Think Silicon

AN APPLIED MATERIALS COMPANY

NEMA®| pix-presso

Starting Guide

Version 1.13

Part Number: D-PIX-DOC

November 10, 2023

Disclaimer

Questions or comments may be directed to:
Think Silicon S.A
Suite B8
Patras Science Park
Rion Achaias 26504, Greece
web: http://www.think-silicon.com
email:info@think-silicon.com
Tel: +30 2610 911543
Fax: +30 2610 911544

Contents

1 Preface

NEMA®| pix-presso is an image conversion software developed by Think Silicon S.A., An Applied Materials Company. Its main purpose is to provide an easy and efficient way of converting images into/from formats suitable for low power embedded platforms.

2 System Requirements

OS: Windows 7 32/64-bit (or newer), Linux (Ubuntu 32/64-bit), MacOS (X), Screen resolution: 800x600 or higher, RAM: at least 256 MB for up to 1080p images (512 MB and 768 MB for 4K and 8K images respectively), Hard Drive: 50 MB available space.

3 Features

The current version of NEMA®| pix-presso supports conversion to/from several image formats. More specifically it supports conversion to PNG, JPG, RGBA8888, ABGR8888, BGRA8888, ARGB8888, RGB24, BGR24, RGBA5650, BGRA5650, RGBA5551, ARGB1555, ABGR1555, BGRA5551, RGBA4444, ARGB4444, ABGR4444, BGRA4444, RGBA332, RGBA2222, BGRA2222, ABGR2222, ARGB2222, AL88, AL44, A8, A4, A2, A1, L8, L4, L2, L1, LUT2, LUT4, LUT16, LUT256 along with Think Silicon's proprietary image formats; TSC™4, TSC™6, TSC™6a and TSVG. Moreover, it supports conversion from the same image formats as well as SVG format.

LUT formats (LUT2, LUT4, LUT16, LUT256) are output only formats and produce two image files: the pallet (`filename_palette.COUNT.rgba`) and indices (`filename_indices.COUNT.gray`). The pallet defines a set of colors and the indices indicate for each pixel an index value in the palette. COUNT is the color count in each pallet (2, 4, 16 or 256).

The image formats have been selected according to the purpose of this tool, which is to offer the user the flexibility to convert images into formats suitable for low power embedded devices. Such devices support uncompressed formats (32-bit RGBA family) and compressed formats with direct memory mapping (16-bit RGBA, A, L and TSC family formats). Along with these formats, PNG is supported as an export format so that the user can convert any kind of images into a widely acceptable display format (and for instance, make displayable, in almost any image viewer software, compressed images). JPG and SVG are supported as import format since they are widely used formats in mainstream computing.

The 32-bit RGBA family, encode each channel of a pixel in 8-bits, and the only difference among these formats is the way that the channels are stored in memory. RGBA5650 uses 5-bits for the red and blue channel and 6-bits for the green channel, while the alpha channel is omitted; the pixel encoding in RGBA5551, RGBA4444 and RGBA332 formats is derived respectively. When converting from bigger formats to smaller ones, eg. RGBA8888 to RGBA332, dithering can be applied as well. A8 and A1 formats keep only the information of the alpha channel (transparency channel encoded in 8 or 1-bits respectively). L8, L4, L2, L1 contain only the luminance of each pixel (encoded in 8, 4, 2 and 1-bit respectively). Figure 1 illustrates the pixel encoding according to these formats.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RGBA8888 (32-bit with transparency)	A7	A6	A5	A4	A3	A2	A1	A0	B7	B6	B5	B4	B3	B2	B1	B0	G7	G6	G5	G4	G3	G2	G1	G0	R7	R6	R5	R4	R3	R2	R1	R0
ABGR8888 (32-bit with transparency)	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0	A7	A6	A5	A4	A3	A2	A1	A0
BGRA8888 (32-bit with transparency)	A7	A6	A5	A4	A3	A2	A1	A0	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
ARGB8888 (32-bit with transparency)	B7	B6	B5	B4	B3	B2	B1	B0	G7	G6	G5	G4	G3	G2	G1	G0	R7	R6	R5	R4	R3	R2	R1	R0	A7	A6	A5	A4	A3	A2	A1	A0
RGB24 (24-bit without transparency)									B7	B6	B5	B4	B3	B2	B1	B0	G7	G6	G5	G4	G3	G2	G1	G0	R7	R6	R5	R4	R3	R2	R1	R0
BGR24 (24-bit without transparency)									R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
RGBA5650 (16-bit without transparency)																	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
BGRA5650 (16-bit without transparency)																	B4	B3	B2	B1	B0	G5	G4	G3	G2	G1	G0	R4	R3	R2	R1	R0
RGBA5551 (16-bit with transparency)																	R4	R3	R2	R1	R0	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0	A0
ARBG1555 (16-bit with transparency)																	A0	R4	R3	R2	R1	R0	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
ABGR1555 (16-bit with transparency)																	A0	B4	B3	B2	B1	B0	G4	G3	G2	G1	G0	R4	R3	R2	R1	R0
BGRA5551 (16-bit with transparency)																	B4	B3	B2	B1	B0	G4	G3	G2	G1	G0	R4	R3	R2	R1	R0	A0
RGBA4444 (16-bit with transparency)																	R3	R2	R1	R0	G3	G2	G1	G0	B3	B2	B1	B0	A3	A2	A1	A0
ARGB4444 (16-bit with transparency)																	A3	A2	A1	A0	R3	R2	R1	R0	G3	G2	G1	G0	B3	B2	B1	B0
ABGR4444 (16-bit with transparency)																	A3	A2	A1	A0	B3	B2	B1	B0	G3	G2	G1	G0	R3	R2	R1	R0
BGRA4444 (16-bit with transparency)																	B3	B2	B1	B0	G3	G2	G1	G0	R3	R2	R1	R0	A3	A2	A1	A0
RGBA332 (8-bit without transparency)																									R2	R1	R0	G2	G1	G0	B1	B0
RGBA2222 (8-bit with transparency)																									R1	R0	G1	G0	B1	B0	A1	A0
BGRA2222 (8-bit with transparency)																									B1	B0	G1	G0	R1	R0	A1	A0
ABGR2222 (8-bit with transparency)																									A1	A0	B1	B0	G1	G0	R1	R0
ARGB2222 (8-bit with transparency)																									A1	A0	R1	R0	G1	G0	B1	B0
AL88 (16-bit grayscale with transparency)																	A7	A6	A5	A4	A3	A2	A1	A0	L7	L6	L5	L4	L3	L2	L1	L0
AL44 (8-bit grayscale with transparency)																									A3	A2	A1	A0	L3	L2	L1	L0
L8 (8-bit grayscale)																									L7	L6	L5	L4	L3	L2	L1	L0
L4 (4-bit grayscale)																													L3	L2	L1	L0
L2 (2-bit grayscale)																															L1	L0
L1 (1-bit grayscale)																																L0
A8 (8-bit transparency only)																									A7	A6	A5	A4	A3	A2	A1	A0
A4 (4-bit transparency only)																													A3	A2	A1	A0
A2 (2-bit transparency only)																															A1	A0
A1 (1-bit transparency only)																																A0

Figure 1: 32-bit RGBA family, 16-bit RGBA, A and L family formats pixel encoding

The TSC family consists of a series of lossy block-based compression algorithms that encode each 4x4-pixels block in 64-bits (TSC™4) or 96-bits (TSC™6 and TSC™6a); for a 32-bit RGBA image, such block occupies 4x4x32 = 512 bits. TSC™4 and TSC™6 formats are suitable for compressing images with no alpha information. The compression of images with alpha information can be handled by the TSC™6a format. Due to the encoding scheme (blocks of 4x4 pixels), the current version of the TSC algorithm can only handle images with dimensions multiples of 4. Each of the TSC family formats is implemented in two different versions of the compression algorithm (Fast and High Quality). The final compressed image occupies the same size in both cases. Nevertheless, the user is free to choose the compression algorithm according to their demands.

The TSVG format is a binary format, suitable for containing SVG information in binary form. The benefits of converting an SVG image to the TSVG binary format is lower memory usage and higher performance compared with parsing an SVG file at runtime. The SVG elements and attributes that are supported for conversion in the TSVG format are detailed in the document *NEMA®| GFX Extensions, TSVG Supported Elements List*

In addition, NEMA®| pix-presso is able to convert (as well as import) images using the Morton order (Z-order) encoding. Enabling this option will not affect the selected format.This encoding technique has the advantage that the pixels within an image are reordered in way that they preserve their locality while being treated as one dimension data. In order to convert an image to a selected format using the Morton order encoding, the respective checkbox needs to be checked.

4 Functionality

4.1 GUI Mode

NEMA®| pix-presso adopts the layout illustrated in Figure 2. This consists of a menu bar, three image view areas (one for the source, target and difference image respectively), a zoom slider, a combo-box that allows the user to select the target image format, a checkbox in order to select whether the converted image will be encoded using the Morton order encoding as well as a status bar for the display of various information generated during runtime (image resolution, zoom scale, output image size and RMSE).

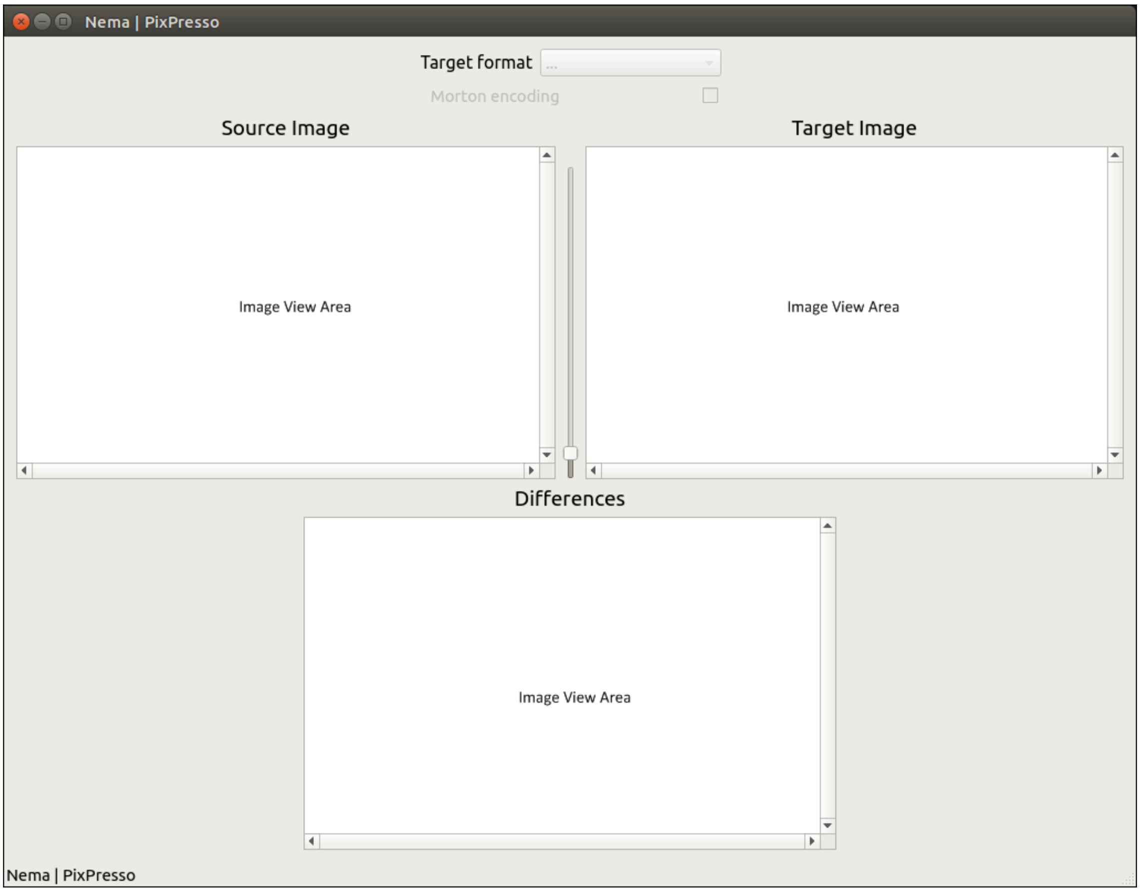

Figure 2: NEMA®| pix-presso Layout

An image can be opened by pressing **Open** in the **File** menu. The next step is to perform a conversion into a desired format. This is performed by selecting the format in the **combo-box**. During the conversion, a progress-bar containing the progress of the procedure is displayed. After the successful conversion of an image from one format to another, a comparison between the original and the final image has to be made, so that the user can evaluate whether the specific conversion meets specific requirements. For this purpose, the RMSE (Root Mean Square Error) metric between the original and the converted image is displayed in the status bar (Figure 3), which calculates a normalized average error value for the whole image distribution. The RMSE values can be within the interval [0, 255], with 0 being the value of a perfect conversion (no information loss) and 255 the worst case value (all information is lost). In case whereas the converted image needs to be encoded according to the Morton order encoding, the respective checkbox needs to be checked. Please note that the formats that support this encoding method are formats of the RGBA-family (32, 16 and 8 bit formats) A-family (A1, A8) and L-family (L8, L4, L2, L1)

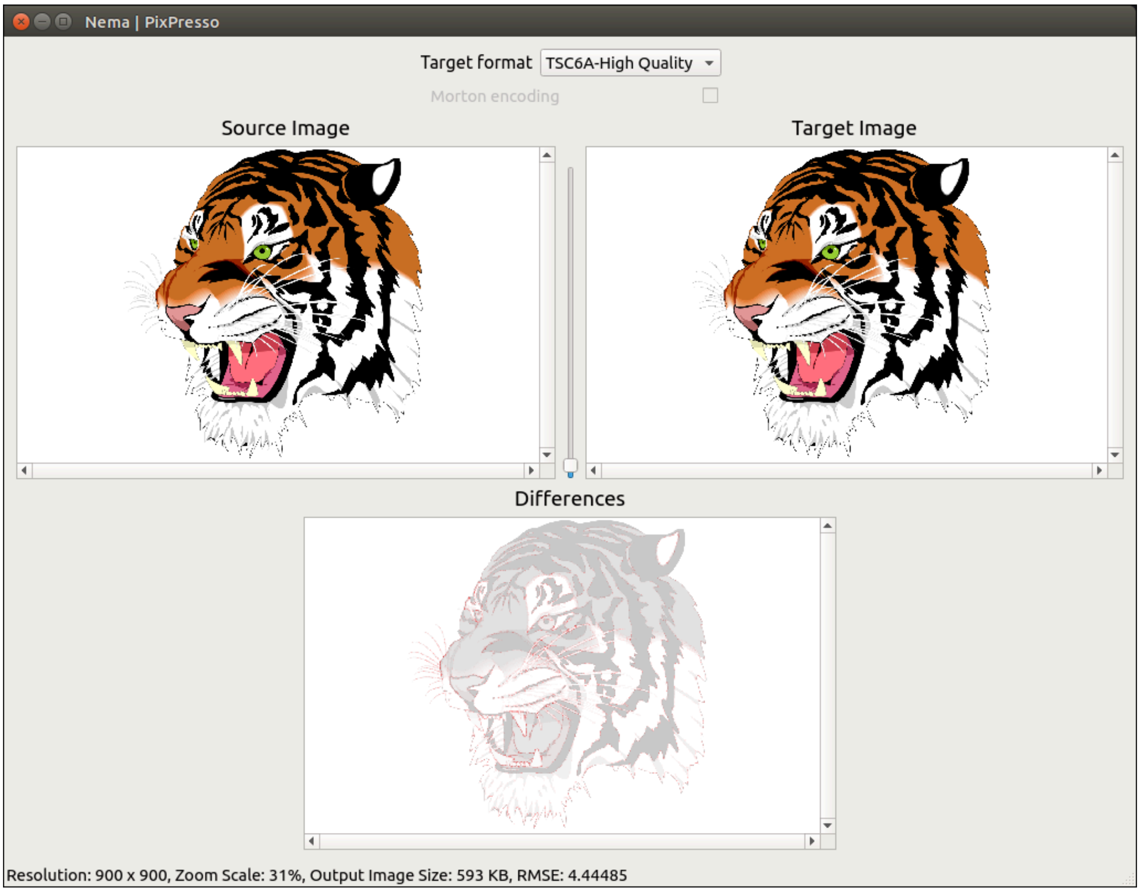

Figure 3: NEMA®| pix-presso state after a conversion

Along with the RMSE, the user can observe differences between the original and the converted image in the **Difference view** area (Figure 3), so that such differences can be spotted locally within the image. When the converted image matches exactly the original one, the user observes an over-lighted version of the original image. In case that errors exist (i.e. miscalculated values in the converted image due to compression), they are highlighted with red hue; the bigger an error is the closer to pure red this hue becomes (Figure 3).

The user can also zoom in or out (digital zoom) within the images to observe them in more details (from 0.03x up to 16x). The zoom functionality can be simply used by changing the value of the **zoom slider**. Alternatively, the user can press and hold the **Ctr key** on their keyboard and scroll the wheel of their mouse device while the mouse cursor is inside an **image view area**. When zooming, the user can observe that the scrollbars of the **image views areas** are auto adjusting. These scrollbars provide a helpful way of navigating within the image in case where the zoomed image cannot fit inside the **image view area**. The same functionality can also be achieved by dragging the mouse cursor in the image; in this case the user should see that the mouse cursor takes a "hand" form.

Another feature of the NEMA®| pix-presso is that the zoom is applied uniformly to all three **image view areas** as well as that corresponding scrollbars are synchronized with each other (Figure 4). Moving one scrollbar of an **image view area** makes the corresponding scrollbars in the other two **image view areas** move respectively. This allows the user to navigate within one image and also be able to observe the same region in the other two **image view areas**, without having to scroll the corresponding scrollbars (i.e. navigating within a sub-region of the **converted** image, makes the same sub-region of the **original** and the **difference** image be displayed).

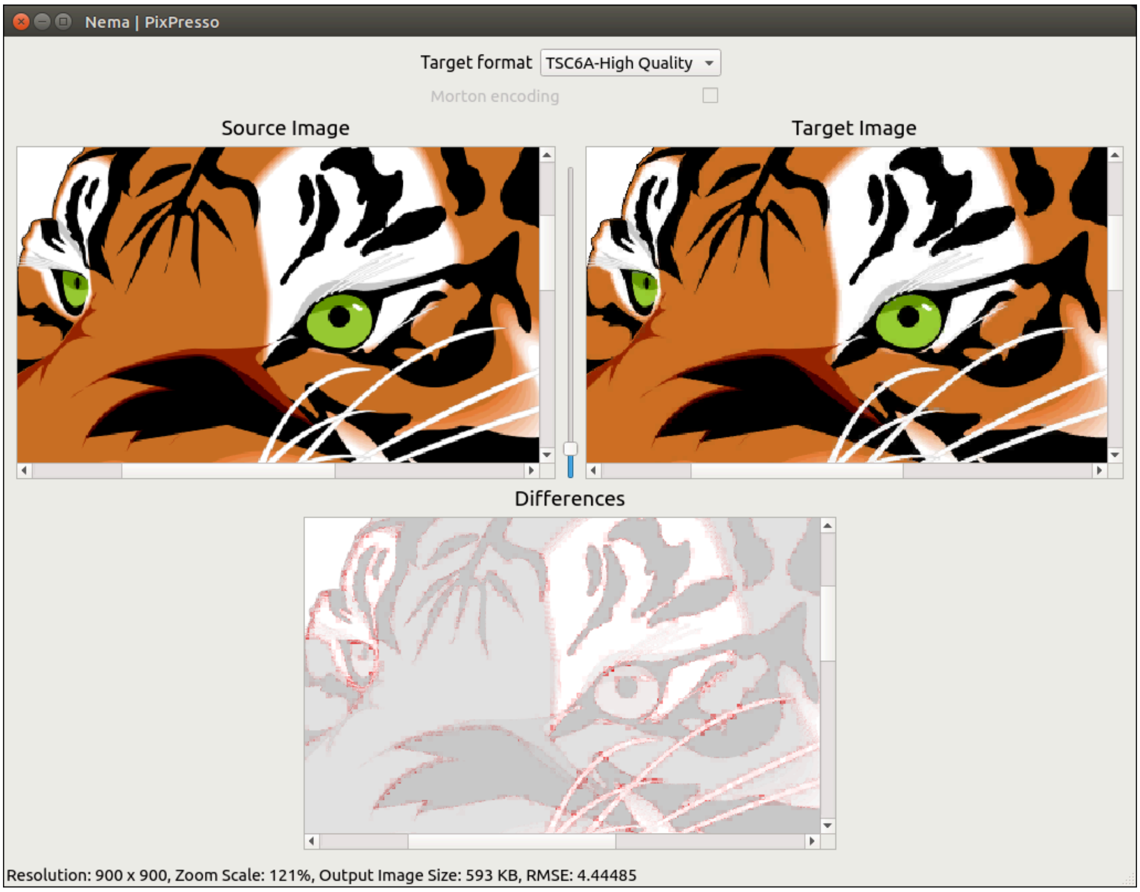

Figure 4: Zooming in a sub region of the original image

The target image can be scaled to a desired resolution, by selecting the desired resolution under the **Edit/Scale** menu. By doing so, the source image will be scaled as well before the conversion. This feature can be particularly usefull in case whereas an `SVG` image needs to be converted to a different format, due to the fact that the `SVG` format can be scaled to any resolution without any scaling artifacts.

Finally, the converted image can be exported in the desired format and be saved in a corresponding file. This is performed by pressing **Export** in the **File** menu. The file extension (i.e. `.tsc4`) is automatically added and the default name is set to be the same as the original image; nevertheless the user can freely set any desired name.

4.2 Command Line Mode

Using NEMA®| pix-presso at GUI Mode, provides an easy way in order to convert a single image and evaluate the conversion quality (differences between the original and the converted image). Nevertheless, NEMA®| pix-presso can also operate in command line mode. At this mode, the command line arguments are used to indicate the tool the source image path, the target image directory (the filename will be the same as the source image, with different file extension), the target image format, the source image resolution (when loading images in formats that do not store this information), the target image resolution (optinal), a flag that indicates wheter the source image is encoded using the Morton order code, another flag to indicate that the target image should be encoded using the Morton order code as well as a flag to indicate the tool that the target image should be exported as header file. The command line arguments are summarized in the following listing.

NEMA®| pix-presso Command Line Arguments

```
Usage options:
     -h,  --help          : Display this help message
     -s,  --src           : Source image path
     -d,  --dst           : Destination image directory (optional). If this field is
                            empty the destination image will be saved inside the
                            source image directory
     -f,  --format        : Target image format (PNG, RGBA8888, ABGR8888, BGRA8888,
                            ARGB8888, RGB24, BGR24, RGBA5650, RGBA5551, RGBA4444,
                            ARGB4444, RGBA3320, A8, A4, A2, A1, A4-LE, A2-LE, A1-LE,
                            L8, L4, L2, L1, L4-LE, L2-LE, L1-LE, TSC4F, TSC4HQ,
                            TSC6F, TSC6HQ, TSC6AF, TSC6AHQ, JPG, RGBA2222, BGRA2222,
                            ARGB2222, ABGR2222, LUT2, LUT4, LUT16, LUT256, AL44,
                            AL88, ARGB1555, BGRA5650, TSVG (beta), ABGR1555, BGRA5551
                            ABGR4444, BGRA4444)
     -sw, --src_width     : Source image width. Necessary when loading ALL
                            formats except png, svg and jpg
     -sh, --src_height    : Source image height. Necessary when loading ALL
                            formats except png, svg and jpg
     -dw, --dst_width     : Destination image width  (optional, in order to scale
                            the output image)
     -dh, --dst_height    : Destination image height (optional, in order to scale
                            the output image)
     -sm, --src_morton    : Source image is encoded using Morton code map
     -dm, --dst_morton    : Destination image will be written using Morton encoding
     -i,  --include       : Save the Destination image as header file (defined as
                            const array)
     -dith,  --dithering : Apply dithering when converting to 16/8bit formats
                            (ordered dithering)
     -op,  --opacity      : Parse opacity attibute (applies to TSVG target format only)
```

NEMA®| pix-presso can be executed in command line mode as following:

Linux based systems

```
$ ./nema_pixpresso -s /home/pictures/image.png -f RGBA8888
```

Windows based systems

```
> nema_pixpresso.exe -s "C:\pictures\image.png" -f RGBA8888
```

Running the above commands, will convert the source image (`image.png`), located in the `/home/pictures` (Linux based system) to RGBA8888 format. The converted image will be saved inside the same directory (`/home/pictures`) as `picture.rgba`. The target resolution is not defined, among the arguments therefore it will be the same as the source image. In order to save the image, the directory needs to have write permission rights. The same applies to Windows based systems. On Mac based systems where *nema_pixpresso.app* is provided, the command line mode can be executed as following:

```
> ./nema_pixpresso.app/Contents/MacOS/nema_pixpresso -s /home/pictures/image.png -f
 RGBA8888
```