r6rs-libraries egg

Andre van Tonders macro and module system, with R6RS (draft) / SRFI-83 compliant libraries
Extension for Chicken Scheme
Version 2.204

Andre van Tonder, some work on the CHICKEN port by felix

Table of Contents

1 About this egg

1.1 Version history

`2.204` Automatically compiles code using this extension with literal-compression

`2.203` Added support for SRFI-61

`2.202` Added a few new procedures, removed dependency on `chicken-config`.

`2.201` Initial release

1.2 Usage

Load this egg like so:

```
(require-extension r6rs-libraries)
```

2 Documentation

For detailed specifications consult the official SRFI documents:

SRFI-72

SRFI-83

Portable modules

2.1 Notes

At least CHICKEN version 2.110 is required.

"Curried" `define` is allowed, as are definitions of the form `(define VARIABLE)` Note that DSSSL extended lambda lists are currently not available. CHICKEN as of version 2.0 supports the abbreviation `#‘` for `quasisyntax`

`import` loads separately compiled libraries automatically. The predefined modules are handled specially since they are bundled in a separate shared object. Other modules are loaded via `require` by prepending the URI-scheme to the library name (if existing) and converting the library name to a symbol. So for example `(import "scheme://syntax-case")` would be required as `scheme-syntax-case`.

Toplevel expressions are by default evaluated in a scope that has the `scheme://chicken` library imported.

If you need to create a module for pre-existing libraries, create a wrapper module. The special form `import-primitives` allows importing unqualified toplevel identifiers into the current scope.

This extension is still in an experimental stadium.

2.2 Languages

Basically any library may be used as the "language" part (the second argument) of a library definition, but it must be available both at compile and run-time. Pre-defined languages are:

`scheme://chicken` [library]

Contains all R5RS and non-standard procedures provided by the basic CHICKEN system, that is everything contained in the `library` and `eval` library units.

`scheme://r6rs` [library]

R5RS procedures and syntax, including `scheme://syntax-case` and SRFI-83.

2.3 Other libraries

`chicken://macros` [library]

Non-standard syntax extensions. Note that `eval-when` is not available and internal `define-values` is not supported)

`chicken://ffi` [library]

Macros for interfacing to foreign code.

`chicken://extras` [library]

`chicken://lolevel` [library]

`chicken://posix` [library]

`chicken://regex` [library]

`chicken://tinyclos` [library]

`chicken://utils` [library]

Non-standard extensions of the CHICKEN system.

The `chicken://tinyclos` module exports the `define-class`, `define-generic` and `define-method` syntax. Methods may only be defined on generic functions that have been previously defined with `define-generic`.

`scheme://syntax-case` [library]

`scheme://srfi-1` [library]

`scheme://srfi-4` [library]

`scheme://srfi-13` [library]

`scheme://srfi-14` [library]

`scheme://srfi-18` [library]

Modules with SRFI functionality.

`scheme://syntax-case` provides the `syntax-case` macro and also defines`syntax-rules`.

2.4 Extensions

`define-for-syntax` [macro]

```
(define-for-syntax ...)
```

Identical to `(begin-for-syntax (define ...))`. Exported by the `scheme://chicken` and `scheme://r6rs` libraries.

`import-primitives` [macro]

```
(import-primitives IDENTIFIER ...)
```

Imports the given identifiers unadorned into the current library. This provides a kind of"backdoor" for accessing internal identifiers or identifiers that are not defined in a library.

3 License

Index