

bigloo-csv

Joseph Donaldson

1 Overview

The bigloo-csv library supports the parsing of csv and csv-like data. By default, it enables the parsing of comma, tab, and pipe separated data. In addition, facilities are provided that enable extending the library to support additional csv-like formats.

The rest of this document describes the bigloo-csv application programming interface.

2 API Reference

read-csv-record *input-port* [*custom-lexer*] [Function]

read-csv-record has one required argument, the input-port of the csv data to parse, and an optional argument indicating the lexer to use, by default the lexer supporting standard csv files. It returns a single record, as a list, or **#eof-object**. Upon error, it will throw an **&invalid-port-error** or **&io-parse-error** exception.

read-csv-records *input-port* [*custom-lexer*] [Function]

read-csv-records has one required argument, the input-port of the csv data to parse, and an optional argument indicating the lexer to use, by default the lexer supporting standard csv files. It returns all of the records, as a list of lists, or **#eof-object**. Upon error, it will throw an **&invalid-port-error** or **&io-parse-error** exception.

csv-for-each *proc input-port* [*custom-lexer*] [Function]

csv-for-each has two required arguments, a procedure to apply to each record and the input-port of the csv data to parse, and an optional argument indicating the lexer to use, by default the lexer supporting standard csv files. It returns **#unspecified**. Upon error, it will throw an **&invalid-port-error** or **&io-parse-error** exception.

csv-map *proc input-port* [*custom-lexer*] [Function]

csv-map has two required arguments, a procedure to apply to each record and the input-port of the csv data to parse, and an optional argument indicating the lexer to use, by default the lexer supporting standard csv files. It returns the results of applying **proc** to each record as a list. Upon error, it will throw an **&invalid-port-error** or **&io-parse-error** exception.

make-csv-lexer *sep quot* [Macro]

make-csv-lexer has two required arguments, a character used to separate records and a character for quoting. It returns custom lexer.

+csv-lexer+ [Variable]

+csv-lexer+ is a bigloo-csv lexer supporting the standard comma-separated value format.

+tsv-lexer+ [Variable]

+tsv-lexer+ is a bigloo-csv lexer supporting the tab-separated value format.

+psv-lexer+ [Variable]

+psv-lexer+ is a bigloo-csv lexer supporting the pipe-separated value format.

The following is a simple example of using the bigloo-csv library. It parses a single record from the given csv data and prints it.

```
(module example
  (library bigloo-csv)
  (main main))

(define +csv-data+ "dog,cat,horse\npig,cow,squirrel")

(define (main args)
  (let ((in (open-input-string +csv-data+)))
    (unwind-protect
      (print (read-csv-record in))
      (close-input-port in))))
```

Index

+

+csv-lexer+.....	2
+psv-lexer+.....	2
+tsv-lexer+.....	2

C

csv-for-each	2
csv-map	2

M

make-csv-lexer	2
----------------------	---

R

read-csv-record	2
read-csv-records	2

Table of Contents

1	Overview	1
2	API Reference	2
	Index	4