'.

530 YCbCrSubSampling : When image format is YCbCr and uses subsampling(cropping of chroma data, all the cameras do this), this value shows how many chroma data values were subsampled. First value shows horizontal, next value shows vertical subsample rate.

531 YCbCrPositioning : When image format is YCbCr and uses 'Subsampling'(cropping of chroma data, all the cameras do this), this defines the chroma sample point of subsampling pixel array. '1' means the center of pixel array, '2' means the datum point.

532 ReferenceBlackWhite : Shows reference value of black point/white point. In case of YCbCr format, first 2 show black/white of Y, next 2 are Cb, last 2 are Cr. In case of RGB format, first 2 show black/white of R, next 2 are G, last 2 are B.

4096 RelatedImageFileFormat : Records the file format of image file. (e.g. "Exif JPEG Ver. 2.1").

4097 RelatedImageLength : ?

33421 CFARepeatPatternDim : ?

33422 CFAPattern : ?

33423 BatteryLevel : ?

33432 Copyright : text

33434 ExposureTime : Exposure time (reciprocal of shutter speed). Unit is seconds.

33437 FNumber : The actual F-number(F-stop) of lens when the image was taken.

33723 IPTC/NAA : ?

34675 InterColorProfile : ?

34850 ExposureProgram : Exposure program that the camera used when image was taken.

34852 SpectralSensitivity : ?

34853 GPSInfo : ?

34855 ISOSpeedRatings : CCD sensitivity equivalent to Ag-Hr film speedrate.

34856 OECF : ?

34857 Interlace : ?

34858 TimeZoneOffset : ?

34859 SelfTimerMode : ?

36864 ExifVersion : Exif version number.

36867 DateTimeOriginal : Date/Time of original image taken.

36868 DateTimeDigitized : Date/Time when image was digitized. Usually, the same as DateTimeOriginal.

37121 ComponentsConfiguration : Shows the order of pixel data. Usually '0x04,0x05,0x06,0x00' is used for RGB-format and '0x01,0x02,0x03,0x00' for YCbCr-format. 0x00:does not exist, 0x01:Y, 0x02:Cb, 0x03:Cr, 0x04:Red, 0x05:Green, 0x06:Bllue.

37122 CompressedBitsPerPixel : The average compression ratio of JPEG (rough estimate).

37377 ShutterSpeedValue : Shutter speed

37378 ApertureValue : The actual aperture value of lens when the image was taken.

37379 BrightnessValue : Brightness of taken subject, unit is APEX. To calculate Exposure(Ev) from BrigtnessValue(Bv), you must add SensitivityValue(Sv). Ev=Bv+Sv Sv=log2(ISOSpeedRating/3.125) ISO100:Sv=5, ISO200:Sv=6, ISO400:Sv=7, ISO125:Sv=5.32.

37380 ExposureBiasValue : Exposure bias(compensation) value of taking picture. Unit is APEX(EV).

37381 MaxApertureValue : Maximum aperture value of lens.

37382 SubjectDistance : Distance to focus point, unit is meters.

37383 MeteringMode : Exposure metering method.

37384 LightSource : Light source, actually this means white balance setting.

37385 Flash : ?

37386 FocalLength : Focal length of lens used to take image, in millimeters.

37387 FlashEnergy : ?

37388 SpatialFrequencyResponse : ?

37389 Noise : ?

37393 ImageNumber : ?

37394 SecurityClassification : ?

37395 ImageHistory : ?

37396 SubjectLocation : ?

37397 ExposureIndex : ?

37398 TIFF/EPStandardID : ?

37520 SubsecTime : Some cameras can take 2~30 pictures per second, but DateTime/DateTimeOriginal/DateTimeDigitized tag can't record the sub-second time. SubsecTime tag is used to record it. For example, DateTimeOriginal = "1996:09:01 09:15:30", SubSecTimeOriginal = "130", Combined original time is "1996:09:01 09:15:30.130"

37521 SubsecTimeOriginal

37522 SubsecTimeDigitized

40960 FlashPixVersion : Stores FlashPix version. If the image data is based on FlashPix formar Ver.1.0, value is "0100".

40961 ColorSpace : Defines Color Space. DCF image must use sRGB color space.

40962 ExifImageWidth : Size of main image.

40963 ExifImageHeight

40964 RelatedSoundFile : If this cameras can record audio data with image, shows name of audio data.

41483 FlashEnergy : ?

41484 SpatialFrequencyResponse : ?

41486 FocalPlaneXResolution : Pixel density at CCD's position. If you have MegaPixel cameras and take a picture by lower resolution(e.g.VGA mode), this value is re-sampled by picture resolution. In such case, FocalPlaneResolution is not same as CCD's actual resolution.

41487 FocalPlaneYResolution

41488 FocalPlaneResolutionUnit : Unit of FocalPlaneXResoluton/FocalPlaneYResolution. none, inch, cm

41492 SubjectLocation : ?

41493 ExposureIndex : Same as ISOSpeedRatings but data is shown as a fraction. Kodak's cameras use this tag instead of ISOSpeedRating.

41495 SensingMethod : Shows type of image sensor unit. '2' means 1 chip color area sensor, most cameras use this type.

41728 FileSource : Indicates the image source. Usually, the image source is digital still camera.

IPTC Data Tags

ThumbNailer can read the IPTC data stored in JPG images. This data is the same data as Adobe Photoshop's “File Info” information. This data can be placed on HTML pages (or composites) with the

“IPTC” operator.

This operator takes four inputs; the first is the filename of the image, the second is the tag number, followed by the section number and the maximum nuber of tags to read. The tag number tells ThumbNailer which IPTC data to read and interpret.

The section number is usually “2”.

Examples:
`&Iptc(srcFile, 120, 2, 1)` = read the caption (id = 120) from section “2”, allow only one.
`&Exif(srcFile, 25, 2, 3)` = read the keywords (id = 25) from section 2, allow up to 3 keywords.

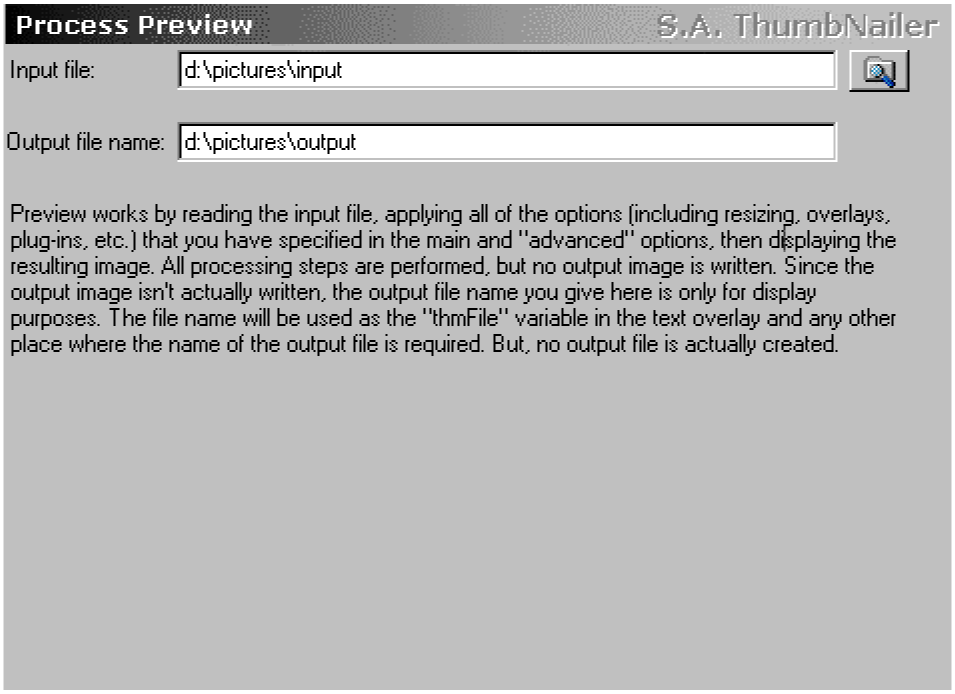
Multiple tags will be separated by commas.

Note: If the tag ID you requested was not in the file (most tags are optional), ThumbNailer will output “…” instead.

Below is a list of some common IPTC data tag IDs.

Tag ID	Description.
5	Origin object name
20	Category
25	Keyword
40	Special instructions
55	Date (in yyyymmdd format)
80	Byline
85	Byline title
90	City
95	State/province
101	Country
103	Original transmission reference
105	Caption header
110	Credit
116	Source
120	Caption
122	Caption writer

Image Process Preview



Input file: d:\pictures\input

Output file name: d:\pictures\output

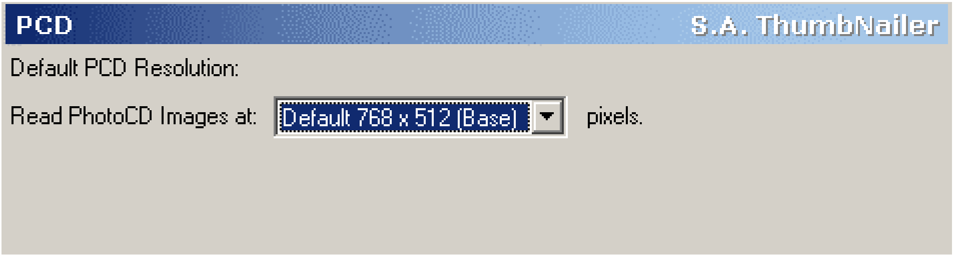
Preview works by reading the input file, applying all of the options (including resizing, overlays, plug-ins, etc.) that you have specified in the main and "advanced" options, then displaying the resulting image. All processing steps are performed, but no output image is written. Since the output image isn't actually written, the output file name you give here is only for display purposes. The file name will be used as the "thmFile" variable in the text overlay and any other place where the name of the output file is required. But, no output file is actually created.

This feature will run a single image through all the image processing steps you've set up, including any selected plug-ins, so that you can get an idea of what your output images will look like. No actual output image is created; your input image is read, processed and displayed for you to look at. It's quicker than re-running an entire batch over and over.

Input File
This is the file to use for the preview. ThumbNailer will read and process this file, then display the result. This file does not have to be in the source folder, or on the same drive, etc.. It can be any file you like.

Output file name
The image process preview doesn't actually create an output file. But, certain features, like text overlays, expect an output file name. For, example, if you have the following expression in a text overlay: "`&Name(thmFile)`", the "thmFile" variable will hold the name of the output (thumbnail) file. So, this field allows you to enter a name for use in situations like this, so you can properly test text overlays. Again, no actual output file is created; the file name simply satisfies any expressions that require the output file path.

PhotoCD Options



Default PCD Resolution:

Read PhotoCD Images at: Default 768 x 512 (Base) pixels.

Default PCD Resolution.
PhotoCD images are stored with multiple sizes (resolutions) of the image in the PCD file. These sizes are:

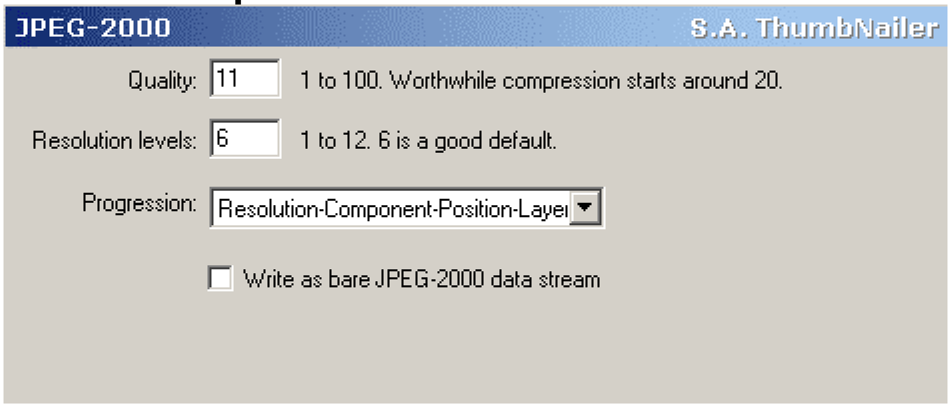
1.	192 x 128	(Base/16)
2.	384 x 256	(Base/4)
3.	768 x 512	(Base)
4.	1536 x 1024	(Base*4)
5.	3072 x 2048	(Base*16)
6.	6144 x 4096	(Base*64)

By default, ThumbNailer will read PhotoCD images at the "Base" resolution of 768x512 pixels. You can choose to override this, if you wish. Why would you want to change this?

1. You want to extract very large "thumbnails" (or convert to large copies in a different format). Image quality is generally better when you are reducing than when enlarging. So if, for example, you wanted to generate 1200x800 images from your PCD images, you will get better results if you tell ThumbNailer to read the PCD images at 1536x1024, than if you keep the default. This happens because the output images will be resized down from 1536x1024 instead of up from 768x512. Be careful, though; there is no benefit to having ThumbNailer read at 3072x2048, if you're reducing to 1200x800. Reading larger images will only cause ThumbNailer to use more memory and take longer to generate the output – there is little chance of better image quality. You only need to read at the smallest resolution that is larger than your desired output size.
2. Likewise, if you only want very small thumbnails, you might choose one of the smaller image sizes, like 384x256. ThumbNailer will be able to read and process the images more quickly.

Future versions of ThumbNailer may be able to anticipate the best PCD image size to read and take some of this decision-making away, but for now, your best bet is to stick with the default resolution unless you have a strong need to change it.

JPEG 2000 Options



Quality: 11 1 to 100. Worthwhile compression starts around 20.

Resolution levels: 6 1 to 12. 6 is a good default.

Progression: Resolution-Component-Position-Layer

□ Write as bare JPEG-2000 data stream

JPEG-2000 is a relatively new format that was designed to store photographic images. It has many features which make it superior to the traditional "JPEG" format; but because it is not widely supported (very few imagine applications support it), many of these features cannot be taken advantage of. So, for now, ThumbNailer provides only very basic JPEG-2000 support. Future versions may expand on this if the format achieves more widespread support.

JPEG-2000 images are stored as a series of images, each of which contains more detail than the previous. In theory, you could request only the sub-images required to satisfy the detail requirements of your application. For example, a quick image preview could be provided by a web server by transmitting only the low detail level sub-images, instead of sending all the detail levels. This could speed up image display. In reality, though, there currently aren't any web servers that can do this.

Quality

Like JPG, JPEG-2000 offers a quality versus compression tradeoff: the higher the compression, the lower the image quality. A good starting value is "20".

Resolution levels

Controls the number of sub-image resolution levels stored in the file. 6 is a good default.

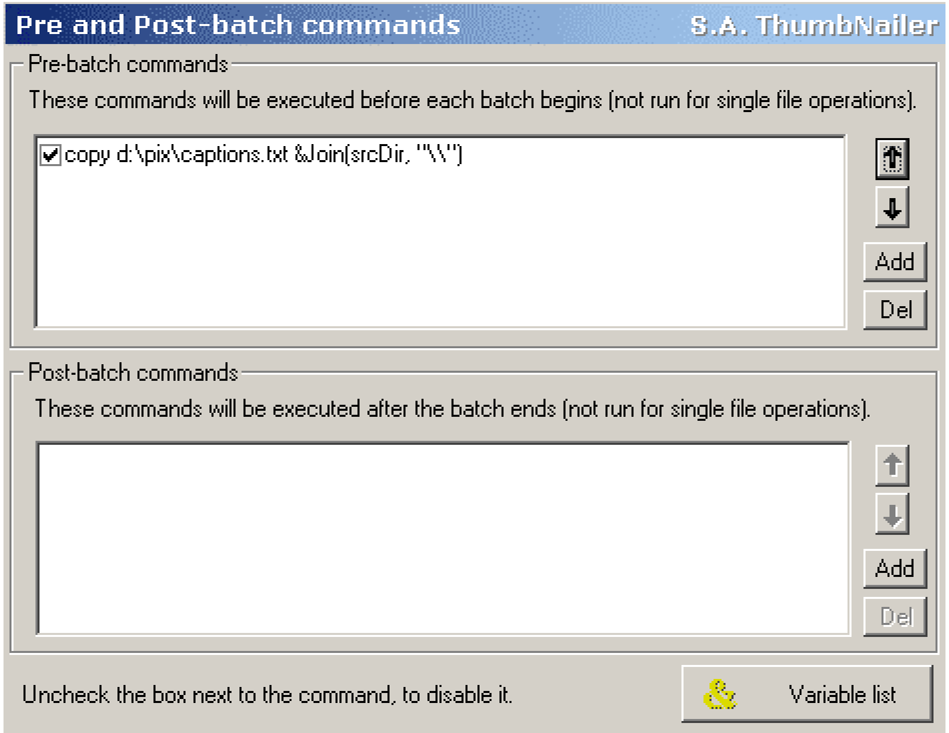
Progression

Controls how the image data is arranged in the file. Viewers could (in theory) display the image data as it is received, rather than waiting for the entire image to be downloaded.

Write as a bare JPEG-2000 data stream

It is best to leave this off.

Pre and Post-batch commands



Pre and Post-batch commands

S.A. ThumbNailer

Pre-batch commands

These commands will be executed before each batch begins (not run for single file operations).

☑ copy d:\pix\captions.txt &Join(srcDir, "\\")

Post-batch commands

These commands will be executed after the batch ends (not run for single file operations).

Uncheck the box next to the command, to disable it.

Pre-batch commands

ThumbNailer will execute these commands before your batch begins. This is done after the contents of the output folder have been moved to the Recycle Bin (if you chose that option in Advanced / Folder Settings / Clear Destination Folder), but before any other processing occurs. You can use this to call any external programs or batch files that you need to. Some samples uses might be to copy files from one folder to another, create folders, etc.. If a command fails, you will be given the opportunity to cancel the rest of the batch.

Post-batch commands

ThumbNailer will execute these commands after the batch ends.

Commands are executed in the order that they appear in the list. You can move commands up or down using the arrows on the left of the list.

Next to each command in the list is a check box. If the box is un-checked, the command will not be executed.

You can use variables and operators in these commands:

> “copy c:\foo.txt &Join(destDir, “\\foo2.txt”)”
>
> This will copy the file C:\foo.txt to the output folder as “foo2.txt”.

Each command you enter will be executed with a base directory of “C:\”.

Macros In Templates

See Macros
See TemplatesSee Operators and Variables

As an alternative to creating macro expressions in the Macro Expressions screen, ThumbNailer (as of v7.3) allows you to create them on your HTML templates. To do this, you need to create a special section called “macros” in your template. Ex:

```
#macro section
@:macros
@+
Macro: _MyMax(__value1, __value2) &If(Grtr(__value1, __value2), __value1, __value2)
Macro: _MyMin(__value1, __value2) &If(Grtr(__value1, __value2), __value2, __value1)
@-
```

This example defines two macros: _MyMax and _MyMin. These macros will output the minimum and maximum (respectively) of their input parameters.

You can define as many macros in the Macros section as you want. The only restriction is that they can only be used from within the template (ie. You can’t use them in other templates).

The syntax for defining a macro is:

Macro: *Name*(*parameters*) &*Expression*(…)

“Name” is the name of your macro. It should be different from any other macro or built-in ThumbNailer operator.

“parameters” are the input values for the macro. Parameters must be separated by commas.

“Expression” is the macro expression – what you want it to do with the input parameters. It must follow the rules of all other expressions.

Note: This feature is only available on ThumbNailer v7.3 and higher. Using templates with “macros” sections in older versions of ThumbNailer will likely cause an error.