Package 'kvh'

July 22, 2025

Type Package

Title Read/Write Files in Key-Value-Hierarchy Format

Version 1.4.2

Date 2022-01-26

Author Serguei Sokol

Maintainer Serguei Sokol <sokol@insa-toulouse.fr>

Description

The format KVH is a lightweight format that can be read/written both by humans and machines. It can be useful in situations where XML or alike formats seem to be an overkill. We provide an ability to parse KVH files in R pretty fast due to 'Rcpp' use.

URL http://serguei.sokol.free.fr/kvh-format/

BugReports https://github.com/sgsokol/kvh/issues

License GPL (>= 2)

Imports Rcpp (>= 0.12.12)

LinkingTo Rcpp

Suggests testthat

NeedsCompilation yes

Biarch yes

RoxygenNote 7.1.1

Encoding UTF-8

Repository CRAN

Date/Publication 2022-01-26 15:22:43 UTC

Contents

---

kvh-package *Write/read KVH (key-value hierarchy) file*

Description

The format KVH is a lightweight format that can be read/written both by humans and machines. It can be useful in situations where XML or alike formats seem to be an overkill. We provide an ability to manipulate kvh files in R with a good efficiency due to Rcpp use. The content read in kvh file is hierarchically organized in nested lists. The key-values are always returned as character strings. It's up to user to convert them further in useful types (numeric vectors, matrices and so on).

Author(s)

Serguei Sokol.

Maintainer: Serguei Sokol <sokol@insa-toulouse.fr>

References

http://serguei.sokol.free.fr/kvh-format/

Examples

```
## Not run:
   # prepare object to write to kvh file
   obj=list(x=structure(1:3, names=letters[1:3]), R=R.version)
   # write it
   obj2kvh(obj, "test", "test.kvh") # will create test.kvh file
   # read it back
   l=kvh_read("test.kvh")
   # check a field
  l$test$x # NB. it has a character values put in a list not a numeric vector as it was in obj.

## End(Not run)
```

esc_kvh_k *Escape Special Characters in a key*

Description

Escape Tabs, Newlines and Backslashes in a string which will be used as a key in a KVH file.

Usage

```
esc_kvh_k(s)
```

Arguments

s	string

Details

Escape is done by butting a backslash before a special character.'

Value

escaped string

esc_kvh_v *Escape Special Characters in a value*

Description

Escape Newlines and Backslashes in a string which will be used as a key in a KVH file.

Usage

```
esc_kvh_v(s)
```

Arguments

s	string

Details

Escape is done by butting a backslash before a special character.'

Value

escaped string

kvh_get_matrix *Get matrix from kvh file*

Description

Given a read connection to kvh file and a vector of keys pointing to a matrix, return this matrix

Usage

```
kvh_get_matrix(f, v)
```

Arguments

f	connection from which kvh file can be read
v	character vector of key-subkeys pointing to a matrix

Details

It is expected that matrix in the kvh file has its upper-leftmost item called "row_col" and it has rownames in the first column and colnames in the first row.

Value

matrix read from kvh

Examples

```
# write a test matrix
obj2kvh(list(comment="this is a test matrix",  m=diag(2)), "li", "test.kvh")
# read it back
mr=kvh_get_matrix(file("test.kvh"), c("li", "m"))
# clean
unlink("test.kvh")
```

kvh_read *Parse file in KVH format*

Description

Returns a list with names formed form kvh keys and values formed from kvh values If a kvh value has sub-keys, it is returned as a nested list. Otherwise it is returned as a character string.

Usage

```
kvh_read(
  fn,
  comment_str = "",
  strip_white = FALSE,
  skip_blank = FALSE,
  split_str = "",
  follow_url = FALSE
)
```

Arguments

fn	character kvh file name.
comment_str	character optional comment string (default empty ""). If non empty, the comment string itself and everything following it on the line is ignored. Note that lines are first appended if end lines are escaped and then a search for a comment string is done.
strip_white	logical optional control of white spaces on both ends of keys and values (default FALSE)
skip_blank	logical optional control of lines composed of only white characters after a possible stripping of a comment (default FALSE)
split_str	character optional string by which a value string can be splitted in several strings (default: empty string, i.e. no splitting)
follow_url	logical optional control of recursive kvh reading and parsing. If set to TRUE and a value starts with 'file://' then the path following this prefix will be passed as argument 'fn' to another 'kvh_read()' call. The list returned by this last call will be affected to the corresponding key instead of the value 'file://...'. If a circular reference to some file is detected, a warning is emmited and the faulty value 'file://...' will be left without change. The rest of the file is proceeded as usual. If a path is relative one (i.e. not strating with '/' neither 'C:/' or alike on windows paltform) then its meant relative to the location of the parent kvh file, not the current working directory.

---

obj2kvh *Writing/Adding an R Object to KVH File.*

Description

Formats an object before writing it in kvh file.

Usage

```
obj2kvh(obj, objname = NULL, conct = stdout(), indent = 0)
```

Arguments

obj	an R object
objname	character object name to write in kvh file
conct	connection opened for writing
indent	is tab offset for object name

Details

Scalar, vector, matrix and list are pre-processed. Other objects are written as an output string of toString() function To add a content to existent file use "a" as open mode `fcn=file("m.kvh", "a")` `obj2kvh()` can be used along the code advancing in the calculations. Writing in a subfield of an already started key requires use of appropriate indent value. The file is started with indent=0 and every sub-field increments the indent by 1. If objname is NULL and obj is not a scalar value, the content of obj is written in kvh file without additional indent.

Value

None

Examples

```
m=matrix(1:6,2,3);
fcn=file("m.kvh", "w");
obj2kvh(m, "m", fcn);
close(fcn);
# clean
unlink("m.kvh")
```

---

obj_by_keys *Get Object Identified by its Keys.*

Description

Given a named nested list returned by kvh_read(), get a particular item from it. The object is identified by a series of hierarchical keys, first key corresponds to the first hierarchical level, the second corresponds to the second and so on.

Usage

```
obj_by_keys(li, keys)
```

Arguments

li	a named nested list returned by kvh_read()
keys	character vector naming key suites to identify an object

Value

an object corresponding to li[[keys[1]][[keys[2]][[...]]. Return NULL if non valid keys.

Index