format-modular egg

A generic implementation of Common LISPs `format` facility.
Extension for Chicken Scheme
Version 1.2

Alejandro Forero Cuervo and Alex Shinn

Table of Contents

1 About this egg

1.1 Version history

`1.2` bugfix in formatter-jump

`1.1 (r4851)`
Small bug fixes, more annotations to the source code

`1.0 (r868)`
First public release

1.2 Usage

Load this egg like so:

```
(require-extension format-modular)
```

2 Documentation

Function that allows you to create formatter functions similar to Common Lisp's format or SRFI 28's format) in a very flexible and modular manner. The resulting functions are reentrant (and thus thread safe).

`make-format-function` [procedure]

```
(make-format-function case-sensitive escape formatters)
```

Create and return a procedure that can be used as a format function. The resulting procedure receives a port, a format string and optional arguments. It parses the format string according to the `formatters` parameter and outputs a result to the port specified. If, instead of a port, `#t` is specified, the results will be sent to the current output port. If, on the other hand, `#f` is passed, the procedure will store its output on a new string and return it.

The returned procedure parses the format string looking for occurrences of the `escape` character (usually `#~`; when one is found, a function from the `formatters` argument is called.

`formatters` is a list of formatters. Each formatter is itself a list of `(char function)` pairs, where `char` is the character for the escape sequence being defined and `function` the function to call when the character, preceeded by `escape<`, is found in the format string.

To produce the functions included in the formatters, use `(formatter-function proc)`, where `proc` is a procedure that receives the following parameters:

`state`
: A structure with internal information. You won't normally inspect this directly but rather pass it to other functions that require it (such as `out-char` and `out-string`).

`start`
: The position in the format string where the `escape` sequence was found. You'll normally just ignore this.

`params`
: A list with paramters that occur between `escape` and `char`(for example, in a format string such as `~23,33,12A`).

`colon`, `atsign`
: Booleans indicating whether those characters occur between `escape` and `char`.

To output info at the current position, the formatter functions should use the `out-char` and `out-string` functions. The receive the `state` parameter (as passed to the formatter) and a character and string respectively.

`format` [procedure]

```
(format port fmtstring . args)
```

This procedure is compliant to SRFI-28

This implementation of `FORMAT` supports the following escape sequences:

`?!P{}^()%/~|_XDOB[;]?*ASCRFEQGT`

The following are known to be missing:

`$&|IY`

If you know of another unsupported standard escape sequence, please contact the authors.

Please consult a Common LISP format reference manual for a detailed description of the format string syntax.

3 LicenseThe Modular Format egg for Chicken Scheme is in the public domain and may be reproduced or copied without

permission from its authors. Citation of the source is appreciated.

Index

F

M