

# Package ‘sattagutils’

June 14, 2020

**Title** utils for manipulating sat tag output

**Version** 0.2.1

**Date** 2020-02-05

**Author** William R. Cioffi [aut, cre]

**Maintainer** William R. Cioffi <wrc14@duke.edu>

**Description** A set of functions and classes to assist with loading, manipulating, running diagnostics, and manipulating, Wildlife Computer sat tag output. Geared primarily toward mk10-\* tags and the data output obtained from the wildlife computers portal, but could be extended to be useful for other types of tags. Also includes functions for processing CLS Goniometer received messages back into a WC-DAP readable format. See <https://williamcioffi.github.io/goniomonitor> for a shiny app to visualize CLS Goniometer data in the field.

**Depends** R (>= 3.4.4)

**License** GPL-3

**Encoding** UTF-8

**URL** <https://github.com/williamcioffi/sattagutils>

**BugReports** <https://github.com/williamcioffi/sattagutils/issues>

**LazyData** true

**Collate** 'batch\_load\_tags.R'  
'censor\_beh.R'  
'es4dataframe.R'  
'sattagstream.R'  
'sattagstream\_extends.R'  
'date2num.R'  
'dateseq.R'  
'duplicated\_sattagstream.R'  
'has\_stream.R'  
'isfunctions.R'  
'latlond.hav.R'  
'load\_dap\_output.R'  
'load\_tag.R'  
'matchtimes.R'  
'merge\_stacks.R'  
'num2date.R'  
'paginate\_series.R'  
'plot\_series.R'

```

'rcsv.R'
'resample_ser.R'
'sattag.R'
'sattag_init.R'
'sattag_show.R'
'sattagstream_show.R'
'sattagutils.R'
'ser2beh.R'
'sort_by_message.R'
'tagstack.R'
'torad.R'
'wch_html2df.R'
'wcsv.R'

```

**RoxygenNote** 7.1.0

## R topics documented:

all-class . . . . .	3
argos-class . . . . .	4
as.data.frame,es4dataframe-method . . . . .	4
batch_load_tags . . . . .	5
behavior-class . . . . .	6
censor_beh . . . . .	6
corrupt-class . . . . .	7
date2num . . . . .	7
dateseq . . . . .	9
DeployID . . . . .	9
duplicated_sattagstream . . . . .	10
es4dataframe . . . . .	11
es4dataframe-class . . . . .	12
fastgps-class . . . . .	13
filename . . . . .	13
getstream . . . . .	14
has_stream . . . . .	15
histos-class . . . . .	15
initialize,sattag-method . . . . .	15
instrument . . . . .	16
is.es4dataframe . . . . .	16
isfunctions . . . . .	17
labels-class . . . . .	17
latlond.hav . . . . .	18
loadtime . . . . .	18
load_dap_output . . . . .	19
load_tag . . . . .	20
location . . . . .	20
locations-class . . . . .	21
matchtimes . . . . .	21
merge,es4dataframe-method . . . . .	22
merge_stacks . . . . .	22
minmaxdepth-class . . . . .	23
num2date . . . . .	23

paginate_series . . . . .	24
plot_series . . . . .	25
Ptt . . . . .	26
rawargos-class . . . . .	26
rsvg . . . . .	27
resample_ser . . . . .	27
rtc-class . . . . .	28
sattag . . . . .	28
sattag-class . . . . .	29
sattagstream . . . . .	29
sattagstream-class . . . . .	30
ser2beh . . . . .	30
series-class . . . . .	31
seriesrange-class . . . . .	32
show,sattag-method . . . . .	32
show,sattagstream-method . . . . .	32
show,tagstack-method . . . . .	33
sort_by_message . . . . .	33
species . . . . .	33
sst-class . . . . .	34
status-class . . . . .	34
streamtype . . . . .	35
summary-class . . . . .	35
tagdir . . . . .	36
tagstack . . . . .	36
tagstack-class . . . . .	37
tagstackdir . . . . .	37
tag_en . . . . .	38
tag_en<- . . . . .	38
tag_st . . . . .	39
tag_st<- . . . . .	39
torad . . . . .	40
wch_html2df . . . . .	40
wcsv . . . . .	41
[,es4dataframe-method . . . . .	41
[,sattag-method . . . . .	42
[,tagstack-method . . . . .	42
[<-,es4dataframe-method . . . . .	42
\$.es4dataframe-method . . . . .	42

**Index****43**

all-class

*an S4 class which extends sattagstream***Description**

corresponds to the \*-All.csv data stream.

**See Also**[sattag-class](#)[sattagstream-class](#)

Other sattagstream types: [argos-class](#), [behavior-class](#), [corrupt-class](#), [fastgps-class](#), [histos-class](#), [labels-class](#), [locations-class](#), [minmaxdepth-class](#), [rawargos-class](#), [rtc-class](#), [series-class](#), [seriesrange-class](#), [sst-class](#), [status-class](#), [summary-class](#)

---

[argos-class](#)*an S4 class which extends sattagstream*

---

**Description**

corresponds to the \*-Argos.csv data stream.

**See Also**[sattag-class](#)[sattagstream-class](#)

Other sattagstream types: [all-class](#), [behavior-class](#), [corrupt-class](#), [fastgps-class](#), [histos-class](#), [labels-class](#), [locations-class](#), [minmaxdepth-class](#), [rawargos-class](#), [rtc-class](#), [series-class](#), [seriesrange-class](#), [sst-class](#), [status-class](#), [summary-class](#)

---

[as.data.frame,es4dataframe-method](#)*convert es4dataframe back to data.frame*

---

**Description**

i use this to quickly grab the underlying data.frame mainly in order to pass it cleanly to S3 methods, but there are other applications.

**Usage**

```
## S4 method for signature 'es4dataframe'
as.data.frame(
  x,
  row.names = NULL,
  option = FALSE,
  ...,
  stringsAsFactors = FALSE
)
```

**Arguments**

<code>x</code>	an <code>es4dataframe</code>
<code>row.names</code>	defaults to <code>NULL</code>
<code>option</code>	boolean defaults to <code>FALSE</code>
<code>...</code>	all the other good stuff
<code>stringsAsFactors</code>	boolean defaults to <code>FALSE</code> because why would you want to do anything else? note that this is not the default behavior of an <code>as.data.frame</code> method.

**Value**

a `dataframe`

**See Also**

`es4dataframe`

**Examples**

```
df1 <- data.frame(a = rnorm(10), b = rnorm(10))
dfx <- es4dataframe(df1)
df2 <- as.data.frame(dfx)
identical(df1, df2)
```

---

<code>batch_load_tags</code>	<i>batch load sat tags</i>
------------------------------	----------------------------

---

**Description**

load sat tags from a directory containing directories downloaded from the wildlife computers portal.

**Usage**

```
batch_load_tags(
  data_dir,
  tags_dir = NA,
  ignore = NA,
  streams = NA,
  stream_delim = "-"
)
```

**Arguments**

<code>data_dir</code>	character points to a directory of sat tags
<code>tags_dir</code>	character vector of tag directory names to include or <code>NA</code> defaults to every directory in <code>data_dir</code>
<code>ignore</code>	character vector of directories to ignore or <code>NA</code> defaults not to ignore any directories.

streams	character vector limiting which streams to search for. NA defaults to all streams. note *-Summary.csv and *-Labels.csv are expected to populate some of the slots of sattag-class.
stream_delim	character defaults to "-". this is what the wildlife computers portal puts between the tag identifier (sometimes DeployID, sometimes Ptt) and the stream identifier (e.g., Argos, RTC, etc.) in the csv files.

**Value**

an S4 object of class `tagstack`

**See Also**

Other tag stream loaders: `load_tag()`

**Examples**

```
## Not run:
tags <- batch_load_tags("path/to/tags/")

## End(Not run)
```

---

behavior-class	<i>an S4 class which extends sattagstream</i>
----------------	---

---

**Description**

corresponds to the \*-Behavior.csv data stream.

**See Also**

`sattag-class`

`sattagstream-class`

Other sattagstream types: `all-class`, `argos-class`, `corrupt-class`, `fastgps-class`, `histos-class`, `labels-class`, `locations-class`, `minmaxdepth-class`, `rawargos-class`, `rtc-class`, `series-class`, `seriesrange-class`, `sst-class`, `status-class`, `summary-class`

---

censor_beh	<i>censor a behavior stream</i>
------------	---------------------------------

---

**Description**

based on depth and duration qualifications

**Usage**

```
censor_beh(b1, depth = 50, duration = 33 * 60)
```

**Arguments**

b1	a behavior data stream or a dataframe approximating one. requires columns DurationMin, DurationMax, What, Start, End, DepthMax, DepthMin.
depth	the minimum depth required to qualify as a dive
duration	the minimum duration required to qualify as a dive in seconds.

---

corrupt-class	<i>an S4 class which extends sattagstream</i>
---------------	---

---

**Description**

corresponds to the \*-Corrupt.csv data stream.

**See Also**

[sattag-class](#)

[sattagstream-class](#)

Other sattagstream types: [all-class](#), [argos-class](#), [behavior-class](#), [fastgps-class](#), [histos-class](#), [labels-class](#), [locations-class](#), [minmaxdepth-class](#), [rawargos-class](#), [rtc-class](#), [series-class](#), [seriesrange-class](#), [sst-class](#), [status-class](#), [summary-class](#)

---

date2num	<i>translate character dates to numeric dates</i>
----------	---

---

**Description**

convert character dates to numeric dates (seconds since UNIX epoch by default).

**Usage**

```
date2num(d, tz = "UTC", ...)

## S4 method for signature 'sattagstream'
date2num(d, tz = "UTC", format = "%H:%M:%S %d-%b-%Y", ...)

## S4 method for signature 'all'
date2num(d, tz = "UTC", format = "%m/%d/%Y %H:%M:%S", ...)

## S4 method for signature 'behavior'
date2num(d, tz = "UTC", format = "%H:%M:%S %d-%b-%Y", ...)

## S4 method for signature 'corrupt'
date2num(d, tz = "UTC", format = "%H:%M:%S %d-%b-%Y", ...)

## S4 method for signature 'fastgps'
date2num(d, tz = "UTC", format = "%H:%M:%S %d-%b-%Y", ...)

## S4 method for signature 'rawargos'
```

```

date2num(d, tz = "UTC", format = "%H:%M:%S %d-%b-%Y", ...)

## S4 method for signature 'rtc'
date2num(d, tz = "UTC", format = "%H:%M:%S %d-%b-%Y", ...)

## S4 method for signature 'series'
date2num(d, tz = "UTC", format = "%H:%M:%S %d-%b-%Y", ...)

## S4 method for signature 'seriesrange'
date2num(d, tz = "UTC", format = "%H:%M:%S %d-%b-%Y", ...)

## S4 method for signature 'locations'
date2num(d, tz = "UTC", format = "%H:%M:%S %d-%b-%Y", ...)

## S4 method for signature 'status'
date2num(d, tz = "UTC", format = "%H:%M:%S %d-%b-%Y", ...)

## S4 method for signature 'summary'
date2num(d, tz = "UTC", format = "%H:%M:%S %d-%b-%Y", ...)

```

### Arguments

<code>d</code>	either a character vector of dates or an object of a class which extends <a href="#">sattagstream</a> .
<code>tz, format</code>	as expected by <a href="#">as.POSIXct</a> .
<code>...</code>	any other good stuff you want to pass to <a href="#">as.POSIXct</a> .

### Details

methods are defined for converting objects of a class which extends `sattagstream`. note that the date formats and column names are inconsistent in the data streams download from the portal. for example, note the odd capitalization patterns in `*-All.csv` (and ambiguous date format). method is not fully implemented for `summary` because i've never seen the date format for `ReleaseDate` and `DeployDate`.

also, beware if you open any data stream csvs in excel and re-save, the dates will likely be put into an absurd ambiguous form. additionally, seconds tend to be obliterated.

this methods may become defunct if wildlife updates their conventions and will have to be updated.

### Methods (by class)

- `sattagstream`: for generic `sattagstreams`
- `all`: for `*-All.csv`. note the different time format.
- `behavior`: for `*-Behavior.csv`
- `corrupt`: for `*-Corrupt.csv`
- `fastgps`: for `*-FastGPS.csv`
- `rawargos`: for `*-RawArgos.csv`
- `rtc`: for `*-RTC.csv`
- `series`: for `*-Series.csv`
- `seriesrange`: for `*-SeriesRange.csv`
- `locations`: for `*-Locations.csv`
- `status`: for `*-Status.csv`
- `summary`: for `*-Summary.csv` note: not sure how to implement `ReleaseDate` or `DeployDate`...



**See Also**[as.POSIXct](#)Other date manipulators: [num2date\(\)](#)**Examples**

```
date2num("1988-03-12 07:00:00", tz = "UTC", format = "%Y-%m-%d %H:%M:%S")
```

---

dateseq	<i>quickly create nice data sequences for plotting</i>
---------	--

---

**Description**

using numeric times (seconds since UNIX epoch) find where to put the daily and hourly tickmarks

**Usage**

```
dateseq(d, hours = FALSE)
```

**Arguments**

**d** a numeric vector of times

**hours** boolean defaults to FALSE. when TRUE will return hourly times.

**Details**

probably could make this more general, but don't need to right now.

**Examples**

```
dateseq(1:(5*24*60*60))
```

---

DeployID	<i>get DeployID</i>
----------	---------------------

---

**Description**

function to extract DeployID.

**Usage**

```
DeployID(x)
```

## S4 method for signature 'sattag'

```
DeployID(x)
```

## S4 method for signature 'tagstack'

```
DeployID(x)
```

**Methods (by class)**

- sattag: method for sattag
- tagstack: method for tagstack

**See Also**

Other slot access functions: [Ptt\(\)](#), [filename\(\)](#), [getstream\(\)](#), [instrument\(\)](#), [loadtime\(\)](#), [location\(\)](#), [species\(\)](#), [streamtype\(\)](#), [tag\\_en\(\)](#), [tag\\_st\(\)](#), [tagdir\(\)](#), [tagstackdir\(\)](#)

---

duplicated\_sattagstream

*find duplicates in a stream*

---

**Description**

find duplicates aware that some columns might be different

**Usage**

```
duplicated_sattagstream(stream, use_cols = colnames(stream))

## S4 method for signature 'sattagstream'
duplicated_sattagstream(stream, use_cols = colnames(stream))

## S4 method for signature 'all'
duplicated_sattagstream(stream, use_cols = colnames(stream))

## S4 method for signature 'behavior'
duplicated_sattagstream(
  stream,
  use_cols = c("Ptt", "Start", "End", "What", "Number", "Shape", "DepthMin",
    "DepthMax", "DurationMin", "DurationMax", "Shallow", "Deep")
)

## S4 method for signature 'fastgps'
duplicated_sattagstream(
  stream,
  use_cols = c("Day", "Time", "LocNumber", "Failures", "Hauled.Out", "Satellites",
    "Range.Bits", "Id", "Range", "Signal", "Doppler", "CNR", "Id.1", "Range.1",
    "Signal.1", "Doppler.1", "CNR.1", "Id.2", "Range.2", "Signal.2", "Doppler.2",
    "CNR.2", "Id.3", "Range.3", "Signal.3", "Doppler.3", "CNR.3", "Id.4", "Range.4",
    "Signal.4", "Doppler.4", "CNR.4", "Id.5", "Range.5", "Signal.5", "Doppler.5",
    "CNR.5", "Id.6", "Range.6", "Signal.6", "Doppler.6", "CNR.6", "Id.7", "Range.7",
    "Signal.7", "Doppler.7", "CNR.7", "Id.8", "Range.8", "Signal.8", "Doppler.8",
    "CNR.8", "Id.9", "Range.9", "Signal.9", "Doppler.9", "CNR.9", "Id.10", "Range.10",
    "Signal.10", "Doppler.10", "CNR.10", "Id.11", "Range.11", "Signal.11", "Doppler.11",
    "CNR.11", "Id.12", "Range.12", "Signal.12", "Doppler.12", "CNR.12", "Id.13",
    "Range.13", "Signal.13", "Doppler.13", "CNR.13", "Id.14", "Range.14", "Signal.14",
    "Doppler.14", "CNR.14")
)
```

```
## S4 method for signature 'series'
duplicated_sattagstream(
  stream,
  use_cols = c("Ptt", "Day", "Time", "Depth", "DRange", "Temperature", "TRange",
    "Activity", "ARange")
)

## S4 method for signature 'seriesrange'
duplicated_sattagstream(
  stream,
  use_cols = c("Ptt", "Start", "End", "MinDepth", "MinDepthAccuracy", "MaxDepth",
    "MaxDepthAccuracy", "MinTemp", "MinTempAccuracy", "MaxTemp", "MaxTempAccuracy",
    "MobMean", "MobSD", "ActivitySum")
)
```

### Arguments

stream	a sattagstream to test for duplicates
use_cols	a character vector of column names or a integer vector of indices to use to generate a key for duplicated.

### Functions

- duplicated\_sattagstream,sattagstream-method: for generic sattagstreams
- duplicated\_sattagstream,all-method: for \*-All.csv
- duplicated\_sattagstream,behavior-method: for \*-Behavior.csv
- duplicated\_sattagstream,fastgps-method: for \*-FastGPS.csv
- duplicated\_sattagstream,series-method: for \*-Series.csv
- duplicated\_sattagstream,seriesrange-method: for \*-SeriesRange.csv

---

es4dataframe	<i>es4dataframe constructor</i>
--------------	---------------------------------

---

### Description

this would be the typical way to instantiate an es4dataframe, an S4 wrapper for data.frame.

### Usage

```
es4dataframe(..., stringsAsFactors = FALSE)
```

### Arguments

...	all that good regular data.frame stuff
stringsAsFactors	boolean defaults to FALSE because why would you do anything else?

**Details**

a well behaving S4 `data.frame` wrapper. i hope what you expect to happen happens. my main concern was that subsetting functions should return an S4 class instead of a `data.frame`.

i've implemented methods for `as.data.frame`, `$`, `[`, `[<`, and `merge`. these methods are just simple wrappers for the S3 methods.

note that `merge` will have to be implemented in any class that extends `es4dataframe` and has slots so that these functions know how to merge those slots.

with the defaults and those definitions many methods work without fuss and return the S4 object including `subset`, `na.omit`, `unique`, `$<-`.

other methods you don't expect to return the S4 class anyway work fine like: `duplicated`, `as.list`, `[`, etc.

i think it makes sense to let some methods default to returning the underlying `data.frame` only and not the S4 class. for example, `edit`. in the applications i have in mind, i don't really want the user to be able to edit the S4 classes data 'by hand'. and if you want that functionality you can always extend the class and add it.

i haven't implemented `cbind` or `rbind` because i can't figure out how to do it. they'd also have to be implemented in any class that extends `es4dataframe` with slots.

**Value**

an S4 class of `es4dataframe`.

**See Also**

[es4dataframe-class](#)

[as.data.frame-es4dataframe-method](#)

**Examples**

```
testdf <- es4dataframe(a = rnorm(10), b = rnorm(10))
```

---

es4dataframe-class	<i>an S4 data.frame</i>
--------------------	-------------------------

---

**Description**

call me hadley.

**See Also**

[es4dataframe](#)

---

fastgps-class	<i>an S4 class which extends sattagstream</i>
---------------	---

---

### Description

corresponds to the \*-FastGPS.csv data stream. this is not a well formatted csv when downloaded from the portal so beware when importing. see [load\\_tag](#).

### See Also

[sattag-class](#)

[sattagstream-class](#)

Other sattagstream types: [all-class](#), [argos-class](#), [behavior-class](#), [corrupt-class](#), [histos-class](#), [labels-class](#), [locations-class](#), [minmaxdepth-class](#), [rawargos-class](#), [rtc-class](#), [series-class](#), [seriesrange-class](#), [sst-class](#), [status-class](#), [summary-class](#)

---

filename	<i>get filename of a sattagstream</i>
----------	---------------------------------------

---

### Description

use this function to get the filename of a [sattagstream-class](#). changing the source filename after construction is currently not supported.

### Usage

```
filename(x)

## S4 method for signature 'sattagstream'
filename(x)

## S4 method for signature 'sattag'
filename(x)

## S4 method for signature 'tagstack'
filename(x)
```

### Arguments

x                      source filename for a csv data stream file.

### Methods (by class)

- `sattagstream`: get the source filename of a `sattagstream`
- `sattag`: return a vector of source filenames of all `sattagstreams` contained in a `sattag`
- `tagstack`: return a vector of source filenames of all `sattagstreams` contained in a `tagstack`

**See Also**

[sattagstream](#)

Other slot access functions: [DeployID\(\)](#), [Ptt\(\)](#), [getstream\(\)](#), [instrument\(\)](#), [loadtime\(\)](#), [location\(\)](#), [species\(\)](#), [streamtype\(\)](#), [tag\\_en\(\)](#), [tag\\_st\(\)](#), [tagdir\(\)](#), [tagstackdir\(\)](#)

---

getstream	<i>get a stream</i>
-----------	---------------------

---

**Description**

function to get a particular streamtype from a [sattag](#) or [tagstack](#).

**Usage**

```
getstream(x, type, squash = FALSE)

## S4 method for signature 'sattag'
getstream(x, type, squash = FALSE)

## S4 method for signature 'tagstack'
getstream(x, type, squash = FALSE)
```

**Arguments**

type	streamtype to extract
squash	boolean defaults to FALSE. If TRUE, will return a single sattagstream.

**Value**

either a tagstack or sattag depending on input.

**Methods (by class)**

- sattag: return all streams of streamtype type
- tagstack: return a tagstack of all streams of type

**See Also**

Other slot access functions: [DeployID\(\)](#), [Ptt\(\)](#), [filename\(\)](#), [instrument\(\)](#), [loadtime\(\)](#), [location\(\)](#), [species\(\)](#), [streamtype\(\)](#), [tag\\_en\(\)](#), [tag\\_st\(\)](#), [tagdir\(\)](#), [tagstackdir\(\)](#)

---

has_stream	<i>has_stream</i>
------------	-------------------

---

**Description**

Checks to see if a tagstack or sattag has a stream

**Usage**

```
has_stream(x, streamname)
```

**Arguments**

x	the tagstack or sattag in question
streamname	the stream you are searching for

**Value**

a bool if any of the streams are streamname

---

histos-class	<i>an S4 class which extends sattagstream</i>
--------------	---

---

**Description**

corresponds to the \*-Histos.csv data stream.

**See Also**

[sattag-class](#)

[sattagstream-class](#)

Other sattagstream types: [all-class](#), [argos-class](#), [behavior-class](#), [corrupt-class](#), [fastgps-class](#), [labels-class](#), [locations-class](#), [minmaxdepth-class](#), [rawargos-class](#), [rtc-class](#), [series-class](#), [seriesrange-class](#), [sst-class](#), [status-class](#), [summary-class](#)

---

initialize,sattag-method	<i>initialize an S4 object of class <a href="#">sattag</a></i>
--------------------------	--

---

**Description**

initialize an S4 object of class [sattag](#)

**Usage**

```
## S4 method for signature 'sattag'
initialize(Object, ...)
```

---

instrument	<i>get instrument</i>
------------	-----------------------

---

### Description

function to extract instrument type (e.g., SPOT6, MK10-A, etc).

### Usage

```
instrument(x)

## S4 method for signature 'sattag'
instrument(x)

## S4 method for signature 'tagstack'
instrument(x)
```

### Methods (by class)

- sattag: method for sattag
- tagstack: method for tagstack

### See Also

Other slot access functions: [DeployID\(\)](#), [Ptt\(\)](#), [filename\(\)](#), [getstream\(\)](#), [loadtime\(\)](#), [location\(\)](#), [species\(\)](#), [streamtype\(\)](#), [tag\\_en\(\)](#), [tag\\_st\(\)](#), [tagdir\(\)](#), [tagstackdir\(\)](#)

---

is.es4dataframe	<i>is function for es4dataframe</i>
-----------------	-------------------------------------

---

### Description

check to see if an object extends [es4dataframe](#)

### Usage

```
is.es4dataframe(obj)
```



---

isfunctions	<i>is functions for sattag and sattagstream</i>
-------------	---

---

## Description

check to see if an object is a [sattag](#) or [sattagstream](#)

## Usage

```
is.sattag(obj)
```

```
is.sattagstream(obj)
```

## Examples

```
## Not run:
tag1 <- load_tag("path/to/tags/tag1")
s1 <- tag1[[1]]

c(is.stream(s1), is.tag(s1), is.sattag(tag1), is.stream(tag1))

## End(Not run)
```

---

labels-class	<i>an S4 class which extends sattagstream</i>
--------------	---

---

## Description

corresponds to the \*-Labels.csv data stream. this csv file appears to be missing an EOF when downloaded from the portal so beware when importing. see [load\\_tag](#).

## See Also

[sattag-class](#)

[sattagstream-class](#)

Other sattagstream types: [all-class](#), [argos-class](#), [behavior-class](#), [corrupt-class](#), [fastgps-class](#), [histos-class](#), [locations-class](#), [minmaxdepth-class](#), [rawargos-class](#), [rtc-class](#), [series-class](#), [seriesrange-class](#), [sst-class](#), [status-class](#), [summary-class](#)

---

latlond.hav	<i>calculate the haversine distance between two geographic coordiantes</i>
-------------	--

---

**Description**

calculate the haversine distance between two geographic coordiantes

**Usage**

```
latlond.hav(lat1, lon1, lat2, lon2)
```

**Arguments**

lat1, lon1	start coordinates (can be vectors)
lat2, lon2	end coordinates (can be vectors)

**Details**

uses 6371 as the radius of the earth in kilometers.

**Value**

distance in kilometers

**References**

<https://www.movable-type.co.uk/scripts/latlong.html>

**See Also**

Other distance functions: [torad\(\)](#)

**Examples**

```
latlond.hav(-6.72, 147, -4.67, -174.52)
```

---

loadtime	<i>get loadtime</i>
----------	---------------------

---

**Description**

function to extract time a tag was loaded into R originally based on when intialize() was called.

**Usage**

```
loadtime(x)

## S4 method for signature 'sattag'
loadtime(x)

## S4 method for signature 'tagstack'
loadtime(x)
```

**Methods (by class)**

- sattag: method for sattag
- tagstack: method for tagstack

**See Also**

Other slot access functions: [DeployID\(\)](#), [Ptt\(\)](#), [filename\(\)](#), [getstream\(\)](#), [instrument\(\)](#), [location\(\)](#), [species\(\)](#), [streamtype\(\)](#), [tag\\_en\(\)](#), [tag\\_st\(\)](#), [tagdir\(\)](#), [tagstackdir\(\)](#)

---

load_dap_output	<i>load dap output</i>
-----------------	------------------------

---

**Description**

load a directory of output csv's created by Argos\ Message\ Decoder.exe into a sane tagstack.

**Usage**

```
load_dap_output(data_dir, stream_delim = "-")
```

**Arguments**

data_dir	the directory path to your data
stream_delim	a character which defaults to "-". This is what Argos\ Message\ Decoder.exe puts between the tag identifier and the stream name in the csv filenames.

**Details**

This functions expects there to be multiple tags concatenated into the same csv files and so returns a tagstack. This isn't a problem if there is only one tag, but it will since return a tagstack of length of 1. You can always un-nest it later. Basically this function just calls [load\\_tag](#) and then deals with the fallout to get things into a nice tagstack. This function also expects Ptt's to be unique, which should be the case for a batch of tags running at the same time, but isn't necessarily true for all time. Nevertheless, the user can't necessarily be relied upon to always have DeployID set so perhaps this is the most reasonable first pass?

**Value**

an S4 object of class [tagstack](#)

**Examples**

```
## Not run:
tag <- load_dap_output("path/to/dap/csvs/")

## End(Not run)
```

---

load_tag	<i>load a single sat tag</i>
----------	------------------------------

---

### Description

given a tag directory from a portal download instantiate, populate, and return a [sattag](#) S4 object.

### Usage

```
load_tag(tag_dir, streams = NA, stream_delim = "-")
```

### Arguments

tag_dir	a path to a tag directory containing csv data streams downloaded from the portal.
streams	a character vector limiting which streams to search for. NA default to all streams. note that *-Summary.csv is expected to populate some of the slots.
stream_delim	character defaults to "-". this is what the wildlife computers portal puts between the tag identifier (sometimes DeployID, sometimes Ptt) and the stream identifier (e.g., Argos, RTC, etc.) in the csv files.

### Value

a [sattag](#) S4 object.

### See Also

Other tag stream loaders: [batch\\_load\\_tags\(\)](#)

### Examples

```
## Not run:
tag1 <- load_tag("~/path/to/tags/tag1")

## End(Not run)
```

---

location	<i>get location</i>
----------	---------------------

---

### Description

function to extract study location as defined in \*-Labels.csv.

### Usage

```
location(x)

## S4 method for signature 'sattag'
location(x)

## S4 method for signature 'tagstack'
location(x)
```

**Methods (by class)**

- `sattag`: method for `sattag`
- `tagstack`: method for `tagstack`

**See Also**

Other slot access functions: `DeployID()`, `Ptt()`, `filename()`, `getstream()`, `instrument()`, `loadtime()`, `species()`, `streamtype()`, `tag_en()`, `tag_st()`, `tagdir()`, `tagstackdir()`

---

locations-class	<i>an S4 class which extends sattagstream</i>
-----------------	---

---

**Description**

corresponds to the \*-Locations.csv data stream.

**See Also**

`sattag-class`

`sattagstream-class`

Other `sattagstream` types: `all-class`, `argos-class`, `behavior-class`, `corrupt-class`, `fastgps-class`, `histos-class`, `labels-class`, `minmaxdepth-class`, `rawargos-class`, `rtc-class`, `series-class`, `seriesrange-class`, `sst-class`, `status-class`, `summary-class`

---

matchtimes	<i>a tiny time matching function</i>
------------	--------------------------------------

---

**Description**

a tiny time matching function

**Usage**

```
matchtimes(t1, t2)
```

**Arguments**

`t1`, `t2`                numeric time vectors.

**See Also**

`findInterval`.

**Examples**

```
t1 <- c(10, 20, 30, 40)
t2 <- c(9, 21, 36, 37)
matchtimes(t1, t2)
matchtimes(t2, t1)
```

---

```
merge, es4dataframe-method
```

*wrapper for S3 method*

---

### Description

wrapper for S3 method

### Usage

```
## S4 method for signature 'es4dataframe'
merge(
  x,
  y,
  by = intersect(names(x), names(y)),
  by.x = by,
  by.y = by,
  all = FALSE,
  all.x = all,
  all.y = all,
  sort = TRUE,
  suffixes = c(".x", ".y"),
  no.dups = TRUE,
  incomparables = NULL,
  ...
)
```

---

```
merge_stacks
```

*merge tagstacks*

---

### Description

sensibly merge two tagstacks together with overlapping sets of tags

### Usage

```
merge_stacks(
  target_stack,
  source_stack,
  by = "Ptt",
  remove_duplicates = FALSE,
  identify_original = FALSE,
  target_lab = "target",
  source_lab = "source"
)
```

**Arguments**

by	the name of the sattag slot to base the merge on. Should be "Ptt" or "DeployID". Defaults to "Ptt".
remove_duplicates	a boolean defaults to FALSE. If TRUE, duplicated rows of data will be retained in the output tagstack.
identify_original	a boolean defaults to FALSE. If TRUE, a character 'original' column will be added to each sattag stream indicating the origin of each row with a character identifier specified by target_lab or source_lab.
target_lab	a string to identify data that came from the target tagstack.
source_lab	a string to identify data that came from the source tagstack.
target	a tagstack, the target stack
source	a tagstack, the source stack

**Value**

an S4 object of class [tagstack](#).

---

minmaxdepth-class	<i>an S4 class which extends sattagstream</i>
-------------------	---

---

**Description**

corresponds to the \*-MinMaxDepth.csv data stream.

**See Also**

[sattag-class](#)

[sattagstream-class](#)

Other sattagstream types: [all-class](#), [argos-class](#), [behavior-class](#), [corrupt-class](#), [fastgps-class](#), [histos-class](#), [labels-class](#), [locations-class](#), [rawargos-class](#), [rtc-class](#), [series-class](#), [seriesrange-class](#), [sst-class](#), [status-class](#), [summary-class](#)

---

num2date	<i>translate numeric dates back to POSIX*</i>
----------	---

---

**Description**

will convert a numeric vector (seconds since UNIX epoch) using as.POSIXct. methods for [sattagstream](#) classes not implemeneted.

**Usage**

```
num2date(d, tz = "UTC", origin = "1970-01-01", ...)
```

**Arguments**

d                      a numeric vector of dates.  
 tz, origin            as expected by [as.POSIXct](#).

**See Also**

[as.POSIXct](#)

Other date manipulators: [date2num\(\)](#)

**Examples**

```
num2date(574153200)
```

---

paginate_series	<i>a pagination tool for plotting series datastream.</i>
-----------------	--

---

**Description**

simple function which provides a menu navigation for seires datastream

**Usage**

```
paginate_series(  
  s,  
  increment = 4 * 48,  
  time = "samples",  
  xaxt = "n",  
  las = 1,  
  ...  
)
```

**Arguments**

s                      a series data stream or a dataframe approximating one. gets passed to [plot\\_series](#).  
 increment            the number of sampling points to show on the plot at one time. also current increments without overlap when you page forward or backwards.  
 time                  a string which indicates what you want the x-axis to be. right now only "samples" (running sample index) or "numeric" (matlab style datenum since the unix epoch) are supported.  
 xaxt                  default set to 'n' so i can plot an axis like you asked for in time.  
 las                   default to 1 why would you do anything else?  
 ...                   all that other good graphics stuff and gets passed to [plot\\_series](#).



---

plot_series	<i>plotting function for series datastream.</i>
-------------	---

---

## Description

simple function to plot series data quickly.

## Usage

```
plot_series(
  s,
  show_gaps = TRUE,
  new = TRUE,
  points = TRUE,
  lines = TRUE,
  pch = 16,
  cex = 0.5,
  col.lines = "black",
  col.points = "black",
  col.gaps = col.lines,
  ylim,
  gap_plotting_buffer = 0.15,
  ylab = "Depth (meters)",
  xlab = "",
  axes = TRUE,
  ...
)
```

## Arguments

s	a series data stream or a dataframe approximating one. requires columns numeric Date, numeric Depth.right now also requires to have at least 2 rows that is so i can calculate the sampling period. see readme for more information about this.
show_gaps	boolean defaults to TRUE. If TRUE, will print blocks at the top of the plotting area to indicate data gaps.
new	boolean defaults to TRUE. draw a new plot if this is TRUE otherwise draw on whatever you have up.
points	boolean defaults to TRUE. draw points.
lines	boolean defaults to TRUE. draw lines.
pch, cex	just reasonable defaults do whatever you want.
col.lines	color of the lines if drawn.
col.points	color of the points if drawn.
...	all the other good plotting stuff. gets passed to plot, points, and lines so think about it.

---

Ptt	<i>get Ptt</i>
-----	----------------

---

### Description

function to extract Ptt.

### Usage

```
Ptt(x)

## S4 method for signature 'sattag'
Ptt(x)

## S4 method for signature 'tagstack'
Ptt(x)
```

### Value

a character representation of the Ptt. why this and not numeric?

### Methods (by class)

- sattag: method for sattag
- tagstack: method for tagstack

### See Also

Other slot access functions: [DeployID\(\)](#), [filename\(\)](#), [getstream\(\)](#), [instrument\(\)](#), [loadtime\(\)](#), [location\(\)](#), [species\(\)](#), [streamtype\(\)](#), [tag\\_en\(\)](#), [tag\\_st\(\)](#), [tagdir\(\)](#), [tagstackdir\(\)](#)

---

rawargos-class	<i>an S4 class which extends sattagstream</i>
----------------	---

---

### Description

corresponds to the \*-RawArgos.csv data stream. this is not a well formatted csv when downloaded from the portal so beware when importing. see [load\\_tag](#).

### See Also

[sattag-class](#)  
[sattagstream-class](#)

Other sattagstream types: [all-class](#), [argos-class](#), [behavior-class](#), [corrupt-class](#), [fastgps-class](#), [histos-class](#), [labels-class](#), [locations-class](#), [minmaxdepth-class](#), [rtc-class](#), [series-class](#), [seriesrange-class](#), [sst-class](#), [status-class](#), [summary-class](#)

---

rcsv	<i>sensible defaults for read.table</i>
------	---

---

**Description**

sensible defaults for read.table

**Usage**

```
rcsv(..., header = TRUE, sep = ",", stringsAsFactors = FALSE)
```

**Arguments**

...	all the regular good stuff for finding files and whatnot.
header	i always want a header
sep	we're talking about comma seperated values here.
stringsAsFactors	default is FALSE. please don't change that. why would you change that?

**Details**

please don't use read\_csv or any of that other nonsense, it breaks everyone's code...

**See Also**

Other sensible csv functions: [wcsv\(\)](#)

**Examples**

```
# use it just like read.table but without the worry
## Not run:
rcsv("file.csv")

## End(Not run)
```

---

resample_ser	<i>resample series</i>
--------------	------------------------

---

**Description**

downsample series to a particular sampling period.

**Usage**

```
resample_ser(s, sampling_period)
```

**Arguments**

s	a series data stream or a dataframe approximating one. requires column numeric Date.
sampling_period	desired sampling period in minutes (sorry).

---

rtc-class	<i>an S4 class which extends sattagstream</i>
-----------	---

---

### Description

corresponds to the \*-RTC.csv data stream.

### See Also

[sattag-class](#)

[sattagstream-class](#)

Other sattagstream types: [all-class](#), [argos-class](#), [behavior-class](#), [corrupt-class](#), [fastgps-class](#), [histos-class](#), [labels-class](#), [locations-class](#), [minmaxdepth-class](#), [rawargos-class](#), [series-class](#), [seriesrange-class](#), [sst-class](#), [status-class](#), [summary-class](#)

---

sattag	<i>constructor for sattag</i>
--------	-------------------------------

---

### Description

use this constructor to create a new sattag object.

### Usage

```
sattag(
  data = list(),
  instrument = character(),
  DeployID = character(),
  Ptt = character(),
  species = character(),
  location = character(),
  t_start = numeric(),
  t_end = numeric(),
  directory = character()
)
```

### Arguments

data	this should be a list of <a href="#">sattagstreams-class</a> .
instrument	instrument type, e.g., "MK10-A"
DeployID	character vector
Ptt	character vector
species	character vector
location	character vector of the study site name.
t_start, t_end	numeric dates (seconds since UNIX epoch).
directory	the original directory from which the tag was loaded.

**Value**

a [sattag-class](#)

---

sattag-class	<i>an S4 class to represent a single sat tag</i>
--------------	--

---

**Description**

represents a single sat tag. holds a small amount of meta data and a list of data streams (extends [list](#)).

**Slots**

instrument instrument type, e.g., "MK10-A"  
 DeployID character vector  
 Ptt character vector  
 species character vector  
 location character vector of the study site name.  
 earliestdata,latestdata numeric dates (seconds since UNIX epoch).  
 directory the original directory from which the tag was loaded.  
 loadtime a character vector set by initialize() when object is instantiated.

---

sattagstream	<i>constructor for sattagstreams</i>
--------------	--------------------------------------

---

**Description**

use this constructor to assign the correct subclass (e.g., location, argos, behavior, etc.) to a new sattagstream object.

**Usage**

```
sattagstream(type = character(), data = data.frame(), filename = character())
```

**Arguments**

type	character of stream type. subclasses exist for: c("all", "argos", "behavior", "corrupt", "fastgp") anything else will become a generic sattagstream. if you want to add a stream type which requires special methods then you'll have to add it and write them. if you don't need the special methods the generic sattagstream will work fine.
data	this is a data frame from a wildlife computer portal downloaded csv data file.
filename	the name of the file the data originally came from.

**Details**

i don't really expect you to use this function very often, but if you do want to make a stream by hand this is the preferred method. if you are importing streams from an existing tag you probably should be by using [load\\_tag](#) or [batch\\_load\\_tags](#) to load a directory downloaded from the portal.

**Value**

an S4 object of the class `sattagstream-class` (or an extension of it)

**Examples**

```
## Not run:
argos.df <- rcsv("PTT-Argos.csv")
sattagstream("argos", argos.df)

## End(Not run)
```

---

<code>sattagstream-class</code>	<i>an S4 class to represent a single data stream in a tag</i>
---------------------------------	---

---

**Description**

a parent class for specific sattag stream types. generally accessed from within a `sattag`.

**Slots**

`streamtype` character of the stream type for convenience (e.g., `argos`, `behavior`, etc.).

`filename` the original filename from which the data stream was derived. this will almost always be a text or csv file.

**See Also**

extends `es4dataframe`.

---

<code>ser2beh</code>	<i>downsample series data to faux behavior data</i>
----------------------	---

---

**Description**

a simple downsampling which takes series data stream as an input and creates an estimate of what the behavior stream would have looked like. tries to interpolate surfacing times for more accurate duration estimates.

**Usage**

```
ser2beh(
  s,
  surface_threshold_meters = 25,
  vrate_ascent_meters_per_second = 1.4,
  vrate_decent_meters_per_second = 1.4,
  dive_definition_threshold_meters = 50,
  period
)
```

## Arguments

- `s` a series data stream or a dataframe approximating one. requires columns numeric Date, numeric Depth, numeric DRange.
- `vrate_ascent_meters_per_second,`  
`vrate_decent_meters_per_second` these are the vertical ascent and decent rates used to interpolate surfacing times. defaults are for *Ziphius cavirostris* (see notes).
- `dive_definition_threshold_meters`  
would be from the behavior settings you want to emulate. this is the threshold to qualify as a behavior dive.
- `period` sampling period of input series data
- `surface_threshold_meters` used  
by the peak finding algorithm to determine if a peak is close enough to the surface to represent a real surfacing event. default 25 is for me. you will need to pick something that makes sense for your species.

## Note

i've set the vertical ascent and decent rates the same for *Ziphius cavirostris* (1.4 m/s). This is based on some experimentation with known surfacings from behavior data, vertical rates from series data, and analysis of dtags by Tyack et al. (2006). Though the ascent rates are much slower for *Z. cavirostris* after deep dives than the decent rates, there is actually no difference in the last several hundred meters and the rate is much faster.

## References

Tyack, P. L., Johnson, M., Soto, N. A., Sturlese, A., & Madsen, P. T. (2006). Extreme diving of beaked whales. *Journal of Experimental Biology*, 209(21), 4238–4253. <https://doi.org/10.1242/jeb.02505>

---

<code>series-class</code>	<i>an S4 class which extends <code>sattagstream</code></i>
---------------------------	--

---

## Description

corresponds to the \*-Series.csv data stream.

## See Also

[sattag-class](#)

[sattagstream-class](#)

Other `sattagstream` types: [all-class](#), [argos-class](#), [behavior-class](#), [corrupt-class](#), [fastgps-class](#), [histos-class](#), [labels-class](#), [locations-class](#), [minmaxdepth-class](#), [rawargos-class](#), [rtc-class](#), [seriesrange-class](#), [sst-class](#), [status-class](#), [summary-class](#)

---

seriesrange-class	<i>an S4 class which extends sattagstream</i>
-------------------	---

---

### Description

corresponds to the \*-SeriesRange.csv data stream.

### See Also

[sattag-class](#)

[sattagstream-class](#)

Other sattagstream types: [all-class](#), [argos-class](#), [behavior-class](#), [corrupt-class](#), [fastgps-class](#), [histos-class](#), [labels-class](#), [locations-class](#), [minmaxdepth-class](#), [rawargos-class](#), [rtc-class](#), [series-class](#), [sst-class](#), [status-class](#), [summary-class](#)

---

show,sattag-method	<i>display an S4 object of class <a href="#">sattag</a></i>
--------------------	---

---

### Description

display an S4 object of class [sattag](#)

### Usage

```
## S4 method for signature 'sattag'
show(object)
```

---

show,sattagstream-method	<i>display an S4 object which extends sattagstream</i>
--------------------------	--

---

### Description

generic method for show.

### Usage

```
## S4 method for signature 'sattagstream'
show(object)
```



---

show,tagstack-method	<i>show tagstack</i>
----------------------	----------------------

---

**Description**

show tagstack

**Usage**

```
## S4 method for signature 'tagstack'
show(object)
```

---

sort_by_message	<i>sort behavior streams</i>
-----------------	------------------------------

---

**Description**

sort behavior streams by start time keeping messages together

**Usage**

```
sort_by_message(beh)
```

**Arguments**

beh                      a behavior sattagstream.

**Value**

a behavior sattagstream sorted.

---

species	<i>get species</i>
---------	--------------------

---

**Description**

function to extract species as defined in \*-Labels.csv.

**Usage**

```
species(x)
```

```
## S4 method for signature 'sattag'
species(x)
```

```
## S4 method for signature 'tagstack'
species(x)
```

**Methods (by class)**

- `sattag`: method for `sattag`
- `tagstack`: method for `tagstack`

**See Also**

Other slot access functions: [DeployID\(\)](#), [Ptt\(\)](#), [filename\(\)](#), [getstream\(\)](#), [instrument\(\)](#), [loadtime\(\)](#), [location\(\)](#), [streamtype\(\)](#), [tag\\_en\(\)](#), [tag\\_st\(\)](#), [tagdir\(\)](#), [tagstackdir\(\)](#)

---

<code>sst-class</code>	<i>an S4 class which extends <code>sattagstream</code></i>
------------------------	--

---

**Description**

corresponds to the `*-SST.csv` data stream.

**See Also**

[sattag-class](#)

[sattagstream-class](#)

Other `sattagstream` types: [all-class](#), [argos-class](#), [behavior-class](#), [corrupt-class](#), [fastgps-class](#), [histos-class](#), [labels-class](#), [locations-class](#), [minmaxdepth-class](#), [rawargos-class](#), [rtc-class](#), [series-class](#), [seriesrange-class](#), [status-class](#), [summary-class](#)

---

<code>status-class</code>	<i>an S4 class which extends <code>sattagstream</code></i>
---------------------------	--

---

**Description**

corresponds to the `*-Status.csv` data stream.

**See Also**

[sattag-class](#)

[sattagstream-class](#)

Other `sattagstream` types: [all-class](#), [argos-class](#), [behavior-class](#), [corrupt-class](#), [fastgps-class](#), [histos-class](#), [labels-class](#), [locations-class](#), [minmaxdepth-class](#), [rawargos-class](#), [rtc-class](#), [series-class](#), [seriesrange-class](#), [sst-class](#), [summary-class](#)

---

streamtype	<i>get stream type of a sattagstream</i>
------------	--

---

## Description

use this function to get the stream type of a [sattagstream-class](#). changing the stream type after construction is currently not supported.

## Usage

```
streamtype(x)

## S4 method for signature 'sattagstream'
streamtype(x)

## S4 method for signature 'sattag'
streamtype(x)

## S4 method for signature 'tagstack'
streamtype(x)
```

## Arguments

x                      stream type.

## Methods (by class)

- `sattagstream`: get the streamtype of a `sattagstream`
- `sattag`: return a vector of stream types of all `sattagstreams` contained in a `sattag`
- `tagstack`: return a vector of stream types of all `sattagstreams` contained in a `tagstack`

## See Also

[sattagstream](#)

Other slot access functions: [DeployID\(\)](#), [Ptt\(\)](#), [filename\(\)](#), [getstream\(\)](#), [instrument\(\)](#), [loadtime\(\)](#), [location\(\)](#), [species\(\)](#), [tag\\_en\(\)](#), [tag\\_st\(\)](#), [tagdir\(\)](#), [tagstackdir\(\)](#)

---

summary-class	<i>an S4 class which extends sattagstream</i>
---------------	---

---

## Description

corresponds to the \*-Summary.csv data stream. used to populate slots in [sattag](#) within [load\\_tag](#).

**See Also**[sattag-class](#)[sattagstream-class](#)

Other sattagstream types: [all-class](#), [argos-class](#), [behavior-class](#), [corrupt-class](#), [fastgps-class](#), [histos-class](#), [labels-class](#), [locations-class](#), [minmaxdepth-class](#), [rawargos-class](#), [rtc-class](#), [series-class](#), [seriesrange-class](#), [sst-class](#), [status-class](#)

---

tagdir

---

*get tagdir*

---

**Description**

function to extract source directory from which the tag was loaded.

**Usage**

```
tagdir(x)
```

```
## S4 method for signature 'sattag'
tagdir(x)
```

```
## S4 method for signature 'tagstack'
tagdir(x)
```

**Methods (by class)**

- `sattag`: method for `sattag`
- `tagstack`: method for `tagstack`

**See Also**

Other slot access functions: [DeployID\(\)](#), [Ptt\(\)](#), [filename\(\)](#), [getstream\(\)](#), [instrument\(\)](#), [loadtime\(\)](#), [location\(\)](#), [species\(\)](#), [streamtype\(\)](#), [tag\\_en\(\)](#), [tag\\_st\(\)](#), [tagstackdir\(\)](#)

---

tagstack

---

*constructor for tagstack*

---

**Description**

use this constructor to create a new tagstack object.

**Usage**

```
tagstack(data = list(), directory = character())
```

**Arguments**

`data` this should be a list of [sattag-class](#).

**Value**

a [tagstack-class](#)

---

tagstack-class	<i>an S4 class to represent a stack of sat tags</i>
----------------	---

---

**Description**

represents a tag stack. wrapper for a [list](#).

**See Also**

[tagstack](#)

---

tagstackdir	<i>get tagstack source directory</i>
-------------	--------------------------------------

---

**Description**

function to extract the source directory for a tagstack

**Usage**

```
tagstackdir(x)
```

**Arguments**

a [tagstack](#)

**Value**

a character representation of the source directory

**See Also**

Other slot access functions: [DeployID\(\)](#), [Ptt\(\)](#), [filename\(\)](#), [getstream\(\)](#), [instrument\(\)](#), [loadtime\(\)](#), [location\(\)](#), [species\(\)](#), [streamtype\(\)](#), [tag\\_en\(\)](#), [tag\\_st\(\)](#), [tagdir\(\)](#)

---

tag_en	<i>get end time of tag</i>
--------	----------------------------

---

### Description

function to extract the data end time for a particular tag. if [load\\_tag](#) or [batch\\_load\\_tags](#) created the tag object then this is set from the LatestDateTime in the summary stream.

### Usage

```
tag_en(x)

## S4 method for signature 'sattag'
tag_en(x)

## S4 replacement method for signature 'sattag'
tag_en(x) <- value

## S4 method for signature 'tagstack'
tag_en(x)
```

### Methods (by class)

- sattag: method for sattag
- sattag: method for sattag
- tagstack: method for tagstack

### See Also

Other slot access functions: [DeployID\(\)](#), [Ptt\(\)](#), [filename\(\)](#), [getstream\(\)](#), [instrument\(\)](#), [loadtime\(\)](#), [location\(\)](#), [species\(\)](#), [streamtype\(\)](#), [tag\\_st\(\)](#), [tagdir\(\)](#), [tagstackdir\(\)](#)

---

tag_en<-	<i>set start time of tag</i>
----------	------------------------------

---

### Description

function to set the data end time for a particular tag. if [load\\_tag](#) or [batch\\_load\\_tags](#) created the tag object then this is initially set from the LatestDateTime in the summary stream.

### Usage

```
tag_en(x) <- value
```

---

tag_st	<i>get start time of tag</i>
--------	------------------------------

---

### Description

function to extract the data start time for a particular tag. if [load\\_tag](#) or [batch\\_load\\_tags](#) created the tag object then this is set from the EarliestDateTime in the summary stream.

### Usage

```
tag_st(x)

## S4 method for signature 'sattag'
tag_st(x)

## S4 replacement method for signature 'sattag'
tag_st(x) <- value

## S4 method for signature 'tagstack'
tag_st(x)

## S4 replacement method for signature 'tagstack'
tag_st(x) <- value
```

### Methods (by class)

- sattag: method for sattag
- sattag: method for sattag
- tagstack: method for tagstack
- tagstack: method for tagstack

### See Also

Other slot access functions: [DeployID\(\)](#), [Ptt\(\)](#), [filename\(\)](#), [getstream\(\)](#), [instrument\(\)](#), [loadtime\(\)](#), [location\(\)](#), [species\(\)](#), [streamtype\(\)](#), [tag\\_en\(\)](#), [tagdir\(\)](#), [tagstackdir\(\)](#)

---

tag_st<-	<i>set start time of tag</i>
----------	------------------------------

---

### Description

function to set the data end time for a particular tag. if [load\\_tag](#) or [batch\\_load\\_tags](#) created the tag object then this is initially set from the EarliestDateTime in the summary stream.

### Usage

```
tag_st(x) <- value
```

---

torad	<i>convert degrees to radians</i>
-------	-----------------------------------

---

**Description**

convert degrees to radians

**Usage**

```
torad(ang)
```

**Arguments**

ang                      angle in degrees

**Value**

angle in radians

**See Also**

Other distance functions: [latlond.hav\(\)](#)

**Examples**

```
torad(180)
```

---

wch_html2df	<i>convert a wch report file (html) to data.frame</i>
-------------	---

---

**Description**

will convert the fields in a standard mk10 wch report file (html) or a directory of such files generated by mk10 host into a `data.frame` with fields as columns and ready for export into a csv.

**Usage**

```
wch_html2df(dir, file)
```

**Arguments**

dir                      a directory of wch report files (html) to convert.  
 file                    a single wch report file (html) to convert

**Details**

this is a kludge. surely there is a better way to do it... there is weird formatting and typos in the html which ever fixed will break this code. you have been warned.

**Value**

a `data.frame`. if multiple input files than each row is a new file. columns correspond to fields.



wcsv

*sensible defaults for write.table***Description**

sensible defaults for write.table

**Usage**

```
wcsv(..., sep = ",", row.names = FALSE)
```

**Arguments**

...	all the regular good stuff for finding files and whatnot.
sep	we're talking about comma seperated values here.
row.names	defaults to FALSE, because that's usually what i want. not when i'm printing a matrix or something though. but here for tables, yes this is what i want.

**Details**

please don't use write\_csv or any of that other nonsense, it breaks everyone's code...

**See Also**

Other sensible csv functions: [rcsv\(\)](#)

**Examples**

```
# use it just like write.table but without the worry
## Not run:
x <- data.frame(x1 = rnorm(10), x2 = rnorm(10))
wcsv(x, "file.csv")

## End(Not run)
```

[,es4dataframe-method *wrapper for S3 method***Description**

wrapper for S3 method

**Usage**

```
## S4 method for signature 'es4dataframe'
x[i, j, ..., drop = TRUE]
```

---

[,sattag-method      *definition for the subset operator [*

---

### Description

what i want this to do is return a sattag as opposed to [[ which should return a stream (and it does by default).

### Usage

```
## S4 method for signature 'sattag'
x[i, j, ..., drop = TRUE]
```

---

[,tagstack-method      *definition for the subset operator*

---

### Description

what i want this to do is return a tagstack as opposed to [[ which should return a tag (and it does by default).

### Usage

```
## S4 method for signature 'tagstack'
x[i, j, ..., drop = TRUE]
```

---

[<-,es4dataframe-method  
                            *wrapper for S3 method*

---

### Description

wrapper for S3 method

### Usage

```
## S4 replacement method for signature 'es4dataframe'
x[i, j, ...] <- value
```

---

\$,es4dataframe-method    *wrapper for S3 methods*

---

### Description

wrapper for S3 methods

### Usage

```
## S4 method for signature 'es4dataframe'
x$name
```

# Index

[, es4dataframe-method, [41](#)  
[, sattag-method, [42](#)  
[, tagstack-method, [42](#)  
[<-, es4dataframe-method, [42](#)  
\$, es4dataframe-method, [42](#)

all-class, [3](#)  
argos-class, [4](#)  
as.data.frame, es4dataframe-method, [4](#)  
as.POSIXct, [8](#), [9](#), [24](#)

batch\_load\_tags, [5](#), [20](#), [29](#), [38](#), [39](#)  
behavior-class, [6](#)

censor\_beh, [6](#)  
corrupt-class, [7](#)

dataframe, [5](#)  
date2num, [7](#), [24](#)  
date2num, all-method (date2num), [7](#)  
date2num, behavior-method (date2num), [7](#)  
date2num, corrupt-method (date2num), [7](#)  
date2num, fastgps-method (date2num), [7](#)  
date2num, locations-method (date2num), [7](#)  
date2num, rawargos-method (date2num), [7](#)  
date2num, rtc-method (date2num), [7](#)  
date2num, sattagstream-method (date2num), [7](#)  
date2num, series-method (date2num), [7](#)  
date2num, seriesrange-method (date2num), [7](#)  
date2num, status-method (date2num), [7](#)  
date2num, summary-method (date2num), [7](#)  
dateseq, [9](#)  
DeployID, [9](#), [14](#), [16](#), [19](#), [21](#), [26](#), [34–39](#)  
DeployID, sattag-method (DeployID), [9](#)  
DeployID, tagstack-method (DeployID), [9](#)  
duplicated\_sattagstream, [10](#)  
duplicated\_sattagstream, all-method (duplicated\_sattagstream), [10](#)  
duplicated\_sattagstream, behavior-method (duplicated\_sattagstream), [10](#)  
duplicated\_sattagstream, fastgps-method (duplicated\_sattagstream), [10](#)  
duplicated\_sattagstream, sattagstream-method (duplicated\_sattagstream), [10](#)  
duplicated\_sattagstream, series-method (duplicated\_sattagstream), [10](#)  
duplicated\_sattagstream, seriesrange-method (duplicated\_sattagstream), [10](#)

es4dataframe, [5](#), [11](#), [12](#), [16](#), [30](#)  
es4dataframe-class, [12](#)

fastgps-class, [13](#)  
filename, [10](#), [13](#), [14](#), [16](#), [19](#), [21](#), [26](#), [34–39](#)  
filename, sattag-method (filename), [13](#)  
filename, sattagstream-method (filename), [13](#)  
filename, tagstack-method (filename), [13](#)  
findInterval, [21](#)

getstream, [10](#), [14](#), [14](#), [16](#), [19](#), [21](#), [26](#), [34–39](#)  
getstream, sattag-method (getstream), [14](#)  
getstream, tagstack-method (getstream), [14](#)

has\_stream, [15](#)  
histos-class, [15](#)

initialize, sattag-method, [15](#)  
instrument, [10](#), [14](#), [16](#), [19](#), [21](#), [26](#), [34–39](#)  
instrument, sattag-method (instrument), [16](#)  
instrument, tagstack-method (instrument), [16](#)  
is.es4dataframe, [16](#)  
is.sattag (isfunctions), [17](#)  
is.sattagstream (isfunctions), [17](#)  
isfunctions, [17](#)

labels-class, [17](#)  
latlond.hav, [18](#), [40](#)  
list, [29](#), [37](#)  
load\_dap\_output, [19](#)  
load\_tag, [6](#), [13](#), [17](#), [19](#), [20](#), [26](#), [29](#), [35](#), [38](#), [39](#)  
loadtime, [10](#), [14](#), [16](#), [18](#), [21](#), [26](#), [34–39](#)  
loadtime, sattag-method (loadtime), [18](#)  
loadtime, tagstack-method (loadtime), [18](#)

- location, [10, 14, 16, 19, 20, 26, 34–39](#)
- location, sattag-method (location), [20](#)
- location, tagstack-method (location), [20](#)
- locations-class, [21](#)
- matchtimes, [21](#)
- merge, es4dataframe-method, [22](#)
- merge\_stacks, [22](#)
- minmaxdepth-class, [23](#)
- num2date, [9, 23](#)
- paginate\_series, [24](#)
- plot\_series, [24, 25](#)
- Ptt, [10, 14, 16, 19, 21, 26, 34–39](#)
- Ptt, sattag-method (Ptt), [26](#)
- Ptt, tagstack-method (Ptt), [26](#)
- rawargos-class, [26](#)
- rcsv, [27, 41](#)
- resample\_ser, [27](#)
- rtc-class, [28](#)
- sattag, [14, 15, 17, 20, 28, 30, 32, 35](#)
- sattag-class, [29](#)
- sattagstream, [8, 14, 17, 23, 29, 35](#)
- sattagstream-class, [30](#)
- ser2beh, [30](#)
- series-class, [31](#)
- seriesrange-class, [32](#)
- show, sattag-method, [32](#)
- show, sattagstream-method, [32](#)
- show, tagstack-method, [33](#)
- sort\_by\_message, [33](#)
- species, [10, 14, 16, 19, 21, 26, 33, 35–39](#)
- species, sattag-method (species), [33](#)
- species, tagstack-method (species), [33](#)
- sst-class, [34](#)
- status-class, [34](#)
- streamtype, [10, 14, 16, 19, 21, 26, 34, 35, 36–39](#)
- streamtype, sattag-method (streamtype), [35](#)
- streamtype, sattagstream-method (streamtype), [35](#)
- streamtype, tagstack-method (streamtype), [35](#)
- summary-class, [35](#)
- tag\_en, [10, 14, 16, 19, 21, 26, 34–37, 38, 39](#)
- tag\_en, sattag-method (tag\_en), [38](#)
- tag\_en, tagstack-method (tag\_en), [38](#)
- tag\_en<-, [38](#)
- tag\_en<-, sattag-method (tag\_en), [38](#)
- tag\_st, [10, 14, 16, 19, 21, 26, 34–38, 39](#)
- tag\_st, sattag-method (tag\_st), [39](#)
- tag\_st, tagstack-method (tag\_st), [39](#)
- tag\_st<-, [39](#)
- tag\_st<-, sattag-method (tag\_st), [39](#)
- tag\_st<-, tagstack-method (tag\_st), [39](#)
- tagdir, [10, 14, 16, 19, 21, 26, 34, 35, 36, 37–39](#)
- tagdir, sattag-method (tagdir), [36](#)
- tagdir, tagstack-method (tagdir), [36](#)
- tagstack, [6, 14, 19, 23, 36, 37](#)
- tagstack-class, [37](#)
- tagstackdir, [10, 14, 16, 19, 21, 26, 34–36, 37, 38, 39](#)
- torad, [18, 40](#)
- wch\_html2df, [40](#)
- wcsv, [27, 41](#)