

Package: pack (via r-universe)

September 5, 2024

Type Package

Title Convert values to/from raw vectors

Version 0.1-1

Date 2008-08-22

Author Josh Ulrich

Maintainer Josh Ulrich <josh.m.ulrich@gmail.com>

Description Functions to easily convert data to binary formats other programs/machines can understand.

License GPL-3

LazyLoad yes

Suggests RUnit

Repository <https://joshualrich.r-universe.dev>

RemoteUrl <https://github.com/joshualrich/pack>

RemoteRef HEAD

RemoteSha 298b650350eb25192ecec54b858b4c53c8665a42

Contents

pack-package	2
numToRaw	2
pack	3
rawToNum	4
unpack	5

Index	7
--------------	----------

pack-package *Convert values to/from raw vectors*

Description

pack allows R programmers to easily put their data into binary formats that other programs / machines can understand.

Details

Package: pack
Type: Package
Version: 0.1-1
Date: 2008-08-22
License: GPL-3
LazyLoad: yes

Author(s)

Author: Josh Ulrich Maintainer: Josh Ulrich <josh.m.ulrich@gmail.com>

References

<http://perldoc.perl.org/functions/pack.html>

numToRaw *Numeric to Raw vector*

Description

Convert numeric values to a raw vector.

Usage

```
numToRaw(x, nBytes = 1)
```

Arguments

x A number to be converted
nBytes The number of bytes to use

Value

A raw vector containing the bytes representing x.

Author(s)

Josh Ulrich

See Also

[rawToNum](#)

Examples

```
# Will be left null padded
(x <- numToRaw(421,4))
rawToNum(x,2)
rawToNum(x,4)
```

pack

Pack raw vectors

Description

Combine values into a raw vector according to the values in template.

Usage

```
pack(template, ...)
```

Arguments

template	A string, see 'Details'
...	Values/objects to be packed into a raw vector

Details

Currently supported template values are:

- 'a' - A null padded string
- 'A' - A space padded string
- 'b' - An ascending bit order binary vector, (must be a multiple of 8 long)
- 'B' - An descending bit order binary vector, (must be a multiple of 8 long)
- 'C' - An unsigned char (8-bit byte/octet) value
- 'v' - An unsigned short (16-bit) in "VAX" (little-endian) order
- 'V' - An unsigned long (32-bit) in "VAX" (little-endian) order
- 'x' - A null byte

Both 'a' and 'A' may be followed by a repeat value. A repeat value of '*' will cause the remainder of the bytes in values to be placed in the last element.

'/' allows packing and unpacking of a sequence of values where the packed structure contains a packed item count followed by the packed items themselves.

If template requires more arguments to pack than actually given, pack pads with null bytes. If template requires fewer arguments to pack than actually given, extra arguments are ignored.

Value

A raw vector following the elements in template.

Author(s)

Josh Ulrich

References

<http://perldoc.perl.org/functions/pack.html>

See Also

[unpack](#)

Examples

```
(x <- pack('A4 C v A8 V', 'pack', 2, 8, 'sequence', 68098))
(u1 <- unpack('A4 C H*', x))
(u2 <- unpack('v/A V', u1[[3]]))
```

rawToNum

Raw to Numeric vector

Description

Convert raw values to numeric.

Usage

```
rawToNum(x, nBytes = 1)
```

Arguments

x	A raw vector to be converted
nBytes	The number of bytes to use

Value

A numeric value containing the bytes in x.

Author(s)

Josh Ulrich

See Also

[numToRaw](#)

Examples

```
# Will be left null padded
(x <- numToRaw(421,4))
rawToNum(x,2)
rawToNum(x,4)
```

 unpack

Unpack raw vectors

Description

Break a raw vector into chunks according to the values in `template`.

Usage

```
unpack(template, ...)
```

Arguments

<code>template</code>	A string, see 'Details'
<code>...</code>	Raw vector(s) to be unpacked

Details

Currently supported `template` values are:

- 'a' - A null padded string (as of R-2.8.0, strings cannot contain embedded nulls)
- 'A' - A space padded string
- 'b' - An ascending bit order binary vector, (must be a multiple of 8 long)
- 'B' - An descending bit order binary vector, (must be a multiple of 8 long)
- 'C' - An unsigned char (8-bit byte/octet) value
- 'v' - An unsigned short (16-bit) in "VAX" (little-endian) order
- 'V' - An unsigned long (32-bit) in "VAX" (little-endian) order
- 'f' - A single-precision float
- 'd' - A double-precision float
- 'x' - Skip next byte, and push nothing onto return value for it
- 'H' - A raw byte

Values 'a', 'A', and 'H' may be followed by a repeat value. A repeat value of '*' will cause the remainder of the bytes in values to be placed in the last element.

'/' allows packing and unpacking of a sequence of values where the packed structure contains a packed item count followed by the packed items themselves.

If there are more `template` values or if the repeat count of a field or a group is larger than what the remainder `...` allows, `unpack` returns `NULL`. If `...` is longer than what is described by `template`, the rest is ignored.

Value

A list with an element for each value in `template`.

Note

When unpacking, `'A'` strips trailing whitespace and nulls and `'a'` returns data verbatim (but with embedded nulls removed, since strings cannot contain embedded nulls as of R-2.8.0).

Author(s)

Josh Ulrich

References

<http://perldoc.perl.org/functions/unpack.html>

See Also

[pack](#)

Examples

```
(x <- pack('A4 C v A8 V', 'pack', 2, 8, 'sequence', 68098))
(u1 <- unpack('A4 C H*', x))
(u2 <- unpack('v/A V', u1[[3]]))
```

Index

- * **package**

- pack-package, 2

- * **utilities**

- numToRaw, 2

- pack, 3

- rawToNum, 4

- unpack, 5

numToRaw, 2, 5

pack, 3, 6

pack-package, 2

rawToNum, 3, 4

unpack, 4, 5