box egg

Boxing
Extension for Chicken Scheme
Version 1.3

Kon Lovett

Table of Contents

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

Index