locale egg

Provides locale operations.
Extension for Chicken Scheme
Version 0.2

Kon Lovett

Table of Contents

1 About this egg 1

1.1 Version history 1
1.2 Usage 1
1.3 Requirements 1

2 Documentation 2

2.1 Timezone 2
2.2 Locale 4
2.3 Locale Category 5

3 License 6

Index 7

1 About this egg

1.1 Version history

`0.2` Exports

`0.1` Initial release

1.2 Usage

Load this egg like so:

```
(require-extension locale)
```

1.3 Requirements

This egg requires the following extensions:

`miscmacros`

2 Documentation

locale is a set of routines supporting locale query operations. The environment locale information is queried upon module load and the corresponding parameters are set.

NOTE: This is a work in progress. Currently only the Posix locale information is supported. Plans are to support the native MacOS X and Windows locale APIs. Changes to this API are almost certain.

2.1 Timezone

Access to local timezone information. A timezone object is composed of a Standard Time Name and Offset, and an optional Summer or Daylight Savings Time Name and Offset. The offset is seconds west (negative) or east (positive) of UTC. The name is some locally accepted timezone name, such as PST. A Daylight Savings Time start rule and end rule are optional components.

Timezone component selectors are 'std-name, 'std-offset, 'dst-name, 'dst-offset, 'dst-start, 'dst-end.

`current-timezone` [parameter]

```
(current-timezone [VALUE])
```

The currently defined timezone. The specified `VALUE` is either a timezone string value, or #f, indicating no timezone. When no timezone value is set the default timezone is UTC.

`current-timezone-components` [procedure]

```
(current-timezone-components)
```

Returns the timezone-components object corresponding to the current-timezone.

`timezone-components?` [procedure]

```
(timezone-components? TIMEZONE-COMPONENTS)
```

Is the specified `TIMEZONE-COMPONENTS` object actually a timezone-components object?

`timezone-component-ref` [procedure]

```
(timezone-component-ref TIMEZONE-COMPONENTS SELECTOR [DEFAULT #f])
```

Returns the timezone-component `SELECTOR` of the `TIMEZONE-COMPONENTS` object, or the `DEFAULT` for a missing component.

`set-timezone-component!` [procedure]

```
(set-timezone-component! TIMEZONE-COMPONENTS SELECTOR VALUE)
```

Sets the timezone-component `SELECTOR` of the `TIMEZONE-COMPONENTS` object to `VALUE`.

`timezone-dst-rule-julian-noleap?` [procedure]

```
(timezone-dst-rule-julian-noleap? TIMEZONE-RULE)
```

Is the specified `TIMEZONE-RULE` object actually a daylight saving time julian day without leap seconds object?

`timezone-dst-rule-julian-leap?` [procedure]
`(timezone-dst-rule-julian-leap? TIMEZONE-RULE)`

Is the specified `TIMEZONE-RULE` object actually a daylight saving time julian day assuming leap seconds object?

`timezone-dst-rule-mwd?` [procedure]
`(timezone-dst-rule-mwd? TIMEZONE-RULE)`

Is the specified `TIMEZONE-RULE` object actually a daylight saving time month.week.day object?

`timezone-dst-rule-offset` [procedure]
`(timezone-dst-rule-offset TIMEZONE-RULE)`

Returns the seconds within day offset component of the specified `TIMEZONE-RULE` object.

`timezone-dst-rule-julian` [procedure]
`(timezone-dst-rule-julian TIMEZONE-RULE)`

Returns the julian day component of the specified `TIMEZONE-RULE` object.

`timezone-dst-rule-month` [procedure]
`(timezone-dst-rule-month TIMEZONE-RULE)`

Returns the month of year component of the specified `TIMEZONE-RULE` object.

`timezone-dst-rule-week` [procedure]
`(timezone-dst-rule-week TIMEZONE-RULE)`

Returns the week of month component of the specified `TIMEZONE-RULE` object.

`timezone-dst-rule-day` [procedure]
`(timezone-dst-rule-day TIMEZONE-RULE)`

Returns the day of week component of the specified `TIMEZONE-RULE` object.

`make-timezone-dst-rule-julian-leap` [procedure]
`(make-timezone-dst-rule-julian-leap JULIAN-DAY OFFSET)`

Returns a daylight saving time julian day assuming leap seconds rule object.

`make-timezone-dst-rule-julian-noleap` [procedure]
`(make-timezone-dst-rule-julian-noleap JULIAN-DAY OFFSET)`

Returns a daylight saving time julian day without leap seconds rule object.

`make-timezone-dst-rule-mwd` [procedure]
`(make-timezone-dst-rule-mwd MONTH WEEK DAY OFFSET)`

Returns a daylight saving time month.week.day rule object.

`posix-timezone-value->timezone-components` [procedure]
`(posix-timezone-value->timezone-components STRING [SOURCE "POSIX"])`

Parses a POSIX timezone string specification, `STRING`, and returns the corresponding timezone-components object, or #f when a parse error occurs. A #f or empty string value is mapped to the default timezone. The optional `SOURCE` indicates what locale system supplied the string.

`posix-load-timezone` [procedure]

```
(posix-load-timezone)
```

Initialize the current-timezone from the TZ environment variable.

2.2 Locale

Access to locale information. A locale object is composed of a Language, an optional Script, an optional Region, an optional Codeset, and an optional Modifier. The language should be an ISO 639-1 or ISO 639-2 name. The Script should be a RFC 3066bis name. The region should be an ISO 3166-1 name. The codeset and modifier forms are locale dependent.

Locale component selectors are 'language, 'script, 'region, 'codeset, and 'modifier.

`current-locale` [parameter]

```
(current-locale [VALUE])
```

The currently defined locale. The specified `VALUE` is either a locale string value, or #f, indicating locale independence. When no locale value is set the default locale is #f.

`current-locale-components` [procedure]

```
(current-locale-components)
```

Returns the locale-components object corresponding to the current-locale.

`locale-components?` [procedure]

```
(locale-components? LOCALE-COMPONENTS)
```

Is the specified `LOCALE-COMPONENTS` object actually a locale-components object?

`locale-component-ref` [procedure]

```
(locale-component-ref LOCALE-COMPONENTS SELECTOR [DEFAULT #f])
```

Returns the locale-component `SELECTOR` of the `LOCALE-COMPONENTS` object, or the `DEFAULT` for a missing component.

`set-locale-component!` [procedure]

```
(set-locale-component! LOCALE-COMPONENTS SELECTOR VALUE)
```

Sets the locale-component `SELECTOR` of the `LOCALE-COMPONENTS` object to `VALUE`.

`posix-locale-value->locale-components` [procedure]

```
(posix-locale-value->locale-components STRING [SOURCE "POSIX"])
```

Parses a POSIX locale string specification, `STRING`, and returns the corresponding locale-components object, or #f when a parse error occurs. A #f or empty string value is mapped to the default locale. The optional `SOURCE` indicates what locale system supplied the string.

`posix-load-locale` [procedure]

```
(posix-load-locale)
```

Initialize the current-locale from the LC_* or LANG environment variables. When both the LC_ALL and LANG environment variables are not set the current-locale is #f, even though some locale-catagories may have values. LC_ALL or LANG should be set if any locale catagories are set.

2.3 Locale Category

Access to the locale information by category.

The locale category selectors are 'COLLATE, 'CTYPE, 'MESSAGES, 'MONETARY, 'NUMERIC, and 'TIME.

`set-locale-category!` [procedure]

```
(set-locale-category! CATEGORY LOCALE-COMPONENTS)
```

Sets the specified `CATEGORY` to the specified `LOCALE-COMPONENTS` object.

`locale-category-ref` [procedure]

```
(locale-category-ref CATEGORY)
```

Returns the specified `CATEGORY` locale-components object, or #f if the category is not valued.

3 License

Index

C

current-locale 4
current-locale-components 4
current-timezone 2
current-timezone-components 2

L

locale-category-ref 5
locale-component-ref 4
locale-components? 4

M

make-timezone-dst-rule-julian-leap 3
make-timezone-dst-rule-julian-noleap 3
make-timezone-dst-rule-mwd 3

P

posix-load-locale 4
posix-load-timezone 4
posix-locale-value->locale-components 4
posix-timezone-value->timezone-components 3

S

set-locale-category! 5
set-locale-component! 4
set-timezone-component! 2

T

timezone-component-ref 2
timezone-components? 2
timezone-dst-rule-day 3
timezone-dst-rule-julian 3
timezone-dst-rule-julian-leap? 3
timezone-dst-rule-julian-noleap? 2
timezone-dst-rule-month 3
timezone-dst-rule-mwd? 3
timezone-dst-rule-offset 3
timezone-dst-rule-week 3