epeg egg

Chicken bindings for the epeg thumbnailer library.
Extension for Chicken Scheme
Version 2.0

Peter Bex

Table of Contents

1 About this egg

1.1 Version history

`2.0` Change API to use `epeg:` prefix for procedures. Change to eggdoc documentation format. Use GC finalizers.

`1.1` Added checks so (where it's possible) exceptions get thrown instead of segfaulting.

`1.0` initial release.

1.2 Requirements

This egg requires the following extensions:

```
syntax-case
```

1.3 Usage

Load this egg like so:

```
(require-extension epeg)
```

2 Documentation

2.1 General overview

Epeg works by first opening a file, then setting the desired properties of the scaled image. Upon encoding, these properties are taken into account. Decoding is deferred until encoding of the new image takes place for optimal speed.

2.2 Image creation, destruction and friends

`epeg:file-open` [procedure]

```
(epeg:file-open filename)
```

Returns a new `epeg:image` object which describes the image stored in the file `filename`.

`epeg:image?` [procedure]

```
(epeg:image? img)
```

Determine if the given `img` object is an epeg image.

`epeg:close` [procedure]

```
(epeg:close img)
```

Destroy the image and close the associated file.

`epeg:encode` [procedure]

```
(epeg:encode img)
```

Store the image to the file as determined by `epeg:file-output-set!`. For other properties of the image, see below.

2.3 Image properties

`epeg:size-get` [procedure]

```
(epeg:size-get img)
```

Returns 2 values: the width and height of the *original* image.

`epeg:file-output-set!` [procedure]

```
(epeg:file-output-set! img filename)
```

Set the output file for subsequent calls to `epeg:encode`.

`epeg:decode-size-set!` [procedure]

```
(epeg:decode-size-set! img width height)
```

Set the size of the new image, in pixels. The original image is decoded into this size.

`epeg:decode-bounds-set!` [procedure]

```
(epeg:decode-bounds-set! img x y width height)
```

Set the bounds of the image to be decoded. This is like cropping the image upon loading it.

`epeg:decode-colorspace-set!` [procedure]

```
(epeg:decode-colorspace-set! img colorspace)
```

Set the colorspace for decoding. This is one of the following:

- `epeg:colorspace-gray8`
- `epeg:colorspace-yuv8`
- `epeg:colorspace-rgb8`
- `epeg:colorspace-bgr8`
- `epeg:colorspace-rgba8`
- `epeg:colorspace-bgra8`
- `epeg:colorspace-argb32`
- `epeg:colorspace-cmyk`

`epeg:colorspace-get` [procedure]

```
(epeg:colorspace-get img)
```

Returns the current colorspace value to use for decoding.The values returned by this is any of the constantsdescribed under `epeg:decode-colorspace-set!`.

`epeg:comment-set!` [procedure]

```
(epeg:comment-set! img comment)
```

Set the comment string for encoding. By default, this is not written to the thumbnail! Use `epeg:thumbnail-comments-enable` to enable this.

`epeg:comment-get` [procedure]

```
(epeg:comment-get img)
```

Returns the comment string to use for encoding, or `#f` if none was set.

`epeg:thumbnail-comments-enable` [procedure]

```
(epeg:thumbnail-comments-enable img)
```

Enable comment field for the thumbnail.

`epeg:thumbnail-comments-enable` [procedure]

```
(epeg:thumbnail-comments-enable img)
```

Disable comment field for the thumbnail. The default setting for comments is disabled.

`epeg:thumbnail-comments-get` [procedure]

```
(epeg:thumbnail-comments-get img)
```

Returns the uri field, width, height and mimetype of the thumbnail that will be written.

`epeg:quality-set!` [procedure]

```
(epeg:quality-set! img percentage)
```

Set the thumbnail's image quality (0 - 100).

`epeg:trim` [procedure]

```
(epeg:trim img)
```

Undocumented

2.4 Pixel query functions

Currently, the pixel query functions have been disabled.

3 Examples

```
;;
;; Almost literal translation of src/bin/epeg_main.c in the epeg distribution.
;;
(use epeg)

(unless (= (length (argv)) 3)
        (printf "Usage: ~A input.jpg thumb.jpg\n" (car (argv)))
        (exit 0))

; This will throw an i/o file exception if the file can't be opened
(define im (epeg:file-open (list-ref (argv) 1)))

(let ((com (epeg:comment-get im)))
    (when com (printf "Comment: ~A\n" com)))

(call-with-values
    (lambda ()
      (epeg:thumbnail-comments-get im))
  (lambda (uri width height mimetype)
    (when mimetype
          (printf "Thumb Mimetype: ~A\n" mimetype)
          (when uri
            (printf "Thumb URI: ~A\n" uri))
          (printf "Thumb Mtime: \n")
          (printf "Thumb Width: ~A\n" width)
          (printf "Thumb Height: ~A\n" height))))

(call-with-values
    (lambda ()
      (epeg:size-get im))
  (lambda (width height)
    (printf "Image size: ~Ax~A\n" width height)))

(epeg:decode-size-set! im 128 96)
(epeg:quality-set! im 80)
(epeg:thumbnail-comments-enable im)
(epeg:comment-set! im "Smelly pants!")
(epeg:file-output-set! im (list-ref (argv) 2))
(epeg:encode im)
(epeg:close im)  ; Not required

(exit 0)
```

4 License

```
Copyright (c) 2004 - 2006, Peter Bex (peter.bex@xs4all.nl)
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. Neither the name of Peter Bex nor the names of any contributors may
   be used to endorse or promote products derived from this software
   without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY PETER BEX AND CONTRIBUTORS ‘‘AS IS’’ AND ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED.  IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.
```

Index