

# srfi-45 egg

---

Primitives for Expressing Iterative Lazy Algorithms  
Extension for Chicken Scheme  
Version 1.0

Kon Lovett

---

## Table of Contents

<b>1</b>	<b>About this egg .....</b>	<b>1</b>
1.1	Version history .....	1
1.2	Usage .....	1
1.3	Requirements .....	1
<b>2</b>	<b>Documentation .....</b>	<b>2</b>
<b>3</b>	<b>License .....</b>	<b>4</b>
	<b>Index .....</b>	<b>5</b>

# 1 About this egg

## 1.1 Version history

1.0            Initial release

## 1.2 Usage

Load this egg like so:

```
(require-extension srfi-45)
```

## 1.3 Requirements

This egg requires the following extensions:

```
box.egg
```

## 2 Documentation

Extends the interpretation of a promise to include lazy and eager promises. Unlike the reference implementation this version does not redefine 'delay'.

For more information see [SRFI-45](#).

**delay** [macro]

(delay EXPRESSION)

Returns a promise, a delayed evaluation of EXPRESSION.

Note that 'delay' is *not* re-defined here. Mentioned only for completeness.

**lazy** [macro]

(lazy EXPRESSION)

Returns a promise for EXPRESSION.

The promise will be iteratively forced, overwriting the promise at each step before the next iteration, so as to avoid storage leaks.

**eager** [macro]

(eager EXPRESSION)

Returns a promise but immediately evaluates EXPRESSION.

Not much of a promise, more of a boxed value. For exposition purposes in algorithms.

**force** [procedure]

(force OBJECT)

Assuming the OBJECT is some kind of promise, returns the result of the delayed evaluation.

**standard-force** [procedure]

(standard-force OBJECT)

Assuming the OBJECT is a standard promise, returns the result of the delayed evaluation.

The R5RS 'force'.

**promise?** [procedure]

(promise? OBJECT)

Some kind of promise?

**standard-promise?** [procedure]

(standard-promise? OBJECT)

The R5RS 'promise?'

**recursive-promise?** [procedure]

(recursive-promise? OBJECT)

Recursive promise?

**lazy-promise?** [procedure]

(lazy-promise? OBJECT)

Lazy recursive promise?

`eager-promise?`

[procedure]

(`eager-promise?` OBJECT)

Eager recursive promise?

This egg is **unsafe**!

### 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.

Does not supersede any restrictions in the source.

## Index

### D

delay ..... 2

### E

eager ..... 2

eager-promise? ..... 3

### F

force ..... 2

### L

lazy ..... 2

lazy-promise? ..... 2

### P

promise? ..... 2

### R

recursive-promise? ..... 2

### S

standard-force ..... 2

standard-promise? ..... 2