Package 'readHAC'

February 2, 2017

Type Package

Title Read Acoustic HAC Format

Version 1.0

Date 2017-02-01

Author Kasper Kristensen [aut, cre]

Maintainer Kasper Kristensen <kaskr@dtu.dk>

Description Read Acoustic HAC format.

License GPL-2

URL https://github.com/kaskr/HAC

BugReports https://github.com/kaskr/HAC/issues

LazyLoad yes

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-02-02 02:05:07

R topics documented:

readHAC-package *Read acoustic HAC raw data*

Description

The HAC data format is a binary format containing so-called `tuples`. A tuple can hold various sorts of information depending on the tuple type. For instance tuples exist to specify positions, echosounder information and acoustic signal data etc. This R package can read, write and subset the HAC data format.

Details

See the description of the ICES HAC standard data exchange format, version 1.60.

References

McQuinn, Ian H., et al. Description of the ICES HAC standard data exchange format, version 1.60. Conseil international pour l'exploration de la mer, 2005. http://biblio.uqar.ca/archives/30005500.pdf

Examples

```
require(readHAC)

#########################################################################
## Example file
hacfile <- system.file("hac", "Hac-test_000001.hac", package="readHAC")

#########################################################################
## Step 1. Read hac data into R
hac <- readHAC(hacfile)
print(hac)

#########################################################################
## Step 2. Select sub-components
pingdata <- ( subset(hac, softwarechannel==3 & type==10000) )
channel  <- ( subset(hac, softwarechannel==3 & type==9001 ) )
echosounder <- ( subset(hac, echosounder==channel$echosounder & type==901) )

#########################################################################
## Step 3. Parse the binary data
print( parseHAC(pingdata) )
info <- parseHAC(channel)[5:7]
s <- ( parseHAC(pingdata)$"Sample value" )
s[s>0] <- NA  ## discard positive dB values
sec <- parseHAC(pingdata)$"Time CPU ANSI"; sec <- sec - min(sec)
flip <- function(x) t( x[nrow(x):1, ] )
image(sec, 1:nrow(s), flip(s), axes=FALSE, ylab="sample")
axis(1)
```

```
at <- seq(nrow(s), 1, by=-100)
axis(2, at=at, labels=nrow(s)-at)
box()
legend("topright", legend=paste(names(info), unlist(info)) )
```

---

parseHAC *Parse binary HAC.*

Description

Parse binary HAC to a list of data values.

Usage

```
parseHAC(hac, split = FALSE, split.by = paste(hac$type, hac$length),
  units = TRUE)
```

Arguments

hac	Object of class HAC to be parsed.
split	Force parsing of incompatiple tuples by first splitting the raw data?
split.by	If split=TRUE then split by this factor.
units	Convert to human readable units?

Details

HAC parsing can be performed for one or multiple tuples of the same type and length. The binary tuples are translated to data values according to the definition document.

Value

Object of class tuple.

---

readHAC *Read HAC data into R.*

Description

Read raw HAC data file

Usage

```
readHAC(file)
```

Arguments

file	File to read.

Details

This function reads the binary HAC format and locates the tuples.

Value

HAC object.

---

writeHAC *Write HAC binary data.*

Description

Write raw HAC data file

Usage

```
writeHAC(x, file)
```

Arguments

x	HAC object
file	File to write to.

Details

This function writes the binary HAC format. The output file begins with "ac 00 00 00" followed by the binary tuples defined by the HAC object x. Note that the function does not perform a check for mandatory tuples.

[.HAC *Extract tuples.*

Description

Extract tuples of HAC object.

Usage

```
## S3 method for class 'HAC'
x[i, ...]
```

Arguments

x	HAC object
i	Integer vector
...	Currently not used

Details

Extract subset of tuples. For instance x[1:2] extracts the first two tuples. Alternatively the method can be indirectly invoked by the `subset` function.

Value

HAC object

Examples

```
x[1:2]
subset(x, type == 10000)
split(x, x$type)
```

Index