Σ
A Library for Ansi Common Lisp

Christopher Mark Gore
cgore@cgore.com
http://www.cgore.com

March 19, 2013

Contents

Chapter 1

Copyright

8729 Lower Marine Road, Saint Jacob, Illinois 62281 USA.
Web: `http://cgore.com`
Email: `cgore@cgore.com`

Chapter 2

The Behave Package

2.1 Macros

2.1.1 The Behavior Macro

2.1.2 The Spec Macro

2.1.3 The Should Macro

2.1.4 The Should-Not Macro

2.1.5 The Should-Be-Null Macro

2.1.6 The Should-Be-A Macro

2.1.7 The Should= Macro

2.1.8 The Should/= Macro

2.1.9 The Should< Macro

2.1.10 The Should> Macro

2.1.11 The Should<= Macro

2.1.12 The Should>= Macro

2.1.13 The Should-Eq Macro

2.1.14 The Should-Not-Eq Macro

2.1.15 The Should-Eql Macro

2.1.16 The Should-Not-Eql Macro

2.1.17 The Should-Equal Macro

2.1.18 The Should-Not-Equal Macro

2.1.19 The Should-EqualP Macro

2.1.20 The Should-Not-EqualP Macro

2.1.21 The Should-String= Macro

2.1.22 The Should-Not-String= Macro

2.1.23 The Should-String/= Macro

Chapter 3

The Control Package

3.1 Macros

3.1.1 The AIf Macro

3.1.2 The A?If Macro

3.1.3 The AAnd Macro

3.1.4 The A?And Macro

3.1.5 The ALambda Macro

3.1.6 The A?Lambda Macro

3.1.7 The ABlock Macro

3.1.8 The A?Block Macro

3.1.9 The ACond Macro

3.1.10 The A?Cond Macro

3.1.11 The AWhen Macro

3.1.12 The A?When Macro

3.1.13 The AWhile Macro

3.1.14 The A?While Macro

3.1.15 The DeleteF Macro

3.1.16 The Do-While Macro

3.1.17 The Do-Until Macro

3.1.18 The For Macro

3.1.19 The Forever Macro

3.1.20 The Multicond Macro

3.1.21 The OpF Macro

3.1.22 The Swap Macro

3.1.23 The Swap-Unless Macro

Chapter 4

The Numeric Package

Chapter 5

The OS Package

5.1 Functions

5.1.1 The Perl Function

5.1.2 The Python Function

5.1.3 The Read-File Function

5.1.4 The Read-Lines Function

5.1.5 The Ruby Function

5.2 Parameters

5.2.1 The *Perl-Path* Parameter

5.2.2 The *Python-Path* Parameter

5.2.3 The *Ruby-Path* Parameter

Chapter 6

The `Probability` Package

6.1 Macros

6.1.1 The `Decaying-Probabiliity?` Macro

6.2 Functions

6.2.1 The `Probability?` Function

6.3 Types

6.3.1 The `Probability` Type

Chapter 7

The `Random` Package

7.1 Macros

7.1.1 The `NShuffle` Macro

7.2 Functions

7.2.1 The `Gauss` Function

7.2.2 The `Random-Argument` Function

7.2.3 The `Coin-Toss` Function

7.2.4 The `Random-In-Range` Function

7.2.5 The `Random-In-Ranges` Function

7.2.6 The `Random-Range` Function

7.2.7 The `Randomize-Array` Function

7.2.8 The `Random-Array` Function

7.3 Generics

7.3.1 The `Random-Element` Generic

7.3.2 The `Shuffle` Generic

Chapter 8

The Sequence Package

8.1 Macros

8.1.1 The Arefable? Macro

8.1.2 The NConcF Macro

8.1.3 The Nthable? Macro

8.1.4 The Set-NthCdr Macro

8.2 Functions

8.2.1 The Array-Values Function

8.2.2 The Nth-From-End Function

8.2.3 The Sequence? Function

8.2.4 The Empty-Sequence? Function

8.2.5 The Join-Symbol-To-All-Following Function

8.2.6 The Join-Symbol-To-All-Preceeding Function

8.2.7 The List-To-Vector Function

8.2.8 The Set-Equal Function

8.2.9 The Simple-Vector-To-List Function

8.2.10 The Sort-Order Function

8.2.11 The The-Last Function

8.2.12 The Vector-To-List Function

8.3 Generics

8.3.1 The Best Generic

8.3.2 The Minimum Generic

8.3.3 The Minimum? Generic

8.3.4 The Maximum Generic

Chapter 9

The `String` Package

The `String` package contains useful tools for working with strings.

9.1 Functions

9.1.1 The `Character-Range` Function

The `character-range` function returns a list of characters from the *start* to the *end* character. Note that this is returning a list, not a string.

Syntax

`(character-range` *start end*`)` $\Longrightarrow$ `'(`*start ... end*`)`

Arguments and Values

Start The character to start the range with, inclusive.

End The character to end the range with, inclusive.

Examples

```
(character-range #\a #\e) ⟹ '(#\a #\b #\c #\d #\e)
(character-range #\e #\a) ⟹ '(#\a #\b #\c #\d #\e)
```

9.1.2 The `Character-Ranges` Function

The `character-ranges` function is a convenience wrapper for `character-range` function, concatenating several calls and making the resultant list contain only unique instances.

Syntax

`(character-ranges` $start_1$ end_1 ... $\Longrightarrow$ `'(`$character_1$ ...`)`

Arguments and Values

Start$_n$ The character to start the nth range with, inclusive.

End$_n$ The character to end the nth range with, inclusive.

Examples

```
(character-ranges #\a #\c #\x #\z) ⟹ '(#\a #\b #\c #\x #\y #\z)
(character-ranges #\a #\c #\a #\c) ⟹ '(#\a #\b #\c)
```

9.1.3 The `Escape-Tildes` Function

9.1.4 The `Replace-Char` Function

9.1.5 The `StrCat` Function

9.1.6 The `StrMult` Function

9.1.7 The `String-Join` Function

9.1.8 The `Stringify` Function

9.1.9 The `To-String` Function

9.2 Methods

9.2.1 The `Split` Methods

Chapter 10

The `Time-Series` Package

10.1 Macros

10.1.1 The `Snap-Index` Macro

10.2 Functions

10.2.1 The `Array-Raster-Line` Function

10.2.2 The `Distance` Function

10.2.3 The `Norm` Function

10.2.4 The `Raster-Line` Function

10.2.5 The `Similar-Points?` Function

10.2.6 The `Time-Series?` Function

10.2.7 The `Time-Multiseries?` Function

10.2.8 The `TMSref` Function

10.2.9 The `TMS-Dimensions` Function

10.2.10 The `TMS-Raster-Line` Function

10.2.11 The `TMS-Values` Function

10.3 Types

10.3.1 The `Time-Multiseries` Type

Chapter 11

The `Truth` Package

11.1 Functions

11.1.1 The `[?]` Function

11.1.2 The `Toggle` Function

11.2 Generics

11.2.1 The `?` Generic

Chapter 12

The `Utilities` Package

12.1 Variables

12.1.1 The `*CGore-Utilities-Packages*` Variable

12.2 Functions

12.2.1 The `Use-All-CGore-Utilities` Function

12.2.2 The `Use-All-Utilities` Function