

box egg

Boxing
Extension for Chicken Scheme
Version 1.3

Kon Lovett

Table of Contents

| | | |
|----------|-----------------------------|----------|
| 1 | About this egg | 1 |
| 1.1 | Version history | 1 |
| 1.2 | Usage | 1 |
| 2 | Documentation | 2 |
| 2.1 | Boxed Values | 2 |
| 2.2 | MzScheme-Like | 3 |
| 3 | License | 4 |
| | Index | 5 |

1 About this egg

1.1 Version history

- 1.3 Exports
- 1.2 Added MzScheme-like box api, changed references
- 1.1 Added references, different implementation of box
- 1.0 Initial release

1.2 Usage

Load this egg like so:

```
(require-extension box)
```

2 Documentation

This egg is **unsafe**.

2.1 Boxed Values

`make-box` [procedure]

(`make-box` [INITIAL-VALUE (void)])

Returns a box with the optional INITIAL-VALUE.

`make-box-immutable` [procedure]

(`make-box-immutable` VALUE)

Returns a box with the VALUE. Signals an error upon attempted set.

`make-box-variable` [macro]

(`make-box-variable` VARIABLE)

Returns a boxed reference to the variable VARIABLE, which must be in lexical-scope.

`box?` [procedure]

(`box?` BOX)

Is it a box?

`box-mutable?` [procedure]

(`box-mutable?` BOX)

Is it a mutable box?

`box-immutable?` [procedure]

(`box-immutable?` BOX)

Is it an immutable box?

`box-variable?` [procedure]

(`box-variable?` BOX)

Is it a boxed variable?

`box-set!` [macro]

(`box-set!` BOX VALUE)

Sets the boxed value to VALUE.

`box-ref` [macro]

(`box-ref` BOX)

Returns the boxed value.

2.2 MzScheme-Like

Needs (use box-mzs).

box [macro]

(box INITIAL-VALUE)

Returns a box with the INITIAL-VALUE.

box-immutable [macro]

(box-immutable VALUE)

Returns a box with the VALUE. Signals an error upon attempted set.

box-variable [macro]

(box-variable VARIABLE)

Returns a boxed reference to the variable VARIABLE, which must be in lexical-scope.

set-box! [macro]

(set-box! BOX VALUE)

Sets the boxed value to VALUE.

unbox [macro]

(unbox BOX)

Returns the boxed value.

3 License

Copyright (c) 2006, Kon Lovett. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the Software), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED ASIS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Index

B

| | |
|----------------------|---|
| box | 3 |
| box-immutable | 3 |
| box-immutable? | 2 |
| box-mutable? | 2 |
| box-ref | 2 |
| box-set! | 2 |
| box-variable | 3 |
| box-variable? | 2 |
| box? | 2 |

M

| | |
|--------------------------|---|
| make-box | 2 |
| make-box-immutable | 2 |
| make-box-variable | 2 |

S

| | |
|----------------|---|
| set-box! | 3 |
|----------------|---|

U

| | |
|-------------|---|
| unbox | 3 |
|-------------|---|