

# Package ‘IgorR’

May 21, 2017

**Type** Package

**Title** Read Binary Files Saved by 'Igor Pro' (Including 'Neuromatic'  
Data)

**Version** 0.8.1

**Author** Greg Jefferis with contributions from Thomas Braun

**Maintainer** Greg Jefferis <jefferis@gmail.com>

**Description** Provides function to read data from the 'Igor Pro' data analysis program by Wavemetrics. The data formats supported are 'Igor' packed experiment format (pxp) and 'Igor' binary wave (ibw). See: <http://www.wavemetrics.com/> for details. Also includes functions to load special pxp files produced by the 'Igor Pro' 'Neuromatic' and 'Nclamp' packages for recording and analysing neuronal data. See <http://www.neuromatic.thinkrandom.com/> for details.

**Imports** bitops, tools

**Suggests** testthat

**License** GPL (>= 2)

**LazyLoad** yes

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2017-05-21 08:29:22 UTC

## R topics documented:

IgorR-package . . . . .	2
IgorR-private . . . . .	2
read.ibw . . . . .	3
read.pxp . . . . .	4
ReadAllNclampLogTables . . . . .	5
ReadNclampLogTable . . . . .	6
SummariseSweepFile . . . . .	6

SweepFilesToDataFrame . . . . .	7
tsp.igorwave . . . . .	8
UpdateSweepDataFrame . . . . .	8
WaveToTimeSeries . . . . .	9

<b>Index</b>	<b>10</b>
--------------	-----------

---

IgorR-package	<i>Read binary files saved by Igor Pro (including Neuromatic data)</i>
---------------	--

---

### Description

This package provides function to read data from the Igor Pro data analysis program by Wavemetrics. The data formats supported are Igor packed experiment format (pxp) and Igor binary wave (ibw). See: <http://www.wavemetrics.com/> for details.

### Details

It also includes some functions to load special ppx files produced by the Neuromatic/Nclamp packages for recording and analysing neuronal #data. See <http://www.neuromatic.thinkrandom.com/> for details.

Key functions in the package include [read.ibw](#), [read.ppx](#).

---

IgorR-private	<i>Private functions in IgorR Package</i>
---------------	---

---

### Description

Private functions in IgorR Package

### Usage

```
.ReadPackedHeader(con, endian)
```

### Arguments

con	an R connection to the file we are reading
endian	either little or big

### Value

a list containing information about the current record

### Author(s)

jefferis

---

read.ibw	<i>Read binary files in the Igor Binary Wave format (IBW)</i>
----------	---

---

## Description

Read binary files in the Igor Binary Wave format (IBW)

## Usage

```
read.ibw(wavefile, Verbose = FALSE, ReturnTimeSeries = FALSE,  
         MakeWave = FALSE, HeaderOnly = FALSE)
```

## Arguments

wavefile	either a character vector containing the path to a file or an R <a href="#">connection</a> .
Verbose	if TRUE, print status information while reading the file.
ReturnTimeSeries	if TRUE, return as an R time series (package <a href="#">ts</a> ).
MakeWave	if TRUE, assign wave to a list in the global user environment.
HeaderOnly	if TRUE, only return the header of the Igor Wave.

## Value

A vector containing the wave data or, if `MakeWave == TRUE`, returns the name of a new R vector containing the data which has been made in the user environment

## Author(s)

jefferis

## Examples

```
# return a list containing the wave  
wavedata=read.ibw(system.file("igor", "version5.ibw", package="IgorR"))  
sum(wavedata)  
  
# make a list containing the wave's data in the users's environment  
wavename=read.ibw(system.file("igor", "version5.ibw", package="IgorR"), MakeWave=TRUE)  
sum(get(wavename))
```

---

read.pxp	<i>Reads an Igor Pro Packed Experiment (.pxp) file</i>
----------	--

---

### Description

Note that PXP files are only partially documented so some contents cannot be parsed (e.g. image data). This function currently reads data records (Igor waves and variables), history, procedures, recreation macros and plain text notebooks. Formatted notebooks cannot be read.

### Usage

```
read.pxp(pxpfile, regex, ReturnTimeSeries = FALSE, Verbose = FALSE,
        StructureOnly = FALSE, ExtractText = FALSE, IgorPlatform = NULL, ...)
```

### Arguments

pxpfile	character vector naming a PXP file or an R <a href="#">connection</a> .
regex	if TRUE, only read records (e.g. waves) in the PXP file whose names match a <a href="#">regex</a> .
ReturnTimeSeries	if TRUE, Igor waves are returned as a <code>link{ts}</code> object with sensible x scaling (FALSE by default).
Verbose	whether to print information to console during loading (numeric values are also allowed 0=none, 1=basic, 2=all).
StructureOnly	(TODO) if TRUE, only the structure of the PXP file for inspection.
ExtractText	whether to extract procedures, recreation macros, history and plain text notebooks (FALSE by default).
IgorPlatform	OS on which Igor file was saved (windows or macintosh).
...	optional parameters passed to <a href="#">read.ibw</a> .

### Details

IgorPlatform will determine in which encoding text is read (WINDOWS-1252 for windows and macintosh for macintosh). Unique abbreviations are acceptable. Defaults to "windows" on Windows, "macintosh" otherwise. Note that Igor Pro 5.5 added a PlatformRecord to the PXP file format which is used to determine the file's platform of origin when available. Since this is information straight from the horse's mouth it will override the IgorPlatform argument.

### Value

A list containing all the individual waves or variables in the PXP file.

### Author(s)

jefferis

**Examples**

```
r=read.pxp(system.file("igor","testexpt.pxp",package="IgorR"))
```

---

ReadAllNclampLogTables

*Read all Nclamp log tables from a directory into a list*

---

**Description**

Read all Nclamp log tables from a directory into a list

**Usage**

```
ReadAllNclampLogTables(logfiledir, pattern = "_log[0-9]+[.]pxp$", ...)
```

**Arguments**

logfiledir      path to directory containing log files (pxp files).  
pattern          optional regular expression – see [list.files](#).  
...              additional parameters for [ReadNclampLogTable](#).

**Value**

A named list containing one dataframe for each parsed log file.

**Author(s)**

jefferis

**Examples**

```
## Not run:  
logfiles <- ReadAllNclampLogTables("/GD/projects/PhysiologyData/logs")  
str(logfiles)  
  
## End(Not run)
```

---

ReadNclampLogTable      *Read the log table produced by Nclamp acquisition software for Igor*

---

### Description

log tables are special Igor .pxp files that contain only variables. Each entry corresponds to a single run of an Nclamp protocol, storing information like protocol name, run time etc.

### Usage

```
ReadNclampLogTable(f, Verbose = FALSE)
```

### Arguments

f                      path to the log file.  
 Verbose              whether to print status information while reading the file.

### Value

A dataframe containing a row for each acquisition protocol run.

### Author(s)

jefferis

---

SummariseSweepFile      *Extract summary information from an Nclamp/Igor Sweep File*

---

### Description

e.g. for import into Physiology database

### Usage

```
SummariseSweepFile(f, Verbose = FALSE)
```

### Arguments

f                      path to an Nclamp/Igor PXP format sweep file.  
 Verbose              if TRUE, print details while parsing underlying PXP file.

### Value

A list of about 25 fields summarising the sweep file.

**Author(s)**

jefferis

**Examples**

```
l <- SummariseSweepFile(system.file("igor", "WedJul407c2_001.pxp", package="IgorR"))
cat("There are", l$NumWaves, "waves in the file each of total duration", l$StimWaveLength,
    "ms and sample duration", l$StimSampleInterval, "ms \n")
```

---

SweepFilesToDataFrame *Summarise multiple sweep files into a single dataframe*

---

**Description**

Note that this is still a little fragile if the lists produced by [SummariseSweepFile](#) do not have consistent field names.

**Usage**

```
SweepFilesToDataFrame(ff)
```

**Arguments**

ff paths to a set of sweep files.

**Value**

A dataframe with rows for each sweep file.

**Author(s)**

jefferis

**See Also**[SummariseSweepFile](#)

---

tsp.igorwave	<i>Return tsp attribute of igor wave (start, end, frequency)</i>
--------------	--

---

**Description**

Note that  $\text{end} = (\text{npts}-1) * \text{deltat}$

**Usage**

```
tsp.igorwave(wave)
```

**Arguments**

wave	Igor wave loaded by read.ibw or read.pxp
------	--

**Value**

numeric vector with elements start, end, frequency

**Author(s)**

jefferis

**See Also**

tsp

---

UpdateSweepDataFrame	<i>Update the CSV file summarising the sweeps in an Nclamp data folder</i>
----------------------	--

---

**Description**

Update the CSV file summarising the sweeps in an Nclamp data folder

**Usage**

```
UpdateSweepDataFrame(folder, outfile = NULL, action = c("update", "force"),
  DryRun = FALSE)
```

**Arguments**

folder	path to the folder.
outfile	path to outfile (default: /path/to/datafolder/datafolder.csv).
action	update newer (default) or force update (not yet implemented).
DryRun	if TRUE, report which files would be processed, but do nothing.



**Value**

TRUE if something happened, FALSE otherwise.

**Author(s)**

jefferis

---

WaveToTimeSeries	<i>Convert an Igor wave (or list of waves) loaded by read.ibw into an R time series</i>
------------------	---

---

**Description**

Where there are multiple waves, they are assumed to be of compatible lengths so that they can be joined together by cbind.

**Usage**

```
WaveToTimeSeries(WaveData, ReturnOriginalDataOnError = TRUE)
```

**Arguments**

WaveData, a wave or list of waves

ReturnOriginalDataOnError

If we can't make a time series, return return original data (default TRUE)

**Value**

a time series or multi time series (ts, mts)

**Author(s)**

jefferis

# Index

## \*Topic **package**

- IgorR-package, [2](#)
- .ReadPackedHeader (IgorR-private), [2](#)
- connection, [3](#), [4](#)
- IgorR (IgorR-package), [2](#)
- IgorR-package, [2](#)
- IgorR-private, [2](#)
- list.files, [5](#)
- read.ibw, [2](#), [3](#), [4](#)
- read.pxp, [2](#), [4](#)
- ReadAllNclampLogTables, [5](#)
- ReadNclampLogTable, [5](#), [6](#)
- regex, [4](#)
- SummariseSweepFile, [6](#), [7](#)
- SweepFilesToDataFrame, [7](#)
- ts, [3](#)
- tsp.igorwave, [8](#)
- UpdateSweepDataFrame, [8](#)
- WaveToTimeSeries, [9](#)