NextGen Raster Command Line Reference 1.6

(c) Artwork Conversion Software Inc.

417 Ingalls St. Santa Cruz CA 95060

[T] +1 831 426 6163 [E] mailto:info@artwork.com [W] http://www.artwork.com

url: http://www.artwork.com/raster/nexgen_rip/index.htm

```
NextGenRaster.exe <Mandatory Arguments> [<Non-Cumulative Arguments>] [<Cumulative Arguments>]
```

Mandatory Arguments (Should appear only once at the beginning in the order specified)

`+input:${INPUT_FILE}`

- Specify the input GDSII/OASIS or DBLOAD cache (memory map) file to be loaded

`+out:${OUTPUT_BASE_PATH}`

- Specify the base path (directory + file name - file extension) where the output files will be created

Non-Cumulative Arguments (Order of appearance does not matter, Multiple occurrence overrides the previous one)

`-roi:${LLX},${LLY},${URX},${URY}`

- Specify region of interest (for generating tiles) with the lower left and upper right points in file units
- Tiling (-tiles) if specified, will be applied on this area of the home view
- If absent, the tiling will be applied to the entire home view
- This argument does not affect the behavior of -window arguments

`-layers:${LAYERLIST}`

- Specify a comma-separated list of layer[:datatype] to be loaded
- If no datatype is specified, all datatypes of that layer will be loaded
- Any layers or datatypes not covered by this list will be dropped as if they don't exist in the file
- e.g 1,2:1,2:2,3:5 Load all datatypes for layer 1, only 2:1, 2:2 for layer 2 and 3:5 for layer 3. Drop any other layer:datatype
- If absent, all layers in the file are loaded

`-cell:${VIEWCELL}`

- Specify the view cell
- The extents of this cell will become the home view
- By default, the top cell with the deepest tree is the view cell

`-pixelsize:${PIXELSIZE}`

- Specify the resolution of the raster image
- The pixel size is in file units
- e.g A pixel size of 1 micron is equivalent to a DPI of 25400
- If absent, pixel size is 1 micron

`-pixelsizeXY:${X_PIXELSIZE},${Y_PIXELSIZE}`

- Specify resolution of the raster image along X and Y
- Both pixel sizes are in file units
- If absent, the resolution is defaulted to 1,1

`-flat`

- Do not use pattern recognition to speed up the rasterization
- Every single polygon will be rasterized separately even if it repeats
- If absent, the Rasterizer uses cell hierarchy to detect repeating patterns within a single window to speed up the rasterization

`-thrnum:${NUM_THREADS}`

- Specify the number of threads to be used for rasterization and formatting
- If absent, the number of threads used is the number of CPUs in the system
- If -thrnum is 1 and -max-buffer-cnt is 0, polygon buffering will be disabled and every polygon in a window will be

rasterized on the fly (very memory efficient but slow)

`-max-buffer-cnt:${MAX_BUFFER_VERT}`

- For better memory management, specify an approximate polygon buffer size (as number of vertices)
- If specified (value > 0), the rasterization will be executed in batches of polygons such that the total number of vertices in each batch is approximately equal to the specified value
- If absent, a default batch size of 1,000,000 vertices will be used
- If the value specified is 0 and -thrnum > 1, the entire window will be buffered and rasterized at once (not memory efficient)
- If the value specified is 0 and -thrnum is 1, polygon buffering will be disabled and each polygon will be rasterized on the fly (very memory efficient but slow)

`-verbose`

- Print additional information to stdout and the log file (if specified)

`-invert`

- Invert the image polarity (white data on black background or scratch on paint)
- If absent, image polarity is black data on white background or paint

`-format:NONE | -format:TIFF | -format:BMP | -format:RAW`

- Specify output file format
- TIFF represents a TIFF file with packbits compression
- BMP represents a monochrome BMP file (uncompressed)
- RAW represents an uncompressed raw monochrome image. This format has a small header that starts with LGRAW00 followed by the image width in pixels (4 bytes), image height in pixels (4 bytes), total number of pixels (8 bytes) and finally the actual image data straight from the raster buffer
- NONE represents no output file. Each window is rasterized but the image is not written to disk
- If absent, -format:TIFF is used

`-fill:solid | -fill:outline | -fill:pathline | -fill:pathpoints`

- Specify a fill mode
- With solid fill, both the edges and the interior of all paths and boundaries is filled with pixels at the specified grey level. This is the default behavior
- With outline fill, only the edges of paths and boundaries is filled. Grey level is not used here
- With pathline fill (used typically with files containing only path data), paths are rasterized as 1 pixel lines connecting the end-points. Boundaries (if present) are rasterized using solid fill
- With pathpoints fill (used typically with files containing only path data), only the vertices of the paths are rasterized as 1 pixel dots. Boundaries (if present) are rasterized using solid fill

`-dither:${0.0_to_1.0}`

- Specify a value for ordered dithering using a 8x8 Bayer matrix.
- The number of pixels drawn in a 8x8 pixel block for solid fill areas decreases as the dither value changes from 1.0 (all pixels) to 0.0 (no pixels)

`-reverse`

- Reverse the direction of rasterization along X (Draw from right to left)
- Default: Draw from left to right

`-log:${LOGFILE} | -log+:${LOGFILE}`

- Enable logging to a file at the specified path (directory + file name + extension)

- If -log+ is used, append to the log file if it already exists. Otherwise, create a new file at the specified path
- If absent, no log file is generated

```
-silent
```

- Do not print messages on the console (stdout/stderr)
- If absent, emitcode, progress and information messages are printed to stdout, errors and warnings are printed to stderr

```
-emitcode
```

- Emit C++ code snippet to stdout/log file every time the QckRaster Library API is used
- Each code snippet is accompanied with C++ comments and parameter values
- This allows the user to understand the usage and flow of the QckRaster Library API
- Each line of the code emitted is preceded by [c++] to separate the it from messages from progress and information updates
- This option should be used for reference and diagnostic purposes only since it may affect performance
- Works in conjunction with -silent and -log

Cumulative Arguments (Order of appearance does not matter, Multiple occurrence has a cumulative effect)

```
-window:LLUR,${LLX},${LLY},${URX},${URY}
```

- Specify a raster window using the lower left and upper right points (in file units)
- -window commands are independent of the -roi command
- The actual raster window extents may be slightly adjusted to account for byte alignment
- Multiple -tiles and -window commands can be used cumulatively to generate a set of raster windows. The order of occurrence of these commands determines the order in which they are rasterized

```
-window:LLWH,${LLX},${LLY},${WIDTH},${HEIGHT}
```

- Specify raster window using the lower left point and width and height (in file units)
- -window commands are independent of the -roi command
- The actual raster window extents may be slightly adjusted to account for byte alignment
- Multiple -tiles and -window commands can be used cumulatively to generate a set of raster windows. The order of occurrence of these commands determines the order in which they are rasterized

```
-window:CWH,${CENTERX},${CENTERY},${WIDTH},${HEIGHT}
```

- Specify raster window using the center point and width and height (in file units)
- -window commands are independent of the -roi command
- The actual raster window extents may be slightly adjusted to account for byte alignment
- Multiple -tiles and -window commands can be used cumulatively to generate a set of raster windows. The order of occurrence of these commands determines the order in which they are rasterized

```
-tiles:NXY,${N_X},${N_Y}
```

- Break a region of interest into a fixed number of raster windows in X and Y
- The region of interest can be specified using the -roi command. If not specified, the home view is the region of interest
- Multiple -tiles and -window commands can be used cumulatively to generate a set of raster windows. The order of occurrence of these commands determines the order in which they are rasterized

```
-tiles:WH,${WIDTH},${HEIGHT}
```

- Break a region of interest into a fixed number of raster windows by width and height
- The region of interest can be specified using the -roi command. If not specified, the home view is the region of interest
- Multiple -tiles and -window commands can be used cumulatively to generate a set of raster windows. The order of occurrence of these commands determines the order in which they are rasterized

`-windows@${FILEPATH}`

- Specify a text file containing a set of -window and/or -tiles command (one per line) as an alternate way to specify raster windows
- The order of occurrence of these commands determines the order in which they are rasterized