

# **MySETIapp**

## **User's guide**

**V1.0.0.1**

**Aug 20,2023**

This file is part of MySETIapp.

MySETIapp is free software : you can redistribute it and /or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

MySETIapp is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with MySETIapp. If not, see <https://www.gnu.org/licenses/>.

MySETIapp is an application that was written to help in analyzing and interpreting the bit stream message from the 'A Sign in Space' project. This is a user driven effort and has no official affiliation with Daniela de Paulis, the SETI Institute, or the European Space Agency or any other collaborating agency or institute.

## Background

'A Sign in Space'  
website, <https://asignin.space/>

A Sign in Space is an interdisciplinary project by media artist Daniela de Paulis, in collaboration with the SETI Institute, the European Space Agency, the Green Bank Observatory and INAF, the Italian National Institute for Astrophysics. The project consists in transmitting a simulated extraterrestrial message as part of a live performance, using an ESA spacecraft as celestial source. The objective of the project is to involve the world - wide Search for Extraterrestrial Intelligence community, professionals from different fields and the broader public in the reception, decoding and interpretation of the message. This process will require global cooperation, bridging a conversation around the topics of SETI, space research and society, across multiple cultures and fields of expertise, <https://www.seti.org/event/sign-space>

The message was transmitted from the ESA's ExoMars Trace Gas Orbiter (TGO) on May 24 at 19:16 UTC/12:15 pm PDT.

It was received by three radio telescopes on earth May 24,2023. A group of individuals in the Discord 'A Sign in Space' channel unscrambled the message from the radio telemetry. The message published as Data17.bin is identified as the correctly transcribed bitstream of the message payload given to ESA.

The next step in the problem is the decoding of the payload bitstream into the next level of the message, the sign in space. After that the interpretation of the sign(s) in the embedded messages can commence.

There is a Discord Interpretation Chat. The group is large and has a wide array of talent. The tools that the group is using is quite varied which includes Excel, Photoshop, GIMP, Java, Python and c/c++.

Several people are also using online tools that use file uploads, typically text based. Importing data into Excel is also generally text based.

Photoshop and GIMP can use raw binary files typically 8 bit, 16 bit or 32 bit per element.

None of these tools use bit packed binary input. The first step has been to translate the bitstream message into a format that can be used by the various tools. The next steps involve examination of the bitstream data to solve how to decode it.

This program contains a set of tools that I have used in the examination of the message. It covers several of the basic files that people are using as a starting point along with several basic bitstream and image formatting , reordering, folding, mirroring, rotation, extraction functions. Hopefully this may make it easier for others people to explore the message.

## Limitations

This is a 64 bit Windows Desktop application. It primarily operates from the user's selection of the operations they wish to perform using the application's menus. The operations consist of file input(s), parameters, and file output.

Currently I removed the antiquated image viewer code from the program. The results from operations in the application can also be saved as BMP files that can be opened in Window Explorer for viewing. The next release of the software will likely include the updated image viewer.

## V1.0.0.1 Release

This is the first release. It includes the Visual Studio project files along with some of the 'message' data files. An application installer can also be used just for the executable application and data files.

The executable and data files can be stored in any writeable folder in Windows. Any installer and/or executable should be scanned for viruses when being downloaded along with a verification that the checksum for the installation matches the download.

The installer will add a desktop shortcut to run the application.

There is configuration file, MySETIapp.ini, that will be created when you run the software that keeps track of what files, parameters, window size and positions are used so that you do not have to start from scratch every time you run the application. This is stored in the same folder as the executable. The application does not need to access, record, or transmit information to the Internet.

The settings are only saved when you press the OK dialog button or when the main window and image window display are closed.

## Installation

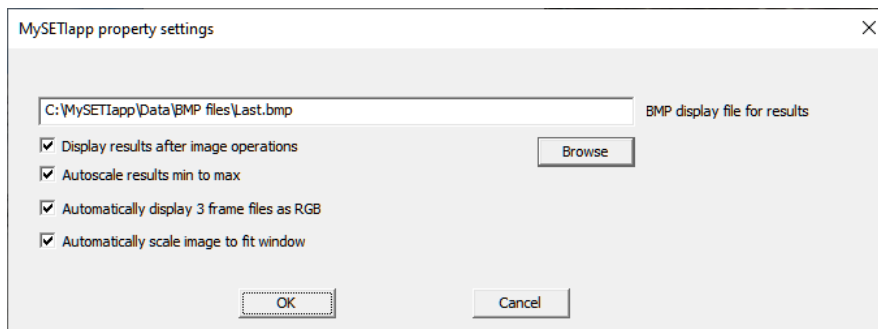
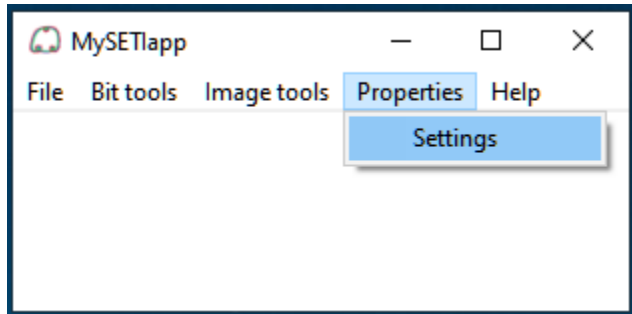
The installer is a very simple one. **The default install location is the windows volume drive \MySETIapp.** This is used because the application needs to be able to have read/write access of the folder it is installed. The 'Programs folder' requires administrative privileges to write to it. I use 'C:\MySETIapp' as the installation location. You can change the folder installation of the application using the browse button during the install.

The project source is also available on GitHub. The source project and solution uses Visual Studio 2019 or later. It has not been tested against earlier versions.

## First time running the application

The first time you open a dialog there are defaulted parameters that should be changed and set, in particular the filenames. Use the browse button to select the appropriate folder location for the files. Click OK in the dialog to save the dialogs parameters for the next time. Pressing the OK button saves the dialog settings. It rarely performs the dialog action. Most of the operations in the application have a specific dialog button to press to perform the operation, like Extract, Append, Reorder, ...

It is recommended that the application 'settings' be set when you first run the application so that it makes it easier to pull up the last BMP results from the last image operation.



**Please note that the user should enter file extensions when specifying filenames.** Default extensions are not added by the file dialogs or when data is saved. If you are saving text files, you should use .txt, .csv or similar extension used for text files. **For binary image files, .raw is recommended so that programs like Photoshop which expect a raw image file to have this extension. Bitmap image files should have an extension .bmp.** Though you are free to decide whatever extension you wish to use. The file extension will not change the file format that gets written.

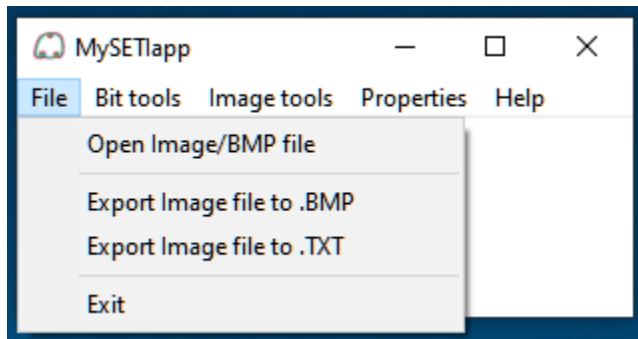
## Menu operations

Operations are centered around reading and generating files based on parameters entered in the dialog for that operation. The dialog typically needs 1 or 2 input filename, a set of operation parameters and an output filename.

The menus are split into 5 groups with 2 main groups that comprise the most of the functionality:

File      **Bit tools**      **Image Tools**      Properties      Help

File



### Open Image/BMP file

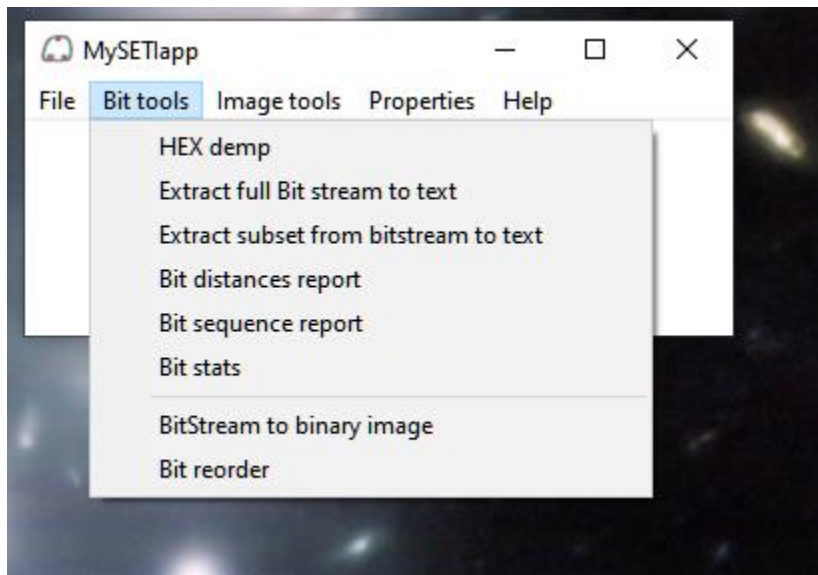
This will be implemented in the next release. Currently it just generates the BMP display file for results set in the application properties settings.

### Export Image file to .BMP file

Exporting to a BMP file can generate 2 types of BMP files. For image files that are single frame or multi-frame not divisible by 3 the BMP is saved as a greyscale image. For image files with multiples of 3 frames there is an option to use an RGB representation. The first frame is Red, 2<sup>nd</sup> frame is assigned Green and the third frame is assigned Blue. Only the first frame or 3 frames (RGB output) are converted into a BMP file. You can use auto scale to scale image from black to white. Auto scaling stretches the greyscale bitmap to represent a black to white image even for a single bit pixel.

'Export image file to text' to text allows an image file to be output in a text form that can be used as inputs for programs like Excel or various online applications.

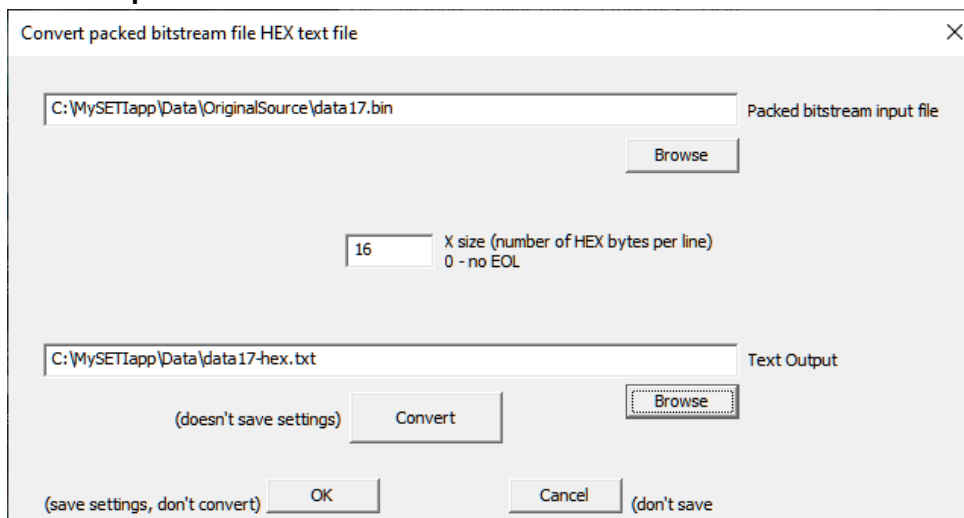
## Bit tools



The initial functionality in analyzing the bitstream are the bit tools that allow you to take the packed binary bit stream and examine it. The bit tools can also be used to dismantle and export packed bitstream file in various text or image representations.

It also has some tools that just help you look at what the bitstreams looks like in terms of spacing and sequences.

## HEX Dump



Hex dump generates a byte oriented hex dump of the input file. This could be either the packed bitstream or and image file. X size is the number of HEX coded bytes to put on a single line. If 0 then all the bytes in the file are output in a single line with no end of line (new line or '\n') at the end. Any file can be HEX dumped. There is white space between the ASCII HEX coded bytes.

## Extract full Bit Stream to text

The dialog box is titled "Convert packed bitstream file to text file". It contains the following fields and controls:

- Packed bitstream input file:** A text box containing "C:\MySETIapp\Data\OriginalSource\data17.bin" with a "Browse" button to its right.
- # bits in prologue:** A text box containing "80" with a "Browse" button to its right.
- #bits in header for a block:** A text box containing "0".
- #bits in block:** A text box containing "65536".
- X size:** A text box containing "256".
- Number of blocks:** A text box containing "1".
- Text Output:** A text box containing "C:\MySETIapp\Data\data17.txt" with a "Browse" button to its right.
- Buttons:** "Convert", "OK", and "Cancel".
- Labels:** "(doesn't save settings)" is placed above the "Convert" button. "(save settings, don't convert)" is placed below the "OK" button. "(don't save settings)" is placed below the "Cancel" button.

This converts the entire bitstream file into a text file. The text file is separated into 4 sections; prologue(set to 0 to exclude), header bits for a block in a file (set to 0 to exclude), block of bits output in the lines of X size bits, comma separated. The block header and blocks are repeated for the number of block(s) specified, everything after the last block is treated as the footer.

## Extract Bitstream to text file

The dialog box is titled "Extract Bitstream to text file". It contains the following fields and controls:

- Packed bitstream input file:** A text box containing "C:\MySETIapp\Data\OriginalSource\data17.bin" with a "Browse" button to its right.
- # bits to skip:** A text box containing "80" with a "Browse" button to its right.
- #bits to output:** A text box containing "65536".
- # of bits per line (0 all on one line):** A text box containing "256".
- Text Output:** A text box containing "C:\MySETIapp\Data\message.txt" with a "Browse" button to its right.
- Buttons:** "Extract", "OK", and "Cancel".
- Labels:** "(doesn't save settings)" is placed above the "Extract" button. "(save settings, don't convert)" is placed below the "OK" button. "(don't save settings)" is placed below the "Cancel" button.

This converts only part of the bitstream file into a text file. The values on a lines are comma separated. This is useful for importing into other programs as csv files. This useful for separating the header, message, and footer with different x sizes.

## Bit distances report

The dialog box titled "Report distance between 1 bits" contains the following elements:

- A text input field for the "Packed bitstream input file" with the path "C:\MySETIapp\Data\OriginalSource\data17.bin".
- A numeric input field for "# bits to skip" with the value "0", accompanied by a "Browse" button.
- A text input field for the "Text Output" with the path "C:\MySETIapp\Data\data17-bit-distances.txt", accompanied by a "Browse" button.
- A "Report" button.
- Three status labels: "(doesn't save settings)" to the left of the "Report" button, "(save settings, don't convert)" to the left of the "OK" button, and "(don't save)" to the right of the "Cancel" button.
- "OK" and "Cancel" buttons at the bottom.

This generates a csv style report of the bit position of each set bit in the bitstream and the distance to the from the last set bit. If the first bit in the file is set the distance is reported as 1.

## Bit Sequences report

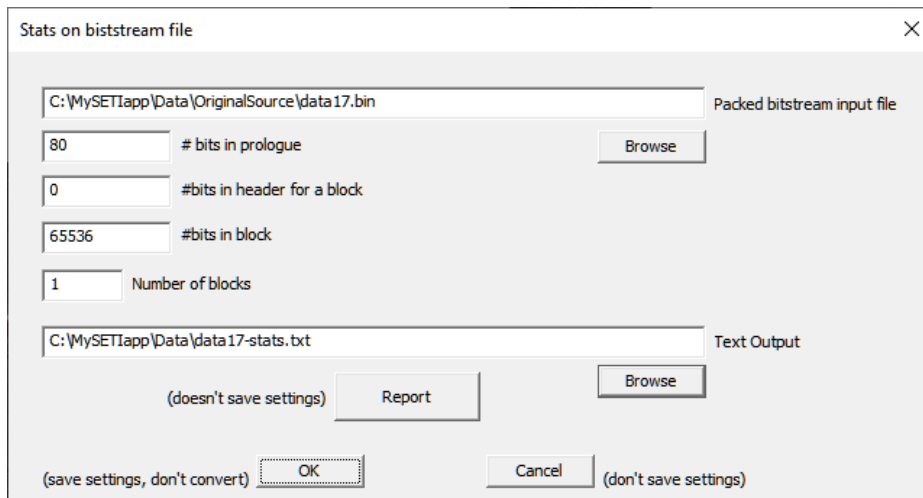
The dialog box titled "Report bit sequences" contains the following elements:

- A text input field for the "Packed bitstream input file" with the path "C:\MySETIapp\Data\OriginalSource\data17.bin".
- A numeric input field for "# bits to skip" with the value "0", accompanied by a "Browse" button.
- A text input field for the "Text Output" with the path "C:\MySETIapp\Data\data17-bit-sequences.txt", accompanied by a "Browse" button.
- A "Report" button.
- Three status labels: "(doesn't save settings)" to the left of the "Report" button, "(save settings, don't convert)" to the left of the "OK" button, and "(don't save)" to the right of the "Cancel" button.
- "OK" and "Cancel" buttons at the bottom.

This generates a csv style report that gives the starting position and length of sequences of set bits.



## Bit Stats



Stats on bistream file

C:\MySETIapp\Data\OriginalSource\data17.bin Packed bitstream input file

80 # bits in prologue Browse

0 #bits in header for a block

65536 #bits in block

1 Number of blocks

C:\MySETIapp\Data\data17-stats.txt Text Output

(doesn't save settings) Report Browse

(save settings, don't convert) OK Cancel (don't save settings)

This generates the following report text file:

Bitstream file stats

File report settings:

Header size:80

Number of Blocks:1

Header size per block:0

Block size:65536

Bit stats:

Number of bits set in prologue (header): 32, 40.0%

Number of bits set in body, block 0: 625, 1.0%

Number of bits found in footer: 80

Number of bits set in footer: 22, 27.5%

Total number of bits set: 679

## Bitstream to binary image

The dialog box is titled "Convert packed bitstream file to Image file". It contains the following fields and controls:

- Packed bitstream input file:** A text box with the path "C:\MySETIapp\Data\OriginalSource\data17.bin" and a "Browse" button.
- # bits in prologue (skipped):** A text box with the value "80".
- MSB to LSB:** An unchecked checkbox.
- #bits in header for a block (skipped):** A text box with the value "0".
- #bits in block:** A text box with the value "65536".
- X size:** A text box with the value "256".
- image bit depth:** A text box with the value "1".
- Number of blocks:** A text box with the value "1".
- ANY FOOTER BITS ARE SKIPPED:** An unchecked checkbox.
- Scale binary 0,1 to 0,255 in output image:** An unchecked checkbox.
- Image Output file:** A text box with the path "C:\MySETIapp\Data\Message.raw" and a "Browse" button.
- Buttons:** "Convert", "OK", "Cancel", and "Browse".
- Labels:** "(doesn't save settings)" and "(save settings, don't convert)".

This does the work of converting packed bitstream files into image files. This includes generating 1D image files which can use 1d linear bit reordering or 1D convolutions on the stream.

The conversion of the bitstream also allows the stream to be treated as having bit depths of 1 to 32 bits. It allows MSB->LSB or LSB->MSB decoding from the bitstream. The output can also be scaled so a binary image is displayed from black to white. The # bits in block should be divisible by (Xsize \* image bit depth). The Ysize of the output image = #bits in block / (X size \* image bit depth). The number of blocks becomes the number of frames in the file.

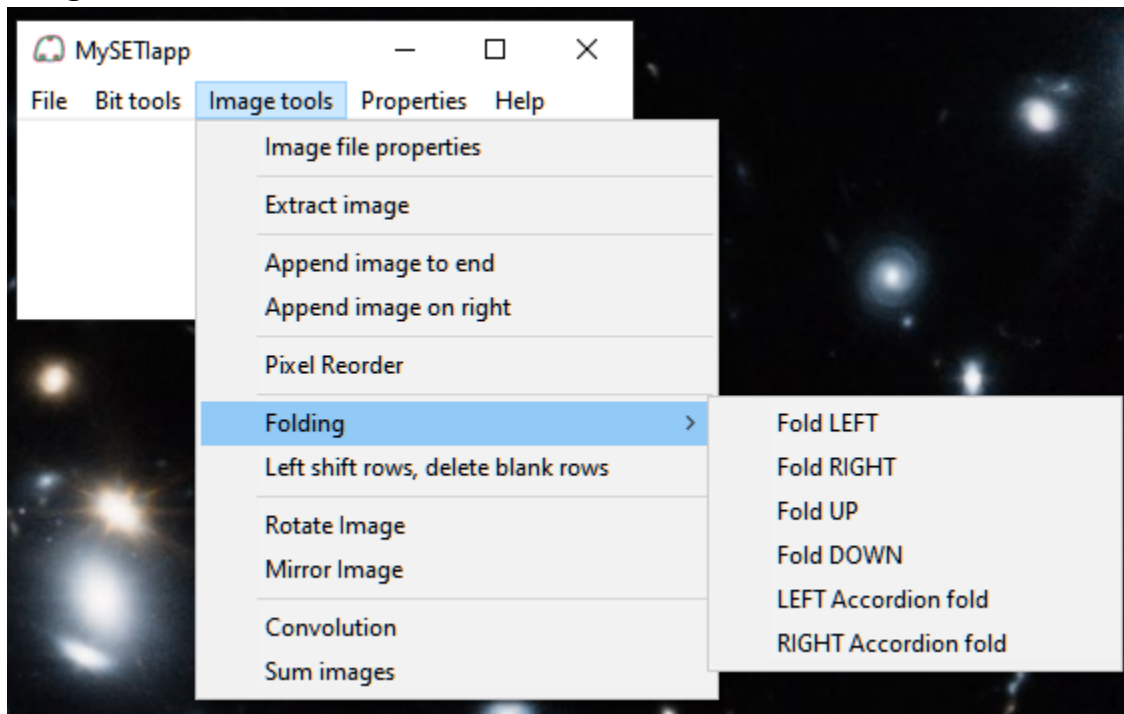
## Bit reorder

The dialog box is titled "Reorder bits, input file must be linear". It contains the following fields and controls:

- Bit Stream as linear image file:** A text box with the path "C:\MySETIapp\Data\LinearMessage.raw" and a "Browse" button.
- Input Image size:** A section with "1 # of frames", "65696 X size", and "1 Y size".
- Pixel reorder file:** A text box with the path "C:\MySETIapp\Data\Reorder\16x1 reverse order.txt" and a "Browse" button.
- Output Image file:** A text box with the path "C:\MySETIapp\Data\Reorder\ReorderedLinearMessage.raw" and a "Browse" button.
- Scale binary to 0,255:** An unchecked checkbox.
- Buttons:** "Reorder", "OK", "Cancel", and "Browse".
- Labels:** "(doesn't save settings)" and "(save settings, don't convert)".

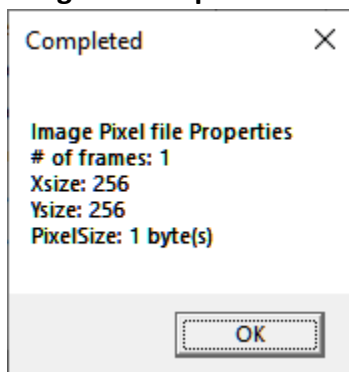
See the Image reordering documentation for the formats that can be used for the reorder files. When using bit reorder from the bit tools menu the reordering must be 1D (Ysize=1). The X size must be divisible by the Xsize of the reordering kernel. The linear image is divided into groups the X size of the kernel. The bits within the kernel are reordered according to the kernel specification.

## Image Tools



The purpose image tools is to manipulate 2D images. This includes mirroring, rotation, folding, extraction, convolution, summation, and appending images.

## Image File Properties



This reports the header information for the selected image file. It does not generate a report file.

## Extract Image

The 'Extract subimage from image' dialog box contains the following fields and controls:

- Input Image file:** A text box containing 'C:\MySETIapp\Data\Message.raw'.
- Input Image size:** A label with '256 X size' and '256 Y size'.
- # of frames:** A text box containing '1'.
- Start frame (0 based):** A text box containing '0' with a 'Browse' button next to it.
- End frame (0 based):** A text box containing '0'.
- X loc:** A text box containing '65'.
- Y loc:** A text box containing '50'.
- Centering:** A checked checkbox labeled 'x loc, y loc is center of subimage'.
- sub image size:** Text boxes for 'X size' (128) and 'Y size' (128).
- output image size:** Text boxes for 'X size' (128) and 'Y size' (128).
- Scale binary to 0,255:** An unchecked checkbox.
- Output Image file:** A text box containing 'C:\MySETIapp\Data\SignA.raw' with a 'Browse' button.
- Buttons:** '(doesn't save settings)', 'Extract', 'Browse', '(save settings, don't convert)', 'OK', 'Cancel', and '(don't save settings)'.

This allows a sub-image to be extracted from a source image. The extraction can be for a range of frames in the source image. The X,Y size of the sub-image must be  $\leq$  the X,Y size of the output image. The output image can be larger than the input image. The sub-image position is either the upper left corner of the sub-image in the source image or the center of the sub-image in the source image. The output image will be zero padded as needed if the sub-image size extends beyond the bounds of the source image. The use of the centering option can be used to center a specific pixel from the input image to be in the center of the output image. This can make alignments to other sub images easier along with ensuring the other sub images are also the same size.

## Append image to end

The 'Append image to end' dialog box contains the following fields and controls:

- Input Image file:** A text box containing 'C:\MySETIapp\Data\SignA.raw' with a 'Browse' button.
- Input Image size:** A label with '128 X size' and '128 Y size'.
- # of frames:** A text box containing '1'.
- Image file to append:** A text box containing 'C:\MySETIapp\Data\SignB.raw' with a 'Browse' button.
- Append as additional frame(s) in file:** A checked checkbox.
- Output Image file:** A text box containing 'C:\MySETIapp\Data\SignAB.raw' with a 'Browse' button.
- Buttons:** '(doesn't save settings)', 'Append', 'Browse', '(save settings, don't convert)', 'OK', 'Cancel', and '(don't save settings)'.

Images are appended at the end of another image or added as additional frame(s) in the image file.

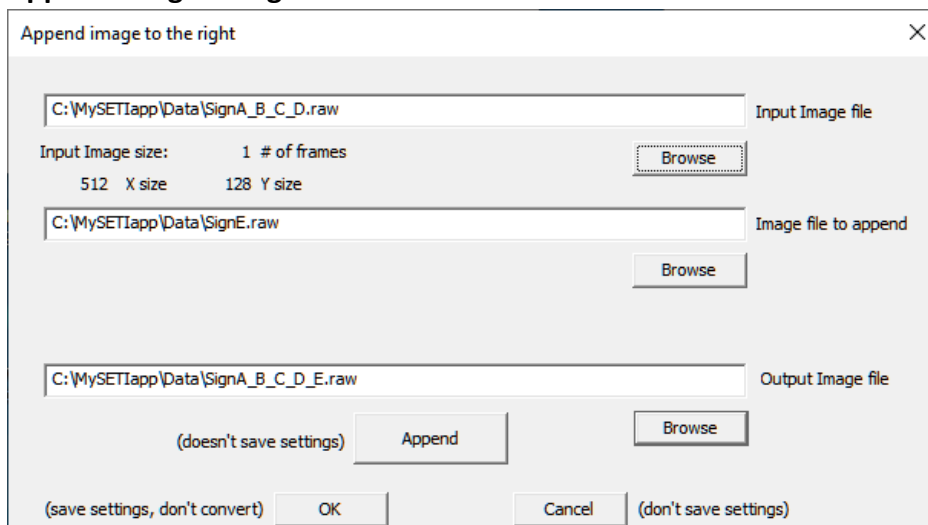
### Appended to frame condition

If the 2<sup>nd</sup> image is being appended on the first frame without being added as an additional frame then the output image will be the Y size of the first image plus the Ysize of the second image. Appending the X size of both input image files be the same along with the same number of frames in each input file.

### Appended as additional frames condition

The first and 2<sup>nd</sup> input image files must have the same X, Y size. The number of frames in each input image file can be different. The number of frames in the output image will be the number of frames from the first input image plus the number of frames from the 2<sup>nd</sup> input image. This is particularly useful when generating a 3 frame file that you want to convert to an RGB representation. When exporting a 3 frame file the 3 frames can be interpreted as frame 1 is Red, frame 2 is Green and frame 3 is Blue.

### Append image on right

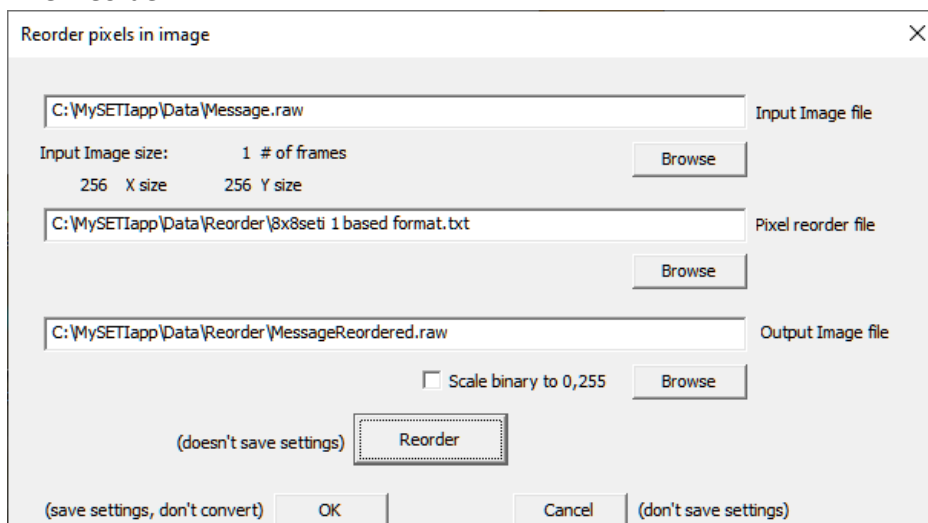


The 'Append image to the right' dialog box contains the following fields and controls:

- Input Image file:** Text field with 'C:\MySETIapp\Data\SignA\_B\_C\_D.raw' and a 'Browse' button.
- Input Image size:** Displays '1 # of frames', '512 X size', and '128 Y size'.
- Image file to append:** Text field with 'C:\MySETIapp\Data\SignE.raw' and a 'Browse' button.
- Output Image file:** Text field with 'C:\MySETIapp\Data\SignA\_B\_C\_D\_E.raw' and a 'Browse' button.
- Buttons:** 'Append', 'OK', 'Cancel', and a 'Browse' button for the output file.
- Labels:** '(doesn't save settings)' is placed above the 'Append' button, and '(save settings, don't convert)' is placed above the 'OK' button.

This appends the second image to the right side of the first image. This can be used to string a set of sub-images together into one image with a common x and Y alignment. Such as the possible 5 signs in the “A Sign In Space” message.

### Pixel Reorder



The 'Reorder pixels in image' dialog box contains the following fields and controls:

- Input Image file:** Text field with 'C:\MySETIapp\Data\Message.raw' and a 'Browse' button.
- Input Image size:** Displays '1 # of frames', '256 X size', and '256 Y size'.
- Pixel reorder file:** Text field with 'C:\MySETIapp\Data\Reorder\8x8set1 based format.txt' and a 'Browse' button.
- Output Image file:** Text field with 'C:\MySETIapp\Data\Reorder\MessageReordered.raw' and a 'Browse' button.
- Checkbox:** 'Scale binary to 0,255' is unchecked.
- Buttons:** 'Reorder', 'OK', 'Cancel', and a 'Browse' button for the output file.
- Labels:** '(doesn't save settings)' is placed above the 'Reorder' button, and '(save settings, don't convert)' is placed above the 'OK' button.

1D/2D pixel reordering divides an image up into blocks which are the size of the reordering kernel. The pixels inside each block are reordered according to the kernel specification. The limit on the size of the kernel is the X,Y size of the image. This of course would allow you to arbitrarily remap any pixel to any other place in the image. Someone could make the resulting image into almost any representation desired. It is more likely that a much smaller reordering kernel would be used.

File format for reordering files

The reordering file for 1D/2D reordering has the following 3 text formats.

The following example kernels reverse the order of pixels left to right in the block and swap the rows so that an 8x2 block of pixel values:

```
0 1 2 3 4 5 6 7
8 9 10 11 12 13 14 15
becomes
15 14 13 12 11 10 9 8
7 6 5 4 3 2 1 0
```

1.) Relative pixel mapping format

n,m

followed by n\*m pairs of values with whitespace between the pairs after the n\*m pairs of values an optional description is recommended. The reordering value is relative to its position in the kernel. A 0,0 means the pixel is not moved

Example file contents format – relative pixel maps

```
8,2
7,1 5,1 3,1 1,1 -1,1 -3,1 -5,1 -7,1
7,-1 5,-1 3,-1 1,-1 -1,-1 -3,-1 -5,-1 -7,-1
8X2 kernel. This swaps the two rows and reverse the order of a row
```

2.) 0 based linear kernel address format

n,m,0

followed by n\*m values with whitespace between the values after the n\*m values an optional description is recommended. Reordering values is the linear address of replacement pixel in the kernel, 0 based.

Example file contents format –0 based linear kernel address

```
8,2,0
15 14 13 12 11 10 9 8
7 6 5 4 3 2 1 0
8X2 kernel. This swaps the two rows and reverse the order of a row
```

3.) 1 based linear kernel address format

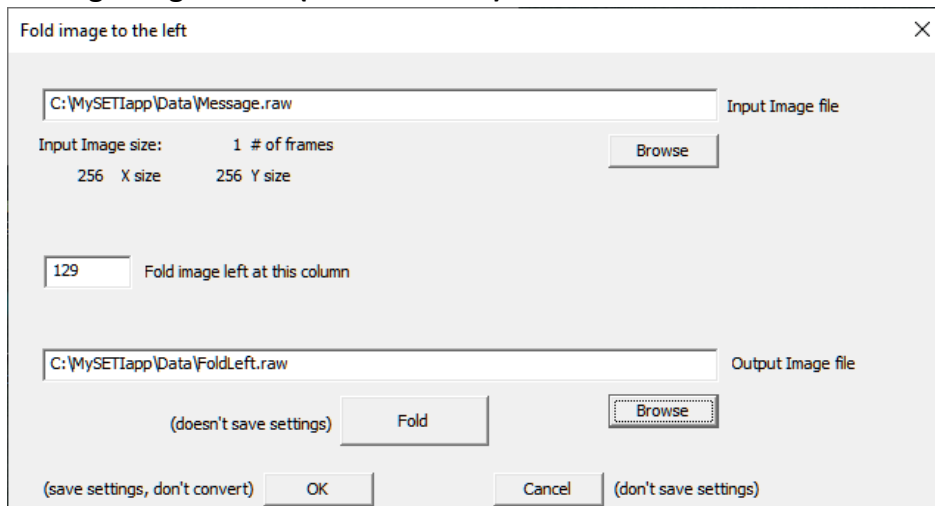
n,m,1

followed by n\*m values with whitespace between the values after the n\*m values an optional description is recommended. Reordering values is the linear address of replacement pixel in the kernel, 1 based.

Example file contents format –1 based linear kernel address

```
8,2,1
16 15 14 13 12 11 10 9
8 7 6 5 4 3 2 1
```

## Folding along column (LEFT or RIGHT)

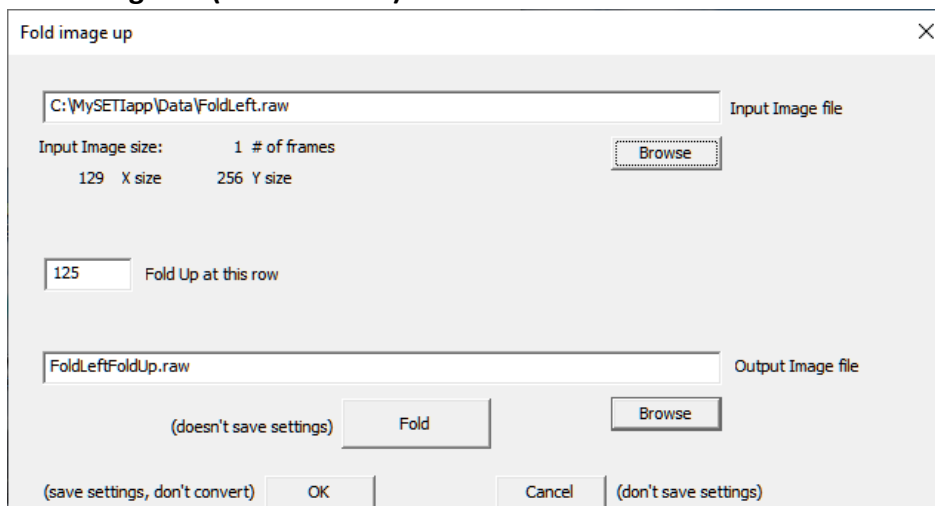


The dialog box is titled "Fold image to the left" with a close button (X) in the top right corner. It contains the following elements:

- An "Input Image file" text box containing the path "C:\MySETIapp\Data\Message.raw".
- An "Input Image size:" section showing "1 # of frames", "256 X size", and "256 Y size".
- A "Browse" button next to the input file path.
- A text box for the fold column, containing the value "129", with the label "Fold image left at this column".
- An "Output Image file" text box containing the path "C:\MySETIapp\Data\FoldLeft.raw".
- A "Browse" button next to the output file path.
- A "Fold" button.
- At the bottom, there are three buttons: "OK" (with the label "(save settings, don't convert)" to its left), "Cancel" (with the label "(don't save settings)" to its right), and a "Fold" button (with the label "(doesn't save settings)" to its left).

The input image is folded at the specified column either LEFT or RIGHT. The LEFT fold is shown above. If the image is not folded exactly in the center the resulting image is enlarged to accommodate the fold location. The resulting new image size is also reported. If there are multiple frames in the image file then each frame is identically folded. The pixels in the fold image are added together. If 2 pixels overlap as a result of the fold the new pixel value is the sum of the 2 overlapped pixels.

## Fold along row (UP or DOWN)



The dialog box is titled "Fold image up" with a close button (X) in the top right corner. It contains the following elements:

- An "Input Image file" text box containing the path "C:\MySETIapp\Data\FoldLeft.raw".
- An "Input Image size:" section showing "1 # of frames", "129 X size", and "256 Y size".
- A "Browse" button next to the input file path.
- A text box for the fold row, containing the value "125", with the label "Fold Up at this row".
- An "Output Image file" text box containing the path "FoldLeftFoldUp.raw".
- A "Browse" button next to the output file path.
- A "Fold" button.
- At the bottom, there are three buttons: "OK" (with the label "(save settings, don't convert)" to its left), "Cancel" (with the label "(don't save settings)" to its right), and a "Fold" button (with the label "(doesn't save settings)" to its left).

The input image is folded at the specified row either UP or DOWN. The UP fold is shown above. If the image is not folded exactly in the center the resulting image is enlarged to accommodate the fold location. The resulting new image size is also reported. If there are multiple frames in the image file then each frame is identically folded. The pixels in the fold image are added together. If 2 pixels overlap as a result of the fold the new pixel value is the sum of the 2 overlapped pixels.

## Accordion fold (LEFT or RIGHT)

The dialog box is titled "Accordion fold image to the left" and has a close button (X) in the top right corner. It contains the following elements:

- An input field for the "Input Image file" with the path "C:\MySETIapp\Data\Message.raw" and a "Browse" button to its right.
- Below the input field, the text "Input Image size: 1 # of frames" is displayed, followed by "256 X size" and "256 Y size". A "Browse" button is also present.
- A numeric input field containing "16" with the label "size of accordion (fold to the left)" to its right.
- An output field for the "Output Image file" with the path "C:\MySETIapp\Data\AccordionLeft.raw" and a "Browse" button to its right.
- At the bottom, there are three buttons: "Fold" (with "(doesn't save settings)" to its left), "OK" (with "(save settings, don't convert)" to its left), and "Cancel" (with "(don't save settings)" to its right).

This function does an accordion fold an image along a vertical axis. It folds the left or right side of the accordion fold to the opposite side of the fold. The folded file is 1/2 the width of the unfolded image. The width of the input image must be divisible by the accordion size. The input image file width must be even. The accordion size must also be even. Think paper being cut into strips that are the width of the accordion size. Then the strip is folded and the folded strips stuck back together.

## Left shift rows, delete blank rows

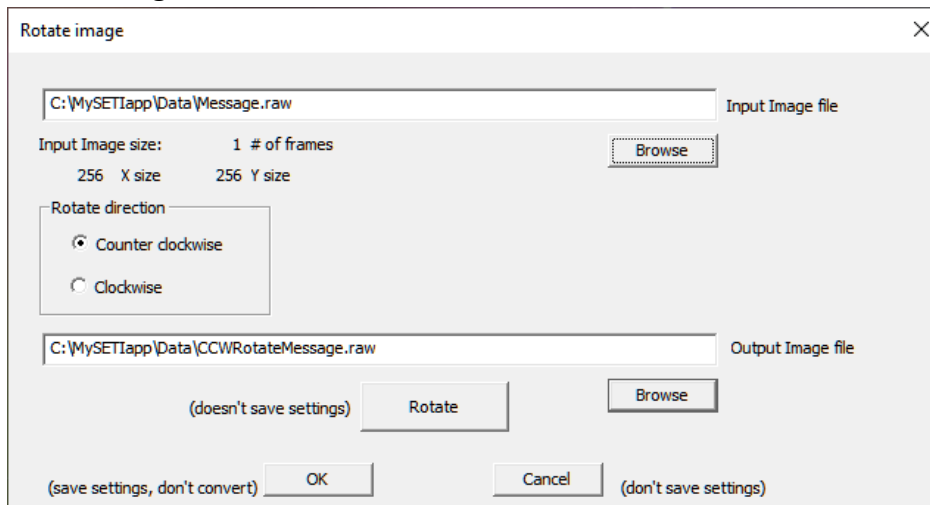
The dialog box is titled "Left shift rows, delete blank rows" and has a close button (X) in the top right corner. It contains the following elements:

- An input field for the "Input Image file" with the path "C:\MySETIapp\Data\Message.raw" and a "Browse" button to its right.
- Below the input field, the text "Input Image size: 1 # of frames" is displayed, followed by "256 X size" and "256 Y size". A "Browse" button is also present.
- An output field for the "Output Image file" with the path "C:\MySETIapp\Data\LeftShiftMessage.raw" and a "Browse" button to its right.
- At the bottom, there are three buttons: "Shift" (with "(doesn't save settings)" to its left), "OK" (with "(save settings, don't convert)" to its left), and "Cancel" (with "(don't save settings)" to its right).

This function shifts a row to the left so that leading 0s are eliminated. Blank rows are also deleted. This operation does not appear to be applied to the 'A Sign In Space' message. It is useful for asynchronous serial streams with varying packet lengths.



## Rotate Image

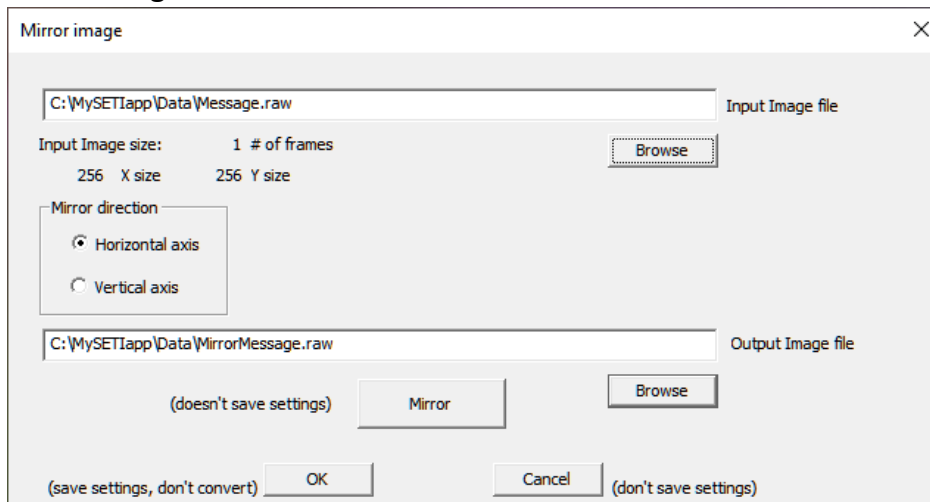


The 'Rotate image' dialog box contains the following elements:

- Input Image file:** A text field with the path 'C:\MySETIapp\Data\Message.raw' and a 'Browse' button to its right.
- Input Image size:** A label '1 # of frames' followed by '256 X size' and '256 Y size'.
- Rotate direction:** A group box containing two radio buttons: 'Counter clockwise' (selected) and 'Clockwise'.
- Output Image file:** A text field with the path 'C:\MySETIapp\Data\CCWRotateMessage.raw' and a 'Browse' button to its right.
- Buttons:** A 'Rotate' button, an 'OK' button, and a 'Cancel' button. The 'OK' button is labeled '(save settings, don't convert)' and the 'Cancel' button is labeled '(don't save settings)'.

This rotates an image counter clockwise or clockwise. If the image is not square the new image X and Y size will change to reflect the rotation.

## Mirror image



The 'Mirror image' dialog box contains the following elements:

- Input Image file:** A text field with the path 'C:\MySETIapp\Data\Message.raw' and a 'Browse' button to its right.
- Input Image size:** A label '1 # of frames' followed by '256 X size' and '256 Y size'.
- Mirror direction:** A group box containing two radio buttons: 'Horizontal axis' (selected) and 'Vertical axis'.
- Output Image file:** A text field with the path 'C:\MySETIapp\Data\MirrorMessage.raw' and a 'Browse' button to its right.
- Buttons:** A 'Mirror' button, an 'OK' button, and a 'Cancel' button. The 'OK' button is labeled '(save settings, don't convert)' and the 'Cancel' button is labeled '(don't save settings)'.

This mirrors an image counter on the specified axis. The If the image is not square the new image X and Y size will change to reflect the rotation.

## Convolution

Image convolution

C:\MySETIapp\Data\Message.raw Input Image file

Input Image size: 1 # of frames  
256 X size 256 Y size Browse

C:\MySETIapp\Data\Convolution\3x3smooth.txt Convolution kernel file  
Browse

C:\MySETIapp\Data\Convolution\ConvolvedMessage.raw Output Image file  
Browse

(doesn't save settings) Convolve

(save settings, don't convert) OK Cancel (don't save settings)

This applies a convolution kernel to the specified image. It does not scale the results afterwards. This function does a kernel convolution on the input image. It does not scale the results afterwards. The convolution kernel is read in from a text file. The kernel weights are floating point numbers and can be less than 0. Kernel sizes do not have to be square and can even be linear (such as a convolution kernel for a 1D image file). If you are not familiar with convolution there is material online and in image processing textbooks that explain it. The border of an image that has been convolved may be missing data due to the convolution. The size of the kernel will dictate how many rows and columns on the border of the image are affected. These will be 0 filled in the output file.

### Kernel file format

n,m                      Kernel size, n wide by m long  
w1 w2 w3 ....        List n\*m long of weights white space delimited  
optional description

### Example

```
3,3
0.071428571 0.142857143 0.071428571
0.142857143 0.142857143 0.142857143
0.071428571 0.142857143 0.071428571
3x3 kernel, smoothing using weighted average
```

## Sum images

Add images

C:\MySETIapp\Data\SignA.raw Input Image file

Input Image size: 1 # of frames  
128 X size 128 Y size

Browse

C:\MySETIapp\Data\SignE.raw Image to add

Input Image size: 1 # of frames  
128 X size 128 Y size

Browse

C:\MySETIapp\Data\SummedAE.raw Output Image file

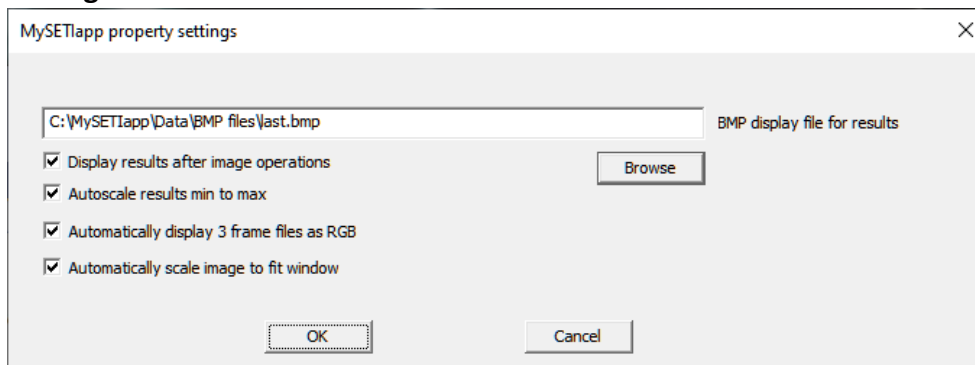
(doesn't save settings) Add (Browse)

(save settings, don't convert) OK Cancel (don't save settings)

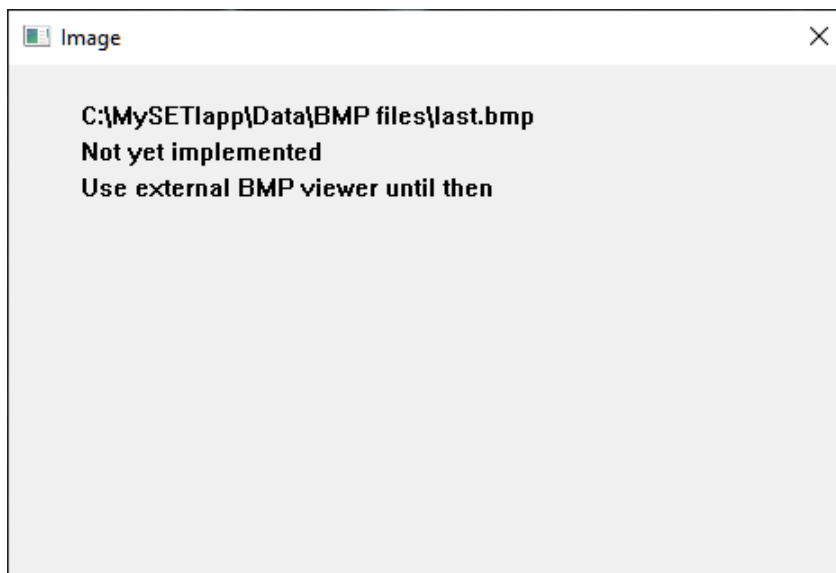
This sums 2 image files together. Both input files must have the same X,Y size and number of frames. If there is more than 1 frame in a file then the summation is frame by frame from each file. So the output image frame 1 would be the sum of Input image frame 1 plus the Image to add frame 1, ...

## Properties and Help menu

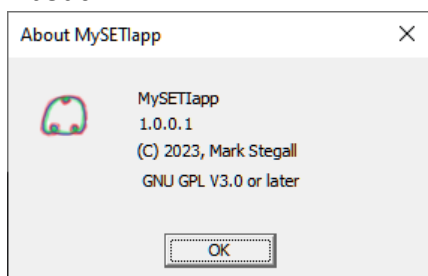
### Settings



The Properties menu allows you to set program settings. Operations that generated an output image file will also generate a BMP file in the folder using the name specified in this settings dialog. The Display results after image operations must be checked for this to happen. This will cause a bitmap image to be displayed in a separate window. **The current image window will display the following until the new image display implementation is completed.** This is expected to be available in the next release.



### About



This displays the application's current About dialog.

## Version information

V1.0.0.1      Aug 20,2023      Initial release