

Package ‘ascii’

January 2, 2012

Maintainer David Hajage <dhajage@gmail.com>

License GPL (>= 2)

Title Export R objects to several markup languages

Type Package

Author David Hajage

Description Coerce R object to asciidoc, txt2tags, restructuredText.org, textile or pandoc syntax. Package comes with a set of drivers for Sweave.

Version 2.1

URL <http://eusebe.github.com/ascii/>, <http://github.com/eusebe/ascii/>

Date 2009-07-20

Depends R (>= 2.13), utils, methods

Suggests survival, Hmisc, xtable, R2HTML, cacheSweave, weaver

Collate

‘asciiAnova.r’ ‘asciiDataFrame.r’ ‘asciiDefault.r’ ‘asciiDensity.r’ ‘asciiDescr.r’ ‘asciiEpi.r’ ‘asciiGlm.r’ ‘asciiHmisc.r’ ‘asci

Repository CRAN

Date/Publication 2011-09-29 11:27:34

R topics documented:

ascii.default	2
asciiCbind	6
Asciidoc	6
asciiList	7
asciiMixed	7
asciiTable	7
cbind.ascii	8

convert	8
createreport	9
fig	11
out	12
paragraph	12
plim	13
print	13
print.fig	14
print.out	15
print.paragraph	15
print.section	16
print.sexpr	16
print.verbatim	17
RtangleAscii	17
section	18
sexpr	18
verbatim	19
Index	20

ascii.default	<i>Export R objects to several markup languages</i>
---------------	---

Description

Convert an R object to an `ascii` object, which can then be printed with `asciidoc`, `txt2tags`, `reStructuredText`, `org`, `textile` or `pandoc` syntax.

Usage

```
## Default S3 method:
```

```
ascii(x, include.rownames = TRUE, include.colnames = TRUE, rownames = NULL, colnames = NULL, format = "f
```

```
## S3 method for class 'describe'
```

```
ascii(x, condense = TRUE, ...)
```

```
## S3 method for class 'sessionInfo'
```

```
ascii(x, locale = TRUE, ...)
```

```
## S3 method for class 'survfit'
```

```
ascii(x, scale = 1, print.rmean = getOption("survfit.print.rmean"), rmean = getOption("survfit.rmean"))
```

```
ascii(x, ...)
```

Arguments

<code>x</code>	An R object of class found among <code>methods(ascii)</code> . If <code>x</code> is a list, it should be a list of character strings (it will produce a bulleted list output by default).
<code>include.rownames</code>	logical. If TRUE the rows names are printed. Default value depends of class of <code>x</code> .
<code>include.colnames</code>	logical. If TRUE the columns names are printed. Default value depends of class of <code>x</code> .
<code>rownames</code>	Character vector (replicated or truncated as necessary) indicating rownames of the corresponding rows. If NULL (default) the row names are not modified
<code>colnames</code>	Character vector (replicated or truncated as necessary) indicating colnames of the corresponding columns. If NULL (default) the column names are not modified
<code>format</code>	Character vector or matrix indicating the format for the corresponding columns. These values are passed to the <code>formatC</code> function. Use "d" (for integers), "f", "e", "E", "g", "G", "fg" (for reals), or "s" (for strings). "f" gives numbers in the usual xxx.xxx format; "e" and "E" give n.ddde+nn or n.dddE+nn (scientific format); "g" and "G" put <code>x[i]</code> into scientific format only if it saves space to do so. "fg" uses fixed format as "f", but digits as number of <i>significant</i> digits. Note that this can lead to quite long result strings. Finally, "nice" is like "f", but with 0 digits if <code>x</code> is an integer. Default depends on the class of <code>x</code> .
<code>digits</code>	Numeric vector of length equal to the number of columns of the resulting table (otherwise it will be replicated or truncated as necessary) indicating the number of digits to display in the corresponding columns. Default is 2.
<code>decimal.mark</code>	The character to be used to indicate the numeric decimal point. Default is ".".
<code>na.print</code>	The character string specifying how NA should be formatted specially. Default is "".
<code>caption</code>	Character vector of length 1 containing the table's caption or title. Set to "" to suppress the caption. Default value is NULL.
<code>caption.level</code>	Character or numeric vector of length 1 containing the caption's level. Can take the following values: 0 to 5, "." (block titles in asciidoc markup), "s" (strong), "e" (emphasis), "m" (monospaced) or "" (no markup). Default is NULL.
<code>width</code>	Numeric vector of length one containing the table width relative to the available width (expressed as a percentage value, 1...99). Default is 0 (all available width).
<code>frame</code>	Character vector of length one. Defines the table border, and can take the following values: "topbot" (top and bottom), "all" (all sides), "none" and "sides" (left and right). The default value is NULL.
<code>grid</code>	Character vector of length one. Defines which ruler lines are drawn between table rows and columns, and can take the following values: "all", "rows", "cols" and "none". Default is NULL.
<code>valign</code>	Vector or matrix indicating vertical alignment of all cells in table. Can take the following values: "top", "bottom" and "middle". Default is "".
<code>header</code>	logical or numeric. If TRUE or 1, 2, ..., the first line(s) of the table is (are) emphasized. The default value depends of class of <code>x</code> .

footer	logical or numeric. If TRUE or 1, the last line(s) of the table is (are) emphasized. The default value depends of class of x.
align	Vector or matrix indicating the alignment of the corresponding columns. Can be composed with "r" (right), "l" (left) and "c" (center). Default value is NULL.
col.width	Numeric vector of length equal to the number of columns of the resulting table (otherwise it will be replicated or truncated as necessary) indicating width of the corresponding columns (integer proportional values). Default is 1.
style	Character vector or matrix indicating the style of the corresponding columns. Can be composed with "d" (default), "s" (strong), "e" (emphasis), "m" (monospaced), "h" (header) "a" (cells can contain any of the AsciiDoc elements that are allowed inside document), "l" (literal), "v" (verse; all line breaks are retained). Default is NULL.
tgroup	Character vector or a list of character vectors defining major top column headings. The default is to have none (NULL).
n.tgroup	A numeric vector or a list of numeric vectors containing the number of columns for which each element in tgroup is a heading. For example, specify tgroup=c("Major 1", "Major 2"), n.tgroup=c(3, 3) if "Major 1" is to span columns 1-3 and "Major 2" is to span columns 4-6.
talign	Character vector of length one defining alignment of major top column headings.
tvalign	Character vector of length one defining vertical alignment of major top column headings.
tstyle	Character vector of length one indicating the style of major top column headings
bgroup	Character vector or list of character vectors defining major bottom column headings. The default is to have none (NULL).
n.bgroup	A numeric vector containing the number of columns for which each element in bgroup is a heading.
balign	Character vector of length one defining alignment of major bottom column headings.
bvalign	Character vector of length one defining vertical alignment of major bottom column headings.
bstyle	Character vector of length one indicating the style of major bottom column headings
lgroup	Character vector or list of character vectors defining major left row headings. The default is to have none (NULL).
n.lgroup	A numeric vector containing the number of rows for which each element in lgroup is a heading. Column names count in the row numbers if include.colnames = TRUE.
lalign	Character vector of length one defining alignment of major left row headings.
lvalign	Character vector of length one defining vertical alignment of major left row headings.
lstyle	Character vector of length one indicating the style of major left row headings
rgroup	Character vector or list of character vectors defining major right row headings. The default is to have none (NULL).

<code>n.rgroup</code>	A numeric vector containing the number of rows for which each element in <code>rgroup</code> is a heading. Column names count in the row numbers if <code>include.colnames = TRUE</code> .
<code>ralign</code>	Character vector of length one defining alignment of major right row headings.
<code>rvalign</code>	Character vector of length one defining vertical alignment of major right row headings.
<code>rstyle</code>	Character vector of length one indicating the style of major right row headings
<code>list.type</code>	Character vector of length one indicating the list type ("bullet", "number", "label" or "none"). If "label", <code>names(list)</code> is used for labels. Default is "bullet".
<code>...</code>	Additional arguments. (Currently ignored.)
<code>condense</code>	default is TRUE to condense the output with regard to the 5 lowest and highest values and the frequency table (<code>describe()</code> in package <code>Hmisc</code>).
<code>locale</code>	show locale information?
<code>scale</code>	A numeric value to rescale the survival time, e.g., if the input data to <code>survfit</code> were in days, <code>scale=365</code> would scale the printout to years (see <code>print.survfit()</code> in package <code>survival</code>).
<code>print.rmean</code>	Option for computation and display of the restricted mean (see <code>print.survfit()</code> in package <code>survival</code>).
<code>rmean</code>	Option for computation and display of the restricted mean (see <code>print.survfit()</code> in package <code>survival</code>).

Details

The nature of the generated output depends on the class of `x`. For example, `summary.table` objects produce a bulleted list while `data.frame` objects produce a table of the entire `data.frame`.

Sometimes, arguments are not active, depending of the features implemented in the markup language generated. All arguments are active when `asciidoc` syntax is produced.

The available method functions for `ascii` are given by `methods(ascii)`. Users can extend the list of available classes by writing methods for the generic function `ascii`. All method functions should return an object of class "ascii".

Value

This function returns an object of class "asciiTable", "asciiList" or "asciiMixed".

Author(s)

David Hajage <dhajage@gmail.com>

Examples

```
data(esoph)
ascii(esoph[1:10,])
tab <- table(esoph$agegp, esoph$alcgp)
ascii(tab)
```

```
print(ascii(tab), type = "t2t")
print(ascii(tab), type = "rest")
print(ascii(tab), type = "org")
ascii(summary(tab))
```

asciiCbind	<i>ascii table generator</i>
------------	------------------------------

Description

ascii table generator

Author(s)

David Hajage

Asciidoc	<i>Sweave wrappers</i>
----------	------------------------

Description

Sweave wrappers

Usage

```
Asciidoc(file, driver = RweaveAsciidoc, syntax = SweaveSyntaxNoweb, encoding = "", ...)
```

```
T2t(file, driver = RweaveT2t, syntax = SweaveSyntaxNoweb, encoding = "", ...)
```

```
ReST(file, driver = RweaveReST, syntax = SweaveSyntaxNoweb, encoding = "", ...)
```

```
Org(file, driver = RweaveOrg, syntax = SweaveSyntaxNoweb, encoding = "", ...)
```

```
Textile(file, driver = RweaveTextile, syntax = SweaveSyntaxNoweb, encoding = "", ...)
```

```
Pandoc(file, driver = RweavePandoc, syntax = SweaveSyntaxNoweb, encoding = "", ...)
```

Arguments

file	Name of Sweave source file.
driver	Sweave driver
syntax	Sweave syntax
encoding	Encoding
...	Further arguments passed to the driver's setup function.

Author(s)

David Hajage <dhajage@gmail.com>

See Also

[Sweave](#)

asciiList	<i>ascii list generator</i>
-----------	-----------------------------

Description

ascii list generator

Author(s)

David Hajage

asciiMixed	<i>ascii mixed generator</i>
------------	------------------------------

Description

ascii mixed generator

Author(s)

David Hajage

asciiTable	<i>ascii table generator</i>
------------	------------------------------

Description

ascii table generator

Author(s)

David Hajage

<code>cbind.ascii</code>	<i>Cbind two ascii objects</i>
--------------------------	--------------------------------

Description

Cbind two ascii objects

Usage

```
cbind.ascii(..., caption = NULL, caption.level = NULL, frame = NULL, grid = NULL, col.width = 1, width =
```

Arguments

<code>...</code>	ascii objects
<code>caption</code>	see ?ascii
<code>caption.level</code>	see ?ascii
<code>frame</code>	see ?ascii
<code>grid</code>	see ?ascii
<code>col.width</code>	see ?ascii
<code>width</code>	see ?ascii

Details

This function binds cols of two ascii table.

Value

An "asciiCbind" object.

Author(s)

David Hajage

<code>convert</code>	<i>Convert a file with specified backend</i>
----------------------	--

Description

Convert a file with specified backend

Usage

```
convert(i, d = NULL, f = NULL, e = NULL, o = NULL, backend = getOption("asciiBackend"), cygwin = FALSE, op
```

Arguments

i	input file
d	output directory
f	format
e	encoding
O	other options
backend	backend ("asciidoc", "t2t" or "pandoc")
cygwin	use cygwin?
open	open resulting file?

Details

This function convert a file with asciidoc, txt2tags or pandoc backend

Value

Nothing

Author(s)

David Hajage

createreport	<i>Report creation</i>
--------------	------------------------

Description

Produce a report
Report generator

Usage

```
createreport(..., list = NULL, file = NULL, format = NULL, open = TRUE, backend = getOption("asciiBacken
```

Arguments

...	R objects (not used if "list" is not NULL)
list	list of R objects
file	name of the output file (without extension)
format	format of the output file
open	open resulting file?
backend	backend
encoding	encoding

options	other options
cygwin	use cygwin?
title	title of the report
author	author of the report
email	email of the author
date	date

Details

Produce a report from a list of R objects. This function can be used directly, or through a Report object (see examples). `Report$new()` creates a new object, `Report$create()` produce a report. Exportation options can be specified with `Report$nameoftheoption <- option` or directly in `Report$create(nameoftheoption = option)`.

Special objects can be used to create sections (see `?section`), paragraphs (see `?paragraph`), verbatim environment (see `?verbatim` and to insert figures (see `?fig`) or inline results (see `?sexpr`). Helpers exist: `Report$addSection()`, `Report$addParagraph()`, `Report$addVerbatim()`, `Report$addFig()`.

It needs a working installation of asciidoc, a2x tool chain, txt2tags, pandoc and/or markdown2pdf.

Value

Nothing

Author(s)

David Hajage

David Hajage

Examples

```
## Not run:
options(asciiType = "asciidoc")
createreport(head(esoph))

r <- Report$new(author = "David Hajage", email = "dhajage at gmail dot com")
r$add(section("First section"))
r$addSection("First subsection", 2)
r$add(paragraph("The data set has", sexpr(nrow(esoph)), " lines. See yourself:"), esoph)
r$addSection("Second subsection: age and alc group", 2)
tab <- with(esoph, table(alcgp, agegp))
r$add(ascii(tab), ascii(summary(tab), format = "nice"))
r$create()
r$format <- "slidy"
r$createt()

r$title <- "R report example"
r$author <- "David Hajage"
r$email <- "dhajage at gmail dot com"
options(asciiType = "pandoc")
```

```
r$backend <- "pandoc"  
r$format <- "odt"  
r$create()  
  
r$create(backend = "markdown2pdf", format = "pdf")  
  
## End(Not run)
```

fig

Insert figure

Description

graph can be used with export function to insert an R graphic.

Usage

```
fig(file = NULL, graph = NULL, format = NULL, ...)
```

Arguments

file	character string (
graph	a recordedplot, a lattice plot, a ggplot, or an expression producing a plot (optional if the file already exists)
format	jpg, png or pdf (or guessed with the file name)
...	additional arguments (passed to format options)

Value

A fig object

Author(s)

David Hajage

out *Export R objects*

Description

out can be used with export function to insert an R results

Usage

```
out(x, results = "verbatim")
```

Arguments

x	an R object
results	if 'verbatim', the output is included in a verbatim environment. If 'ascii', the output is taken to be already proper markup and included as is.

Value

An out object

Author(s)

David Hajage

paragraph *Create a paragraph*

Description

paragraph can be used with export function to add... a paragraph

Usage

```
paragraph(..., new = TRUE)
```

Arguments

...	strings composing the paragraph
new	whether to create a new paragraph or to continue a preceding one

Value

A paragraph object.

Author(s)

David Hajage

plim *format p values*

Description

format p values

Usage

```
plim(p, digits = 4)
```

Arguments

p	p values
digits	number of digits

Value

formatted p values

Author(s)

David Hajage

print *Print ascii object*

Description

Function displaying the asciidoc, txt2tags, reStructuredText, org or textile code associated with the supplied object of class `ascii`.

Show method for `ascii` objects

Arguments

x	An object of class "asciiTable", "asciiList", "asciiMixed", "asciiCbind" or "Report".
type	Type of syntax produce. Possible values for type are "asciidoc", "t2t", "rest", "org", "textile" or "pandoc". Default value produce asciidoc syntax.
file	A character string naming the file to print to. Default is NULL (print to the console).
append	If TRUE, code will be appended to file instead of overwriting it. Default value is FALSE

escape	If TRUE, characters in list.escape will be printed with a \. Default value is FALSE
list.escape	Character vector. Default value is c("_", "\\^")
help	logical print help? (objects of class "Report")
...	Additional arguments. (Currently ignored.)
object	ascii or Report object

Details

The package provides the new global option `asciiType`. Default value is "asciidoc" (see examples).

Author(s)

David Hajage <dhajage@gmail.com>

See Also

[ascii](#)

Examples

```
data(esoph)
ascii(esoph[1:10,])
print(ascii(esoph[1:10,]), type = "t2t")
print(ascii(esoph[1:10,]), type = "rest")
print(ascii(esoph[1:10,]), type = "org")
print(ascii(esoph[1:10,]), type = "textile")
print(ascii(esoph[1:10,]), type = "pandoc")
options(asciiType = "rest")
ascii(esoph[1:10,])
options(asciiType = "asciidoc")
```

print.fig

Print an graph object

Description

Print an graph object

Usage

```
## S3 method for class 'fig'
print(x, backend = getOption("asciiBackend"), ...)
```

Arguments

x	an graph object
backend	ascii backend
...	not used

Author(s)

David Hajage

print.out *Print an out object*

Description

Print an out object

Usage

```
## S3 method for class 'out'  
print(x, backend = getOption("asciiBackend"), ...)
```

Arguments

x	an out object
backend	ascii backend
...	not used

Author(s)

David Hajage

print.paragraph *Print a paragraph object*

Description

Print a paragraph object

Usage

```
## S3 method for class 'paragraph'  
print(x, ...)
```

Arguments

x	a paragraph object
...	not used

Author(s)

David Hajage

<code>print.section</code>	<i>Print a section object</i>
----------------------------	-------------------------------

Description

Print a section object

Usage

```
## S3 method for class 'section'
print(x, backend = getOption("asciiBackend"), ...)
```

Arguments

x	a section object
backend	ascii backend
...	not used

Author(s)

David Hajage

<code>print.sexpr</code>	<i>Print a sexpr object</i>
--------------------------	-----------------------------

Description

Print a sexpr object

Usage

```
## S3 method for class 'sexpr'
print(x, ...)
```

Arguments

x	a sexpr object
...	not used

Author(s)

David Hajage

<code>print.verbatim</code>	<i>Print a verbatim object</i>
-----------------------------	--------------------------------

Description

Print a verbatim object

Usage

```
## S3 method for class 'verbatim'  
print(x, backend = getOption("asciiBackend"), ...)
```

Arguments

x	a verbatim object
backend	ascii backend
...	not used

Author(s)

David Hajage

<code>RtangleAscii</code>	<i>RtangleAscii</i>
---------------------------	---------------------

Description

RtangleAscii

section	<i>Create a section</i>
---------	-------------------------

Description

section can be used with `export` function to add... a section

Usage

```
section(caption, caption.level = 1)
```

Arguments

caption	a string
caption.level	caption level

Value

A section object.

Author(s)

David Hajage

sexpr	<i>Insert an inline R result</i>
-------	----------------------------------

Description

sexpr can be used with `export` function to insert an inline R results

Usage

```
sexpr(x)
```

Arguments

x	an R results (of length one)
---	------------------------------

Value

A sexpr object.

Author(s)

David Hajage

`verbatim`*Create a verbatim paragraph*

Description

`verbatim` can be used with `export` function to add a verbatim paragraph

Usage

```
verbatim(...)
```

Arguments

... strings composing the paragraph (line by line)

Value

A verbatim object.

Author(s)

David Hajage

Index

*Topic **IO**

Asciidoc, 6

*Topic **file**

Asciidoc, 6

*Topic **print**

ascii.default, 2

print, 13

ascii, 14

ascii(ascii.default), 2

ascii.default, 2

asciiCbind, 6

Asciidoc, 6

asciiList, 7

asciiMixed, 7

asciiTable, 7

cbind.ascii, 8

convert, 8

createreport, 9

fig, 11

graph(fig), 11

Org(Asciidoc), 6

out, 12

Pandoc(Asciidoc), 6

paragraph, 12

plim, 13

print, 13

print,asciiCbind-method(print), 13

print,asciiList-method(print), 13

print,asciiMixed-method(print), 13

print,asciiTable-method(print), 13

print,Report-method(print), 13

print.fig, 14

print.out, 15

print.paragraph, 15

print.section, 16

print.sexpr, 16

print.verbatim, 17

Report(createreport), 9

ReST(Asciidoc), 6

RtangleAscii, 17

section, 18

sexpr, 18

show,asciiCbind-method(print), 13

show,asciiList-method(print), 13

show,asciiMixed-method(print), 13

show,asciiTable-method(print), 13

show,Report-method(print), 13

Sweave, 7

T2t(Asciidoc), 6

Textile(Asciidoc), 6

verbatim, 19