proplist egg

Provides property list operations.
Extension for Chicken Scheme
Version 1.2

Kon Lovett

Table of Contents

1 About this egg

1.1 Version history

`1.2`	Exports
`1.1`	"Real" property list
`1.0`	Initial release

1.2 Requirements

This egg requires the following extensions:

`srfi-1`, `srfi-69`

1.3 Usage

Load this egg like so:

```
(require-extension proplist)
```

2 Documentation

proplist is a lisp inspired set of procedures supporting property list operations.

proplist supports the pre-2.2 CHICKEN "disembodied" property list functionality, but without the original functions (get) and (put!), which can be easily defined.

2.1 Property list

Lisp-like property list procedures.

The name `NAME` and property key`PROPKEY` arguments below should be interned symbols. Identity matching is performed.

`getprop` [procedure]

```
(getprop NAME PROPKEY [DEFVAL])
```

Returns the first property `PROPKEY` value of name `NAME`, or the optional default value `DEFVAL`, when no such property. When a default value is not supplied #f is assumed.

`putprop!` [procedure]

```
(putprop! NAME [PROPKEY VALUE] ...)
```

Sets the first property `PROPKEY` value of name `NAME` to the supplied value `VALUE`. Does not replace an existing property!

`setprop!` [procedure]

```
(setprop! NAME [PROPKEY VALUE] ...)
```

Sets the property `PROPKEY` value of name `NAME` to the supplied value `VALUE`. Replaces any existing property!

`remprop!` [procedure]

```
(remprop! NAME [PROPKEY] ...)
```

Removes the first property `PROPKEY` of name `NAME`.

`remprop/all!` [procedure]

```
(remprop/all! NAME [PROPKEY])
```

Removes every property `PROPKEY` value of name `NAME` matching the specified key, or removes name when a key is not specified.

`get-proplist` [procedure]

```
(get-proplist NAME [PROPKEY ...])
```

Returns the first property `PROPKEY` value of name `NAME` for every property key specified, as a property list. All properties returned when no key specified.

`get-properties` [procedure]

```
(get-properties NAME [PROPKEY ...])
```

Returns the first property `PROPKEY` value of name `NAME` for every property key specified, as an association list. All properties returned when no key specified.

`rem-properties!` [procedure]

```
(rem-properties! NAME PROPKEY ...)
```

Removes the first property `PROPKEY` value of name `NAME` for every property specified.

`rem-properties/all!` [procedure]

```
(rem-properties/all! NAME [PROPKEY ...])
```

Removes every property `PROPKEY` value of name `NAME` for every property specified, or deletes name when a key is not specified.

2.2 Disembodied property list

Pre-2.2 CHICKEN-like property list procedures. These are not particularly useful, as no new functionality is provided. However, they allow the property list procedures to accept hash-tables, and do something reasonable.

`getprop` [procedure]

```
(getprop HASHTABLE PROPKEY [DEFVAL])
```

Returns the property `PROPKEY` value of the hashtable `HASHTABLE`, or the optional default value`DEFVAL`, when no such property. When a default value is not supplied #f is assumed.

`putprop!` [procedure]

```
(putprop! HASHTABLE [PROPKEY VALUE] ...)
```

Sets the property `PROPKEY` value of the hashtable `HASHTABLE` to the supplied value `VALUE`.

`setprop!` [procedure]

```
(setprop! HASHTABLE [PROPKEY VALUE] ...)
```

Sets the property `PROPKEY` value of hashtable `HASHTABLE` to the supplied value `VALUE`. Replaces any existing property!

`remprop!` [procedure]

```
(remprop! HASHTABLE PROPKEY ...)
```

Removes the property `PROPKEY` from the hashtable `HASHTABLE`.

2.3 Property list I/O

Property list SRFI-10 support procedures.

A reader-ctor is defined for the external text representation of a property list.

`print-proplist` [procedure]

```
(print-proplist NAME [PORT])
```

Writes the external text representation of the property list for name `NAME` to the output port `PORT`. Port defaults to the (current-output-port).

`set-proplist!` [procedure]

```
(set-proplist! NAME PROPLIST)
```

Sets the property list for name `NAME` to the supplied property list `PROPLIST`.

Though not an I/O operation this procedure is considered too dangerous for regular use and buried in this section.

3 License

Copyright (c) 2005, 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