

futures egg

A straightforward implementation of "futures" for concurrent evaluation of expressions.
Extension for Chicken Scheme
Version 1.1

Table of Contents

1	About this egg	1
1.1	Version history	1
1.2	Usage	1
2	Documentation	2
3	License	3
	Index	4

1 About this egg

1.1 Version history

- 1.1 Thomas Chust added some missing functions
- 1.0 Initial release

1.2 Usage

Load this egg like so:

```
(require-extension futures)
```

2 Documentation

This extension provides syntax and data types for *futures* - concurrently evaluated expressions that memoize their result

future [macro]
 (future EXPRESSION)

Starts evaluation of EXPRESSION in a separate thread and returns a *future* object representing the computation.

future? [procedure]
 (future? X)

Returns **#t** if X is an object returned by a **future** expression or **#f** otherwise.

force [procedure]
 (force X)

If X is a future, then the current threads waits until the expression evaluated at the point of creating the future is complete. If an error occurred during that evaluation, or when the future executed to completion, then **force** will return the result(s) of the future expression.

If the argument is not a future, then the usual semantics of **force** apply.

future-complete? [procedure]
 (future-complete? FUTURE)

Returns **#t** if the future has completed execution, or if an exception occurred during its evaluation, or **#f** otherwise.

future-failed? [procedure]
 (future-failed? FUTURE)

Returns **#t** if an exception occurred during evaluation of the future or **#f** otherwise.

future-condition [procedure]
 (future-condition FUTURE)

Returns the error-condition if an exception occurred during evaluation of the future or **#f** if the future is still executing or executed without an error.

3 License

Copyright (c) 2005, Felix L. Winkelmann. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Index

force.....	2	future-condition.....	2
future.....	2	future-failed?.....	2
future-complete?.....	2	future?.....	2