

# uuid-lib egg

---

OSF DCE 1.1 UUID  
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	Requirements .....	1
1.3	Usage .....	1
<b>2</b>	<b>Documentation .....</b>	<b>2</b>
2.1	Predicates .....	2
2.2	Operations .....	2
2.3	Errors .....	3
<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 Requirements

This egg requires the following extensions:

`syntax-case`

## 1.3 Usage

Load this egg like so:

`(require-extension uuid-lib)`

## 2 Documentation

Wrapper for `uuid/uuid.h` procedures on MacOS X, and Linux.

This is a module. You must `(import uuid-lib)` to access these procedures.

### 2.1 Predicates

`uuid?` [procedure]

(`uuid? UUID`)

Is `UUID` a `uuid`?

`uuid-null?` [procedure]

(`uuid-null? UUID`)

Is `UUID` the "null" `uuid`?

`uuid=?` [procedure]

(`uuid=? UUID1 UUID2`)

Are `UUID1` and `UUID2` equal?

`uuid<>?` [procedure]

(`uuid<>? UUID1 UUID2`)

Are `UUID1` and `UUID2` not equal?

`uuid<?` [procedure]

(`uuid<? UUID1 UUID2`)

Does `UUID1` order below `UUID2`?

`uuid>?` [procedure]

(`uuid>? UUID1 UUID2`)

Does `UUID1` order above `UUID2`?

`uuid<=?` [procedure]

(`uuid<=? UUID1 UUID2`)

Does `UUID1` order below or the same as `UUID2`?

`uuid>=?` [procedure]

(`uuid>=? UUID1 UUID2`)

Does `UUID1` order above or the same as `UUID2`?

### 2.2 Operations

`uuid-clear!` [procedure]

(`uuid-clear! UUID`)

Returns the `UUID` set to the null `uuid`.

`uuid-copy` [procedure]

(`uuid-copy UUID`)

Returns a copy of the `UUID`.

`uuid-generate` [procedure]

`(uuid-generate [#:method #f])`

Returns a new uuid.

Method is `#f`, `'random`, or `'time`.

`uuid-parse` [procedure]

`(uuid-parse UUID-TEXT)`

Returns a new uuid from the UUID external string representation `UUID-TEXT`.

`uuid-unparse` [procedure]

`(uuid-unparse UUID [#:case #f])`

Returns the external string representation of `UUID`.

Case is `#f`, `'upper`, or `'lower`.

## 2.3 Errors

Argument errors are signalled using `error`.

When a non-uuid is given as an argument the result is `#f`.

### 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

uuid-clear! .....	2	uuid<>? .....	2
uuid-copy .....	2	uuid<? .....	2
uuid-generate .....	3	uuid=? .....	2
uuid-null? .....	2	uuid>=? .....	2
uuid-parse .....	3	uuid>? .....	2
uuid-unparse .....	3	uuid? .....	2
uuid<=? .....	2		