

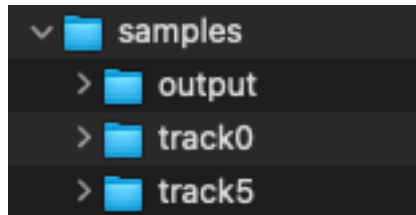
Some notes on specifying filename templates for MAX2spherebatch

The method chosen here for specifying sequences of files is fairly standard. One uses the % symbol as per standard C printf() command, in this case we only use %d plus length options. There are only two likely options one will use

%d by itself will expand to 1, 2, 3 etc

%04d will expand to 0001, 0002, 0003 etc

In this example there is a directory called samples, containing subdirectories track0 and track5. The files are named img0.jpg, img2.jpg and so on for each directory.



If one runs the MAX2spherebatch command from the directory above samples then one might choose, as a minimum:

```
MAX2spherebatch -n 1 samples/track%d/img%d.jpg
```

The first %d will be used to identify the track number, 0 or 5.

The second %d will be used to identify the frame index, 1 upwards.

Note the -n 1 is only needed because the software expects the first filename index to be 0.

Also note that the end image sequence index need not be specified, the software will exit if it fails to read the next image. But one could convert a subset of frames with

```
MAX2spherebatch -n 20 -m 30 samples/track%d/img%d.jpg
```

The result of the above will be frames located in the track0 directory with "_sphere" concatenated to the name. eg: img1_sphere.jpg, img2_sphere.jpg and so on.

If one creates a directory calls, say, "output" then one can do this

```
MAX2spherebatch -n 1 -o samples/output/sphere%03d.jpg samples/track%d/img%d.jpg
```

In this case the spherical images will be saved in the "output" directory and called sphere001.jpg, sphere002.jpg and so on.

Usage string

Typing MAX2spherebatch by itself gives some help on the options, currently as follows:

```
Usage: MAX2spherebatch [options] sequencetemplate
```

The sequence filename template should contain two %d entries. The first will be populated with the track number 0 or 5, the second is the frame sequence number, see -n and -m below. So for example, if there are 1000 frames called track0_frame0001.jpg, track5_0001.jpg, ... then the program might be called as follows:

```
MAX2spherebatch -w 4096 -n 1 -m 1000 track%d_frame%04d.jpg
```

Or if directories are used with frames track0/frame1.jpg, track5/1000.jpg, ...

```
MAX2spherebatch -w 4096 -n 1 -m 1000 track%d/frame%4d.jpg
```

Options

```
-w n    Sets the output image width, default: -1
-a n    Sets antialiasing level, default = 2
-o s    Specify the output filename template, default is based on track 0 name uses track 2
        If specified then it should contain one %d field for the frame number
-n n    Start index for the sequence, default: 0
-m n    End index for the sequence, default: 100000
-d      Enable debug mode, default: off
```

Using the -d option can be helpful if things aren't working out. And if you are reporting errors please provide the terminal output with -d enabled.

How lookuptables are handled.

MAX2spherebatch will first look for a lookup table.

If it finds one it will read it and use it during the current processing run.

If it doesn't find a lookup table (or if the read above fails) it will create one and save it to disk and then use it during the current processing run.

Lookup tables rely on four values: the template number, the output width and height, and the antialiasing value. In the case above where the output image width was autodetermined the lookup table is called

0_5376_2688_2.data

Template 0, width 5376, height 2688 and antialiasing of 2.

The lookup table does take almost no time to read, compared to calculating the lookup table. However it does end up taking a decent amount of disk space, almost 700MB in this case.