The Data Frame I/O Reference Manual

Data frame I/O, version 1.0.0

Steve Nunez <steve@symbolics.tech>

This manual was generated automatically by Declt 4.0b2.

Table of Contents

Copying

This program is distributed under the terms of the Microsoft Public License.

1 Systems

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

1.1 dfio

Common Lisp library for reading and writing data-frames

Long Name
Data frame I/O

Author Steve Nunez <steve@symbolics.tech>

Source Control
(GIT https://github.com/Lisp-Stat/dfio.git)

Bug Tracker
https://github.com/Lisp-Stat/dfio/issues

License MS-PL

Long Description
DFIO is used to read and write data frames from string, streams or disk. When building a library like this, one question that needs to be answered is "where do you draw the line on formats?". For example, should DFIO read/write JSON? The answer to that question is 'no', and the reason is that there are many JSON formats – they are applicaton specific. Application specific format should be read in an application specific package, and this is the case, for example, with Vega. Formats like CSV, HD5 or Feather are application agnostic and should be included in DFIO.

Version 1.0.0

Dependencies
- alexandria (system).
- anaphora (system).
- data-frame (system).
- dexador (system).
- fare-csv (system).
- let-plus (system).

Source [dfio.asd], page 5.

Child Components
- [pkgdcl.lisp], page 5 (file).
- [decimal.lisp], page 5 (file).
- [string-table.lisp], page 5 (file).
- [data-column.lisp], page 6 (file).
- [utils.lisp], page 6 (file).
- [write.lisp], page 7 (file).
- [delimited-text.lisp], page 7 (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/dfio.asd

Source [dfio.asd], page 5.

Parent Component
[dfio], page 3 (system).

ASDF Systems
[dfio], page 3.

2.1.2 dfio/pkgdcl.lisp

Source [dfio.asd], page 5.

Parent Component
[dfio], page 3 (system).

Packages

- [dfio.decimal], page 9.
- [dfio.string-table], page 9.
- [dfio], page 10.
- [dfio.data-column], page 10.

2.1.3 dfio/decimal.lisp

Dependency
[pkgdcl.lisp], page 5 (file).

Source [dfio.asd], page 5.

Parent Component
[dfio], page 3 (system).

Public Interface

- [parse-rational], page 13 (function).
- [parse-rational-error], page 16 (condition).
- [parse-real], page 14 (function).

Internals

- [+exponent-chars+], page 18 (special variable).
- [gobble-positive-integer], page 19 (function).
- [gobble-sign], page 19 (function).

2.1.4 dfio/string-table.lisp

Dependency
[decimal.lisp], page 5 (file).

Source [dfio.asd], page 5.

Parent Component
[dfio], page 3 (system).

Public Interface

- [print-object], page 16 (method).
- [string-table], page 14 (function).
- [string-table], page 17 (structure).
- [string-table-add], page 14 (function).
- [string-table-count], page 15 (function).
- [string-table-duplicate], page 17 (condition).
- [string-table-intern], page 15 (function).
- [string-table-lookup], page 15 (function).
- [string-table-not-found], page 17 (condition).
- [string-table-strings], page 15 (function).

Internals

- [copy-string-table], page 19 (function).
- [string-table-get], page 20 (function).
- [(setf string-table-get)], page 20 (function).
- [string-table-p], page 20 (function).
- [string-table-table], page 20 (reader).
- [(setf string-table-table)], page 20 (writer).

2.1.5 dfio/data-column.lisp

Dependency

[string-table.lisp], page 5 (file).

Source [dfio.asd], page 5.

Parent Component

[dfio], page 3 (system).

Public Interface

- [data-column], page 13 (function).
- [data-column], page 17 (class).
- [data-column-add], page 13 (function).
- [data-column-counts], page 13 (function).
- [data-column-vector], page 13 (function).

Internals [non-negative-integer], page 20 (type).

2.1.6 dfio/utils.lisp

Dependency

[data-column.lisp], page 6 (file).

Source [dfio.asd], page 5.

Parent Component

[dfio], page 3 (system).

Public Interface

- [string-to-keyword], page 15 (function).
- [string-to-symbol], page 15 (function).
- [symbol-name-to-pathname], page 15 (function).

Internals

- [%in-stream], page 19 (function).
- [%out-stream], page 19 (function).
- [*default-external-format*], page 18 (special variable).
- [str-strm-file], page 20 (type).
- [with-csv-output-stream], page 19 (macro).
- [with-input-stream], page 19 (macro).

2.1.7 dfio/write.lisp

Dependency
[utils.lisp], page 6 (file).

Source [dfio.asd], page 5.

Parent Component
[dfio], page 3 (system).

Public Interface

- [save], page 14 (function).
- [write-df], page 16 (function).
- [write-properties], page 16 (function).

2.1.8 dfio/delimited-text.lisp

Dependency
[write.lisp], page 7 (file).

Source [dfio.asd], page 5.

Parent Component
[dfio], page 3 (system).

Public Interface

- [read-csv], page 14 (function).
- [write-csv], page 15 (function).

Internals [csv-to-data-columns], page 19 (function).

3 Packages

Packages are listed by definition order.

3.1 dfio.decimal

Source [pkgdcl.lisp], page 5.

Use List

- anaphora.
- common-lisp.
- let-plus.

Used By List

[dfio.data-column], page 10.

Public Interface

- [parse-rational], page 13 (function).
- [parse-rational-error], page 16 (condition).
- [parse-real], page 14 (function).

Internals

- [+exponent-chars+], page 18 (special variable).
- [gobble-positive-integer], page 19 (function).
- [gobble-sign], page 19 (function).

3.2 dfio.string-table

Source [pkgdcl.lisp], page 5.

Use List

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

Used By List

[dfio.data-column], page 10.

Public Interface

- [string-table], page 14 (function).
- [string-table], page 17 (structure).
- [string-table], page 18 (slot).
- [string-table-add], page 14 (function).
- [string-table-count], page 15 (function).
- [string-table-duplicate], page 17 (condition).
- [string-table-intern], page 15 (function).
- [string-table-lookup], page 15 (function).
- [string-table-not-found], page 17 (condition).
- [string-table-strings], page 15 (function).

Internals

- [copy-string-table], page 19 (function).
- [string-table-get], page 20 (function).
- [(setf string-table-get)], page 20 (function).
- [string-table-p], page 20 (function).
- [string-table-table], page 20 (reader).
- [(setf string-table-table)], page 20 (writer).

3.3 dfio

Source [pkgdcl.lisp], page 5.

Use List

- alexandria.
- anaphora.
- common-lisp.
- data-frame.
- [dfio.data-column], page 10.
- let-plus.

Used By List

lisp-stat.

Public Interface

- [read-csv], page 14 (function).
- [save], page 14 (function).
- [string-to-keyword], page 15 (function).
- [string-to-symbol], page 15 (function).
- [symbol-name-to-pathname], page 15 (function).
- [write-csv], page 15 (function).
- [write-df], page 16 (function).
- [write-properties], page 16 (function).

Internals

- [%in-stream], page 19 (function).
- [%out-stream], page 19 (function).
- [*default-external-format*], page 18 (special variable).
- [csv-to-data-columns], page 19 (function).
- [str-strm-file], page 20 (type).
- [with-csv-output-stream], page 19 (macro).
- [with-input-stream], page 19 (macro).

3.4 dfio.data-column

Source [pkgdcl.lisp], page 5.

Use List

- anaphora.
- common-lisp.

- [dfio.decimal], page 9.
- [dfio.string-table], page 9.
- let-plus.

Used By List
[dfio], page 10.

Public Interface
- [data-column], page 13 (function).
- [data-column], page 17 (class).
- [data-column-add], page 13 (function).
- [data-column-counts], page 13 (function).
- [data-column-vector], page 13 (function).

Internals [non-negative-integer], page 20 (type).

4 Definitions

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

4.1 Public Interface

4.1.1 Ordinary functions

`data-column` (**&key** *map-alist default-float-format*) [Function]

Package [`dfio.data-column`], page 10.

Source [`data-column.lisp`], page 6.

`data-column-add` (*data-column string*) [Function]

Package [`dfio.data-column`], page 10.

Source [`data-column.lisp`], page 6.

`data-column-counts` (*data-column*) [Function]

Return the counts.

Package [`dfio.data-column`], page 10.

Source [`data-column.lisp`], page 6.

`data-column-vector` (*data-column*) [Function]

Return the collected elements as a vector.

Package [`dfio.data-column`], page 10.

Source [`data-column.lisp`], page 6.

`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 9.

Source [decimal.lisp], page 5.

`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 9.

Source [decimal.lisp], page 5.

`read-csv` (*source* **&key** *skip-first-row? column-keys-or-function package map-alist*) [Function]
Read a CSV file, stream, string or URL 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 indicates the package to intern column names into.

MAP-ALIST maps values during the import. This is useful if you want special mappings for missing, though the mechanism is general. Returns two values, the data-frame and the source

Package [dfio], page 10.

Source [delimited-text.lisp], page 7.

`save` (*data-symbol pathspec* **&optional** *suffix*) [Function]
Save DF in the file named by PATHSPEC. By default, a suffix of .lisp is added.

Package [dfio], page 10.

Source [write.lisp], page 7.

`string-table` (**&key** *table*) [Function]

Package [dfio.string-table], page 9.

Source [string-table.lisp], page 5.

`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 9.

Source [string-table.lisp], page 5.

`string-table-count` (*string-table*) [Function]
Number of distinct strings in the table.

Package [`dfio.string-table`], page 9.

Source [`string-table.lisp`], page 5.

`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 9.

Source [`string-table.lisp`], page 5.

`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 9.

Source [`string-table.lisp`], page 5.

`string-table-strings` (*string-table*) [Function]
List of strings in STRING-TABLE.

Package [`dfio.string-table`], page 9.

Source [`string-table.lisp`], page 5.

`string-to-keyword` (*string*) [Function]
Map string to a keyword.

The current implementation replaces #. and #space with a #-, and upcases all other characters.

Package [`dfio`], page 10.

Source [`utils.lisp`], page 6.

`string-to-symbol` (*string*) [Function]
Map STRING to a symbol in PACKAGE, replacing #., #- and #space with a #-, and upcasing all other characters. Exports symbol.

Package [`dfio`], page 10.

Source [`utils.lisp`], page 6.

`symbol-name-to-pathname` (*string*) [Function]
Map the symbol-name of S to something that can be part of a logical-pathname

Package [`dfio`], page 10.

Source [`utils.lisp`], page 6.

`write-csv` (*df stream* **&key** *add-first-row separator quote eol*) [Function]
Write DF to STRING-OR-STREAM in CSV format. STRING-OR-STREAM can be a STREAM, a STRING or a file PATHSPEC.

Keywords:
string-or-stream: stream to write to. Default: nil, returning a string

add-first-row: add column names as the first row
separator: separator to use when reading or writing CSV files. A character. By default, a comma: #,
quote: quote character to use when reading or writing CSV files. A character. By default, a double-quote: #" eol: line ending to use when writing CSV files. A string. By default, +CRLF+ as specified by creativyst.

Notes:
The :newline keyword requires a sequence, so use :newline '(#newline)

Package [dfio], page 10.

Source [delimited-text.lisp], page 7.

`write-df` (*data-symbol* **&optional** *stream*) [Function]
Write DATA-SYMBOL to STREAM in a format suitable for reading back in with the Lisp reader

Package [dfio], page 10.

Source [write.lisp], page 7.

`write-properties` (*data-symbol property* **&optional** *stream*) [Function]
Write the variable PROPERTY strings to stream so they can be read back in when LOADed from a lisp file. By convention, the name of the function that sets the property is the same as the property. Example (write-property mtcars :label)

Package [dfio], page 10.

Source [write.lisp], page 7.

4.1.2 Standalone methods

`print-object` ((*string-table* [`string-table`], *page 17*) *stream*) [Method]

Source [string-table.lisp], page 5.

4.1.3 Conditions

`parse-rational-error` [Condition]
Error used by parse-rational and parse-real.

Package [dfio.decimal], page 9.

Source [decimal.lisp], page 5.

Direct superclasses
`error`.

Direct slots

`string` [Slot]
Package `common-lisp`.
Initform `(quote nil)`
Initargs `:string`

`message` [Slot]
Initform `(quote nil)`
Initargs `:message`

`string-table-duplicate` [Condition]
String is already in the table.

Package [`dfio.string-table`], page 9.

Source [`string-table.lisp`], page 5.

Direct superclasses
`error`.

`string-table-not-found` [Condition]
String not found in table.

Package [`dfio.string-table`], page 9.

Source [`string-table.lisp`], page 5.

Direct superclasses
`error`.

4.1.4 Structures

`string-table` [Structure]
A table of distinct strings, optionally mapping each one to a value.

Package [`dfio.string-table`], page 9.

Source [`string-table.lisp`], page 5.

Direct superclasses
`structure-object`.

Direct methods
[`print-object`], page 16.

Direct slots

`table` [Slot]
Type `hash-table`
Initform `(make-hash-table :test (function equalp))`
Readers [`string-table-table`], page 20.
Writers [`(setf string-table-table)`], page 20.

4.1.5 Classes

`data-column` [Class]
Package [`dfio.data-column`], page 10.

Source [`data-column.lisp`], page 6.

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]
Package [`dfio.string-table`], page 9.
Type `dfio.string-table:string-table`
Initform `(dfio.string-table:string-table)`

4.2 Internals

4.2.1 Special variables

`*default-external-format*` [Special Variable]
External format used for opening files

Package [`dfio`], page 10.

Source [`utils.lisp`], page 6.

`+exponent-chars+` [Special Variable]
Default exponent characters.

Package [`dfio.decimal`], page 9.

Source [`decimal.lisp`], page 5.

4.2.2 Macros

`with-csv-output-stream` ((*name inp*) **&body** *body*) [Macro]

Package [dfio], page 10.

Source [utils.lisp], page 6.

`with-input-stream` ((*name inp*) **&body** *body*) [Macro]

Package [dfio], page 10.

Source [utils.lisp], page 6.

4.2.3 Ordinary functions

`%in-stream` (*source*) [Function]

Package [dfio], page 10.

Source [utils.lisp], page 6.

`%out-stream` (*source*) [Function]

creates a stream from the given thing, trying to DWIM

Package [dfio], page 10.

Source [utils.lisp], page 6.

`copy-string-table` (*instance*) [Function]

Package [dfio.string-table], page 9.

Source [string-table.lisp], page 5.

`csv-to-data-columns` (*source skip-first-row?* **&key** *map-alist*) [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 10.

Source [delimited-text.lisp], page 7.

`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 9.

Source [decimal.lisp], page 5.

`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 9.

Source [decimal.lisp], page 5.

`string-table-get` (*string-table string*) [Function]
Synonym for GETHASH, used internally.

Package [`dfio.string-table`], page 9.

Source [`string-table.lisp`], page 5.

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

Package [`dfio.string-table`], page 9.

Source [`string-table.lisp`], page 5.

`string-table-p` (*object*) [Function]

Package [`dfio.string-table`], page 9.

Source [`string-table.lisp`], page 5.

`string-table-table` (*instance*) [Reader]
`(setf string-table-table)` (*instance*) [Writer]

Package [`dfio.string-table`], page 9.

Source [`string-table.lisp`], page 5.

Target Slot
[`table`], page 17.

4.2.4 Types

`non-negative-integer` () [Type]

Package [`dfio.data-column`], page 10.

Source [`data-column.lisp`], page 6.

`str-strm-file` () [Type]

Package [`dfio`], page 10.

Source [`utils.lisp`], page 6.

Appendix A Indexes

A.1 Concepts

(Index is nonexistent)

A.2 Functions

%

(

C

D

F

G

M

P

R

S

W

A.3 Variables

*

+

D

F

I

M

R

S

T

A.4 Data types

C

D

F

N

P

S

T

U

W