

# The Data Frame Input/Output Reference Manual

---

Common Lisp library for reading data from text files (eg CSV), version 1.3

Steve Nunez <steve@symbolics.tech>  
Tamas Papp <tkpapp@gmail.com>

---

# Table of Contents

<b>1</b>	<b>Systems .....</b>	<b>1</b>
1.1	dfio.....	1
<b>2</b>	<b>Files .....</b>	<b>3</b>
2.1	Lisp.....	3
2.1.1	dfio.asd.....	3
2.1.2	dfio/decimal.lisp.....	3
2.1.3	dfio/string-table.lisp.....	3
2.1.4	dfio/data-column.lisp.....	4
2.1.5	dfio/dfio.lisp.....	4
<b>3</b>	<b>Packages .....</b>	<b>5</b>
3.1	dfio.decimal.....	5
3.2	dfio.string-table.....	5
3.3	dfio.data-column.....	6
3.4	dfio.....	6
<b>4</b>	<b>Definitions .....</b>	<b>7</b>
4.1	Exported definitions.....	7
4.1.1	Functions.....	7
4.1.2	Conditions.....	10
4.1.3	Structures.....	10
4.1.4	Classes.....	11
4.2	Internal definitions.....	12
4.2.1	Special variables.....	12
4.2.2	Functions.....	12
4.2.3	Types.....	13
<b>Appendix A</b>	<b>Indexes .....</b>	<b>15</b>
A.1	Concepts.....	15
A.2	Functions.....	16
A.3	Variables.....	17
A.4	Data types.....	18



# 1 Systems

The main system appears first, followed by any subsystem dependency.

## 1.1 dfio

### Maintainer

Steve Nunez <steve@symbolics.tech>

### Author

Tamas Papp <tkpapp@gmail.com>

### License

MS-PL

### Description

Common Lisp library for reading data from text files (eg CSV).

### Version

1.3

### Dependencies

- alexandria
- anaphora
- cl-csv
- data-frame
- let-plus

### Source

[dfio.asd], page 3, (file)

### Directory

s:/src/dfio/

### Components

- [decimal.lisp], page 3, (file)
- [string-table.lisp], page 3, (file)
- [data-column.lisp], page 4, (file)
- [dfio.lisp], page 4, (file)



## 2 Files

Files are sorted by type and then listed depth-first from the systems components trees.

### 2.1 Lisp

#### 2.1.1 dfio.asd

**Location** dfio.asd

**Systems** [dfio], page 1, (system)

#### 2.1.2 dfio/decimal.lisp

**Parent** [dfio], page 1, (system)

**Location** decimal.lisp

**Packages** [dfio.decimal], page 5,

##### Exported Definitions

- [parse-rational], page 8, (function)
- [parse-rational-error], page 10, (condition)
- [parse-real], page 8, (function)

##### Internal Definitions

- [+exponent-chars+], page 12, (special variable)
- [gobble-positive-integer], page 12, (function)
- [gobble-sign], page 12, (function)

#### 2.1.3 dfio/string-table.lisp

##### Dependency

[decimal.lisp], page 3, (file)

**Parent** [dfio], page 1, (system)

**Location** string-table.lisp

**Packages** [dfio.string-table], page 5,

##### Exported Definitions

- [string-table], page 9, (function)
- [string-table], page 10, (structure)
- [string-table-add], page 9, (function)
- [string-table-count], page 9, (function)
- [string-table-duplicate], page 10, (condition)
- [string-table-intern], page 9, (function)
- [string-table-lookup], page 9, (function)
- [string-table-not-found], page 10, (condition)
- [string-table-strings], page 9, (function)

##### Internal Definitions

- [copy-string-table], page 12, (function)
- [string-table-get], page 12, (function)
- [(setf string-table-get)], page 13, (function)

- [string-table-p], page 13, (function)
- [string-table-table], page 13, (function)
- [(setf string-table-table)], page 13, (function)

#### 2.1.4 dfio/data-column.lisp

**Dependency**

[string-table.lisp], page 3, (file)

**Parent**

[dfio], page 1, (system)

**Location**

data-column.lisp

**Packages**

[dfio.data-column], page 6,

**Exported Definitions**

- [data-column], page 7, (function)
- [data-column], page 11, (class)
- [data-column-add], page 7, (function)
- [data-column-counts], page 7, (function)
- [data-column-vector], page 7, (function)

**Internal Definitions**

[non-negative-integer], page 13, (type)

#### 2.1.5 dfio/dfio.lisp

**Dependency**

[data-column.lisp], page 4, (file)

**Parent**

[dfio], page 1, (system)

**Location**

dfio.lisp

**Packages**

[dfio], page 6,

**Exported Definitions**

- [csv-to-data-frame], page 7, (function)
- [data-frame-to-csv], page 7, (function)
- [string-to-keyword], page 9, (function)

**Internal Definitions**

- [2d-array-to-list], page 12, (function)
- [csv-to-data-columns], page 12, (function)

## 3 Packages

Packages are listed by definition order.

### 3.1 dfio.decimal

**Source** [decimal.lisp], page 3, (file)

**Use List**

- let-plus
- anaphora
- common-lisp

**Used By List**

[dfio.data-column], page 6,

**Exported Definitions**

- [parse-rational], page 8, (function)
- [parse-rational-error], page 10, (condition)
- [parse-real], page 8, (function)

**Internal Definitions**

- [+exponent-chars+], page 12, (special variable)
- [gobble-positive-integer], page 12, (function)
- [gobble-sign], page 12, (function)

### 3.2 dfio.string-table

**Source** [string-table.lisp], page 3, (file)

**Use List**

- let-plus
- anaphora
- alexandria
- common-lisp

**Used By List**

[dfio.data-column], page 6,

**Exported Definitions**

- [string-table], page 9, (function)
- [string-table], page 10, (structure)
- [string-table-add], page 9, (function)
- [string-table-count], page 9, (function)
- [string-table-duplicate], page 10, (condition)
- [string-table-intern], page 9, (function)
- [string-table-lookup], page 9, (function)
- [string-table-not-found], page 10, (condition)
- [string-table-strings], page 9, (function)

**Internal Definitions**

- [copy-string-table], page 12, (function)



- [string-table-get], page 12, (function)
- [(setf string-table-get)], page 13, (function)
- [string-table-p], page 13, (function)
- [string-table-table], page 13, (function)
- [(setf string-table-table)], page 13, (function)

### 3.3 dfio.data-column

**Source** [data-column.lisp], page 4, (file)

**Use List**

- let-plus
- [dfio.string-table], page 5,
- [dfio.decimal], page 5,
- anaphora
- common-lisp

**Used By List**

- vgl<sub>t</sub>
- [dfio], page 6,

**Exported Definitions**

- [data-column], page 7, (function)
- [data-column], page 11, (class)
- [data-column-add], page 7, (function)
- [data-column-counts], page 7, (function)
- [data-column-vector], page 7, (function)

**Internal Definitions**

[non-negative-integer], page 13, (type)

### 3.4 dfio

**Source** [dfio.lisp], page 4, (file)

**Use List**

- [dfio.data-column], page 6,
- let-plus
- cl-csv
- anaphora
- alexandria
- common-lisp

**Used By List**

vgl<sub>t</sub>

**Exported Definitions**

- [csv-to-data-frame], page 7, (function)
- [data-frame-to-csv], page 7, (function)
- [string-to-keyword], page 9, (function)

**Internal Definitions**

- [2d-array-to-list], page 12, (function)
- [csv-to-data-columns], page 12, (function)

## 4 Definitions

Definitions are sorted by export status, category, package, and then by lexicographic order.

### 4.1 Exported definitions

#### 4.1.1 Functions

**csv-to-data-frame** *STREAM-OR-STRING &key SKIP-FIRST-ROW?* [Function]  
*COLUMN-KEYS-OR-FUNCTION*

Read a CSV file (or stream, or string) into a DATA-FRAME, which is returned.

When SKIP-FIRST-ROW?, the first row is read separately and COLUMN-KEYS-OR-FUNCTION is used to form column keys.

When COLUMN-KEYS-OR-FUNCTION is a sequence, it is used for column keys, regardless of the value of SKIP-FIRST-ROW?.

**Package** [dfio], page 6,

**Source** [dfio.lisp], page 4, (file)

**data-column** *&key MAP-ALIST DEFAULT-FLOAT-FORMAT* [Function]

**Package** [dfio.data-column], page 6,

**Source** [data-column.lisp], page 4, (file)

**data-column-add** *DATA-COLUMN STRING* [Function]

**Package** [dfio.data-column], page 6,

**Source** [data-column.lisp], page 4, (file)

**data-column-counts** *DATA-COLUMN* [Function]

Return the counts.

**Package** [dfio.data-column], page 6,

**Source** [data-column.lisp], page 4, (file)

**data-column-vector** *DATA-COLUMN* [Function]

Return the collected elements as a vector.

**Package** [dfio.data-column], page 6,

**Source** [data-column.lisp], page 4, (file)

**data-frame-to-csv** *DF &key STREAM ADD-FIRST-ROW* [Function]  
*(SEPARATOR \*SEPARATOR\*) (QUOTE \*QUOTE\*) (ESCAPE*  
*\*QUOTE-ESCAPE\*) (NEWLINE \*WRITE-NEWLINE\*)*  
*(ALWAYS-QUOTE \*ALWAYS-QUOTE\*)*

Write a data-frame to a stream.

Keywords:

stream: stream to write to. Default: nil.

nil - writes the rows to a string and returns it  
 an open stream

a pathname (overwrites if the file exists)

quote: quoting character. Defaults to \*quote\*

escape: escaping character. Defaults to `*quote-escape*`  
 newline: newline character. Defaults to `*write-newline*`  
 always-quote: Defaults to `*always-quote*`  
 add-first-row: Add column names as the first

Notes:

The `:newline` keyword requires a sequence, so use `:newline '(#newline)` or use `cl-interpol`

**Package** [dfio], page 6,

**Source** [dfio.lisp], page 4, (file)

**parse-rational** *STRING &key START END EXPONENT-CHARS* [Function]  
 Parse a decimal rational in (subseq string start end) of the form  
 [sign][whole][.fraction]][exponent] where

sign ::= + | - | empty  
 whole ::= digit\*  
 fraction ::= digit\*  
 exponent ::= exponent-char[sign]digit+

with the restriction that `WHOLE` and `FRACTION` cannot be empty at the same time.  
`EXPONENT-CHAR` is a string and contains the valid exponent chars.

Whitespace is NOT trimmed, and leads to an error. In case of a parsing failure, `PARSE-RATIONAL-ERROR` is used.

Return (values `NUMBER` `DECIMAL-DOT?` `EXPONENT-CHAR`). `NUMBER` is a `RATIONAL`, `DECIMAL-DOT?` is `T` when a decimal dot is present, otherwise `NIL`, `EXPONENT-CHAR` contains the exponent character, `NIL` if not present.

Numbers of the form `.112` and `112.` are valid syntax, representing `0.112` and `112.0`, respectively.

Examples:

```
(parse-rational "7") => (values 7 NIL NIL)
(parse-rational "7.") => (values 7 T NIL)
(parse-rational "0.7") => (values 7/10 T NIL)
(parse-rational ".7") => (values 7/10 T NIL)
(parse-rational "7.e2") => (values 700 T #e)
(parse-rational ".7d1") => (values 7 T #d)
```

**Package** [dfio.decimal], page 5,

**Source** [decimal.lisp], page 3, (file)

**parse-real** *STRING &key START END S-FLOAT F-FLOAT D-FLOAT L-FLOAT E-FLOAT* [Function]

Wrapper for `PARSE-RATIONAL`, converting non-integers to floats. The float type is determined by the `-float` arguments for each exponent character. Integers are not converted to floats. Return a single value, type of (or integer float).

See `PARSE-RATIONAL` for accepted formats, errors, etc.

**Package** [dfio.decimal], page 5,

- Source** [decimal.lisp], page 3, (file)
- string-table &key (TABLE TABLE)** [Function]
- Package** [dfio.string-table], page 5,
- Source** [string-table.lisp], page 3, (file)
- string-table-add STRING-TABLE STRING &optional VALUE** [Function]
- Add STRING mapped to VALUE to STRING-TABLE, raising STRING-TABLE-DUPLICATE if STRING is already in the table. Return VALUE.
- Package** [dfio.string-table], page 5,
- Source** [string-table.lisp], page 3, (file)
- string-table-count STRING-TABLE** [Function]
- Number of distinct strings in the table.
- Package** [dfio.string-table], page 5,
- Source** [string-table.lisp], page 3, (file)
- string-table-intern STRING-TABLE STRING &optional NEW-VALUE** [Function]
- If STRING is already in STRING-TABLE, return its value, otherwise add it and return NEW-VALUE. When used with the default argument for NEW-VALUE, EQUAL strings are always mapped to values that are EQ.
- Package** [dfio.string-table], page 5,
- Source** [string-table.lisp], page 3, (file)
- string-table-lookup STRING-TABLE STRING** [Function]
- Return the value corresponding to STRING in STRING-TABLE, or raise the STRING-TABLE-NOT-FOUND error.
- Package** [dfio.string-table], page 5,
- Source** [string-table.lisp], page 3, (file)
- string-table-strings STRING-TABLE** [Function]
- List of strings in STRING-TABLE.
- Package** [dfio.string-table], page 5,
- Source** [string-table.lisp], page 3, (file)
- string-to-keyword STRING** [Function]
- Map string to a keyword.
- This is the default for constructing column keys for CSV files.
- The current implementation replaces #. and #space with a #-, and upcases all other characters.
- Package** [dfio], page 6,
- Source** [dfio.lisp], page 4, (file)

### 4.1.2 Conditions

`parse-rational-error ()` [Condition]

Error used by `parse-rational` and `parse-real`.

**Package** [dfio.decimal], page 5,

**Source** [decimal.lisp], page 3, (file)

**Direct superclasses**  
error (condition)

**Direct slots**

**string** [Slot]

**Initform** (quote :string)

**message** [Slot]

**Initform** (quote :message)

`string-table-duplicate ()` [Condition]

String is already in the table.

**Package** [dfio.string-table], page 5,

**Source** [string-table.lisp], page 3, (file)

**Direct superclasses**  
error (condition)

`string-table-not-found ()` [Condition]

String not found in table.

**Package** [dfio.string-table], page 5,

**Source** [string-table.lisp], page 3, (file)

**Direct superclasses**  
error (condition)

### 4.1.3 Structures

`string-table ()` [Structure]

A table of distinct strings, optionally mapping each one to a value.

**Package** [dfio.string-table], page 5,

**Source** [string-table.lisp], page 3, (file)

**Direct superclasses**  
structure-object (structure)

**Direct methods**  
print-object (method)

**Direct slots**

**table** [Slot]

**Type** hash-table

**Initform** (make-hash-table :test (function equalp))

**Readers** [string-table-table], page 13, (function)

**Writers** [(setf string-table-table)], page 13, (function)

### 4.1.4 Classes

`data-column ()` [Class]

**Package** [dfio.data-column], page 6,

**Source** [data-column.lisp], page 4, (file)

**Direct superclasses**

standard-object (class)

**Direct slots**

`reverse-elements` [Slot]

**Type** list

`default-float-format` [Slot]

**Type** symbol

**Initargs** :default-float-format

`float-count` [Slot]

**Type** dfio.data-column::non-negative-integer

**Initform** 0

`integer-count` [Slot]

**Type** dfio.data-column::non-negative-integer

**Initform** 0

`integer-min` [Slot]

**Type** integer

**Initform** 0

`integer-max` [Slot]

**Type** integer

**Initform** 0

`map-count` [Slot]

**Type** dfio.data-column::non-negative-integer

**Initform** 0

`map-table` [Slot]

**Type** dfio.string-table:string-table

**Initargs** :map-table

`string-count` [Slot]

**Type** dfio.data-column::non-negative-integer

**Initform** 0

`string-table` [Slot]

**Type** dfio.string-table:string-table

**Initform** (dfio.string-table:string-table)

## 4.2 Internal definitions

### 4.2.1 Special variables

**+exponent-chars+** [Special Variable]  
 Default exponent characters.  
**Package** [dfio.decimal], page 5,  
**Source** [decimal.lisp], page 3, (file)

### 4.2.2 Functions

**2d-array-to-list** *ARRAY* [Function]  
 Helper for CSV writing.  
**Package** [dfio], page 6,  
**Source** [dfio.lisp], page 4, (file)

**copy-string-table** *INSTANCE* [Function]  
**Package** [dfio.string-table], page 5,  
**Source** [string-table.lisp], page 3, (file)

**csv-to-data-columns** *STREAM-OR-STRING SKIP-FIRST-ROW?* [Function]  
 Read a CSV file (or stream, or string), accumulate the values in DATA-COLUMNS, return a list of these. Rows are checked to have the same number of elements.  
 When SKIP-FIRST-ROW?, the first row is read separately and returned as the second value (list of strings), otherwise it is considered data like all other rows.  
**Package** [dfio], page 6,  
**Source** [dfio.lisp], page 4, (file)

**gobble-positive-integer** *STRING START END* [Function]  
 If (SUBSEQ STRING START END) starts with a nonnegative integer (ie a sequence of digits 0-9), return the integer and position at which it ends as two values.  
 Otherwise, return NIL and 0.

START < END has to hold, END cannot be NIL. Consequences are undefined when START >= END.

**Package** [dfio.decimal], page 5,  
**Source** [decimal.lisp], page 3, (file)

**gobble-sign** *STRING START* [Function]  
 Return (values SIGNUM INDEX), where SIGNUM is -1 or 1 depending on whether (CHAR STRING START) was a sign, and INDEX is the index of the subsequent character (START or START+1).

**Package** [dfio.decimal], page 5,  
**Source** [decimal.lisp], page 3, (file)

**string-table-get** *STRING-TABLE STRING* [Function]  
 Synonym for GETHASH, used internally.  
**Package** [dfio.string-table], page 5,

**Source** [string-table.lisp], page 3, (file)

**Writer** [(setf string-table-get)], page 13, (function)

(setf string-table-get) *VALUE STRING-TABLE STRING* [Function]  
 Synonym for (SETF GETHASH), used internally, checks that STRING is a string.

**Package** [dfio.string-table], page 5,

**Source** [string-table.lisp], page 3, (file)

**Reader** [string-table-get], page 12, (function)

string-table-p *OBJECT* [Function]

**Package** [dfio.string-table], page 5,

**Source** [string-table.lisp], page 3, (file)

string-table-table *INSTANCE* [Function]

(setf string-table-table) *VALUE INSTANCE* [Function]

**Package** [dfio.string-table], page 5,

**Source** [string-table.lisp], page 3, (file)

### 4.2.3 Types

non-negative-integer () [Type]

**Package** [dfio.data-column], page 6,

**Source** [data-column.lisp], page 4, (file)





## Appendix A Indexes

### A.1 Concepts

#### D

dfio.asd.....	3
dfio/data-column.lisp.....	4
dfio/decimal.lisp.....	3
dfio/dfio.lisp.....	4
dfio/string-table.lisp.....	3

#### F

File, Lisp, dfio.asd.....	3
File, Lisp, dfio/data-column.lisp.....	4
File, Lisp, dfio/decimal.lisp.....	3
File, Lisp, dfio/dfio.lisp.....	4

File, Lisp, dfio/string-table.lisp.....	3
---	---

#### L

Lisp File, dfio.asd.....	3
Lisp File, dfio/data-column.lisp.....	4
Lisp File, dfio/decimal.lisp.....	3
Lisp File, dfio/dfio.lisp.....	4
Lisp File, dfio/string-table.lisp.....	3

## A.2 Functions

(  
 (setf string-table-get) ..... 13  
 (setf string-table-table) ..... 13

## 2

2d-array-to-list ..... 12

## C

copy-string-table ..... 12  
 csv-to-data-columns ..... 12  
 csv-to-data-frame ..... 7

## D

data-column ..... 7  
 data-column-add ..... 7  
 data-column-counts ..... 7  
 data-column-vector ..... 7  
 data-frame-to-csv ..... 7

## F

Function, (setf string-table-get) ..... 13  
 Function, (setf string-table-table) ..... 13  
 Function, 2d-array-to-list ..... 12  
 Function, copy-string-table ..... 12  
 Function, csv-to-data-columns ..... 12  
 Function, csv-to-data-frame ..... 7  
 Function, data-column ..... 7  
 Function, data-column-add ..... 7  
 Function, data-column-counts ..... 7  
 Function, data-column-vector ..... 7  
 Function, data-frame-to-csv ..... 7  
 Function, gobble-positive-integer ..... 12

Function, gobble-sign ..... 12  
 Function, parse-rational ..... 8  
 Function, parse-real ..... 8  
 Function, string-table ..... 9  
 Function, string-table-add ..... 9  
 Function, string-table-count ..... 9  
 Function, string-table-get ..... 12  
 Function, string-table-intern ..... 9  
 Function, string-table-lookup ..... 9  
 Function, string-table-p ..... 13  
 Function, string-table-strings ..... 9  
 Function, string-table-table ..... 13  
 Function, string-to-keyword ..... 9

## G

gobble-positive-integer ..... 12  
 gobble-sign ..... 12

## P

parse-rational ..... 8  
 parse-real ..... 8

## S

string-table ..... 9  
 string-table-add ..... 9  
 string-table-count ..... 9  
 string-table-get ..... 12  
 string-table-intern ..... 9  
 string-table-lookup ..... 9  
 string-table-p ..... 13  
 string-table-strings ..... 9  
 string-table-table ..... 13  
 string-to-keyword ..... 9

## A.3 Variables

+

+exponent-chars+ ..... 12

D

default-float-format ..... 11

F

float-count ..... 11

I

integer-count ..... 11

integer-max ..... 11

integer-min ..... 11

M

map-count ..... 11

map-table ..... 11

message ..... 10

R

reverse-elements ..... 11

S

Slot, default-float-format ..... 11

Slot, float-count ..... 11

Slot, integer-count ..... 11

Slot, integer-max ..... 11

Slot, integer-min ..... 11

Slot, map-count ..... 11

Slot, map-table ..... 11

Slot, message ..... 10

Slot, reverse-elements ..... 11

Slot, string ..... 10

Slot, string-count ..... 11

Slot, string-table ..... 11

Slot, table ..... 10

Special Variable, +exponent-chars+ ..... 12

string ..... 10

string-count ..... 11

string-table ..... 11

T

table ..... 10

## A.4 Data types

### C

Class, <code>data-column</code> .....	11
Condition, <code>parse-rational-error</code> .....	10
Condition, <code>string-table-duplicate</code> .....	10
Condition, <code>string-table-not-found</code> .....	10

### D

<code>data-column</code> .....	11
<code>dfio</code> .....	1, 6
<code>dfio.data-column</code> .....	6
<code>dfio.decimal</code> .....	5
<code>dfio.string-table</code> .....	5

### N

<code>non-negative-integer</code> .....	13
---	----

### P

Package, <code>dfio</code> .....	6
Package, <code>dfio.data-column</code> .....	6
Package, <code>dfio.decimal</code> .....	5
Package, <code>dfio.string-table</code> .....	5
<code>parse-rational-error</code> .....	10

### S

<code>string-table</code> .....	10
<code>string-table-duplicate</code> .....	10
<code>string-table-not-found</code> .....	10
Structure, <code>string-table</code> .....	10
System, <code>dfio</code> .....	1

### T

Type, <code>non-negative-integer</code> .....	13
---	----