

# Package ‘bupaR’

December 4, 2017

**Type** Package

**Title** Business Process Analysis in R

**Version** 0.3.2

**Date** 2017-12-04

**Description** Comprehensive Business Process Analysis toolkit. Creates S3-class for event log objects, and related handler functions. Imports related packages for filtering event data, computation of descriptive statistics, handling of 'Petri Net' objects and visualization of process maps. See also packages 'edeaR', 'processmapR', 'eventdataR' and 'processmonitR'.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1

**Imports** magrittr, dplyr, data.table, shiny, miniUI, purrr, tidyr,  
glue, forcats, rlang

**URL** <https://www.bupar.net>

**NeedsCompilation** no

**Author** Gert Janssenswillen [aut, cre]

**Maintainer** Gert Janssenswillen <gert.janssenswillen@uhasselt.be>

**Repository** CRAN

**Date/Publication** 2017-12-04 19:46:48 UTC

## R topics documented:

activities . . . . .	2
activities_to_eventlog . . . . .	3
activity_id . . . . .	4
activity_instance_id . . . . .	4
activity_labels . . . . .	5
act_collapse . . . . .	6
act_recode . . . . .	6

act_unite . . . . .	7
bupaR . . . . .	8
cases . . . . .	8
case_id . . . . .	9
case_labels . . . . .	9
case_list . . . . .	10
durations . . . . .	10
eventlog . . . . .	11
filter_attributes . . . . .	12
group_by_activity . . . . .	13
group_by_activity_instance . . . . .	13
group_by_case . . . . .	14
group_by_resource . . . . .	14
group_by_resource_activity . . . . .	15
lifecycle_id . . . . .	15
mapping . . . . .	16
n_activities . . . . .	17
n_activity_instances . . . . .	17
n_cases . . . . .	18
n_events . . . . .	19
n_resources . . . . .	19
n_traces . . . . .	20
print.eventlog . . . . .	21
print.eventlog_mapping . . . . .	21
resources . . . . .	22
resource_id . . . . .	22
resource_labels . . . . .	23
re_map . . . . .	24
simple_eventlog . . . . .	24
summary.eventlog . . . . .	25
timestamp . . . . .	25
traces . . . . .	26
trace_list . . . . .	27

<b>Index</b>	<b>28</b>
--------------	-----------

---

activities	<i>Activities</i>
------------	-------------------

---

## Description

Returns a `tbl_df` containing a list of all activity types in the event log, with their absolute and relative frequency

**Usage**

```

activities(eventlog)

## S3 method for class 'eventlog'
activities(eventlog)

## S3 method for class 'grouped_eventlog'
activities(eventlog)

```

**Arguments**

eventlog            The event log to be used. An object of class eventlog.

**Methods (by class)**

- eventlog: Generate activity list for eventlog
- grouped\_eventlog: Generate activity list for grouped eventlog

**See Also**

[activity\\_id,activity\\_instance\\_id,eventlog](#)

---

activities\_to\_eventlog

*Create event log from list of activity instances*

---

**Description**

Create event log from list of activity instances

**Usage**

```

activities_to_eventlog(activity_log, case_id, activity_id, resource_id,
  timestamps)

```

**Arguments**

activity_log	A data.frame where each row is an activity instances
case_id	Column name of the case identifier
activity_id	Column name of the activity identifier
resource_id	Column name of the resource identifier
timestamps	A vector of column names containing different timestamp. To column names will be transformed to lifecycle identifiers

---

activity_id	<i>Activity classifier</i>
-------------	----------------------------

---

**Description**

Get the activity classifier of an object of class eventlog.

**Usage**

```
activity_id(x)

## S3 method for class 'eventlog'
activity_id(x)

## S3 method for class 'eventlog_mapping'
activity_id(x)
```

**Arguments**

x                    An eventlog of eventlog\_mapping

**Methods (by class)**

- eventlog: Retrieve activity identifier from eventlog
- eventlog\_mapping: Retrieve activity identifier from eventlog mapping

**See Also**

[eventlog](#), [mapping](#)

Other Eventlog classifiers: [activity\\_instance\\_id](#), [case\\_id](#), [lifecycle\\_id](#), [mapping](#), [resource\\_id](#), [timestamp](#)

---

activity_instance_id	<i>Activity instance classifier</i>
----------------------	-------------------------------------

---

**Description**

Get the activity instance classifier of an object of class eventlog.

**Usage**

```
activity_instance_id(x)

## S3 method for class 'eventlog'
activity_instance_id(x)

## S3 method for class 'eventlog_mapping'
activity_instance_id(x)
```

**Arguments**

x                    An eventlog of eventlog\_mapping

**Methods (by class)**

- eventlog: Retrieve activity instance identifier from eventlog
- eventlog\_mapping: Retrieve activity instance identifier from eventlog mapping

**See Also**

Other Eventlog classifiers: [activity\\_id](#), [case\\_id](#), [lifecycle\\_id](#), [mapping](#), [resource\\_id](#), [timestamp](#)

---

activity_labels	<i>Get vector of activity labels</i>
-----------------	--------------------------------------

---

**Description**

Retrieve a vector containing all unique activity labels

**Usage**

```
activity_labels(eventlog)

## S3 method for class 'eventlog'
activity_labels(eventlog)
```

**Arguments**

eventlog            Eventlog

**Methods (by class)**

- eventlog: Retrieve activity labels from eventlog

---

act_collapse	<i>Collapse activity labels of a sub process into a single activity</i>
--------------	---

---

**Description**

Collapse activity labels of a sub process into a single activity

**Usage**

```
act_collapse(eventlog, ...)
```

```
## S3 method for class 'eventlog'  
act_collapse(eventlog, ...)
```

**Arguments**

eventlog	An eventlog object
...	A series of named character vectors. The activity labels in each vector will be collapsed into one activity with the name of the vector.

**Methods (by class)**

- eventlog: Collapse activity labels of a subprocess into a single activity

**See Also**

Other Activity processing functions: [act\\_recode](#), [act\\_unite](#)

---

act_recode	<i>Recode activity labels</i>
------------	-------------------------------

---

**Description**

Recode one or more activity labels through specifying their old and new label

**Usage**

```
act_recode(eventlog, ...)
```

```
## S3 method for class 'eventlog'  
act_recode(eventlog, ...)
```

**Arguments**

eventlog      An object of class eventlog.  
 ...            A sequence of named character vectors of length one where the names gives the new label and the value gives the old label. Labels not mentioned will be left unchanged.

**Methods (by class)**

- eventlog: Recode activity labels of event log

**See Also**

[eventlog](#), [activity\\_id](#), [act\\_unite](#)

Other Activity processing functions: [act\\_collapse](#), [act\\_unite](#)

---

act_unite	<i>Unite activity labels</i>
-----------	------------------------------

---

**Description**

Recode two or different more activity labels two a uniform activity label

**Usage**

```
act_unite(eventlog, ...)  
  
## S3 method for class 'eventlog'  
act_unite(eventlog, ...)
```

**Arguments**

eventlog      An object of class eventlog.  
 ...            A series of named character vectors. The activity labels in each vector will be replaced with the name.

**Methods (by class)**

- eventlog: Unite activity labels in event log

**See Also**

[eventlog](#), [activity\\_id](#), [act\\_recode](#)

Other Activity processing functions: [act\\_collapse](#), [act\\_recode](#)

---

bupaR

*bupaR - Business Process Analysis in R*

---

### Description

Functionalities for process analysis in R. This packages implements an S3-class for event log objects, and related handler functions. Imports related packages for subsetting event data, computation of descriptive statistics, handling of Petri Net objects and visualization of process maps.

---

cases

*Cases*

---

### Description

Provides a fine-grained summary of an event log with characteristics for each case: the number of events, the number of activity types, the timespan, the trace, the duration and the first and last event type.

### Usage

```
cases(eventlog)

## S3 method for class 'eventlog'
cases(eventlog)
```

### Arguments

eventlog      An eventlog object. eventlog.

### Methods (by class)

- eventlog: Constructy list of cases in an eventlog



---

case_id	<i>Case classifier</i>
---------	------------------------

---

**Description**

Get the case classifier of an object of class eventlog

**Usage**

```
case_id(x)

## S3 method for class 'eventlog'
case_id(x)

## S3 method for class 'eventlog_mapping'
case_id(x)
```

**Arguments**

x                    An eventlog of eventlog\_mapping

**Methods (by class)**

- eventlog: Retrieve case identifier from eventlog
- eventlog\_mapping: Retrieve case identifier from eventlog mapping

**See Also**

[eventlog](#), [mapping](#)

Other Eventlog classifiers: [activity\\_id](#), [activity\\_instance\\_id](#), [lifecycle\\_id](#), [mapping](#), [resource\\_id](#), [timestamp](#)

---

case_labels	<i>Get vector of case labels</i>
-------------	----------------------------------

---

**Description**

Retrieve a vector containing all unique case labels

**Usage**

```
case_labels(eventlog)

## S3 method for class 'eventlog'
case_labels(eventlog)
```

**Arguments**

eventlog      Eventlog

**Methods (by class)**

- eventlog: Retrieve case labels from eventlog

---

case_list	<i>Case list</i>
-----------	------------------

---

**Description**

Construct list of cases

**Usage**

```
case_list(eventlog)
```

```
## S3 method for class 'eventlog'
case_list(eventlog)
```

**Arguments**

eventlog      Eventlog object

**Methods (by class)**

- eventlog: Return case list

---

durations	<i>Durations</i>
-----------	------------------

---

**Description**

Computes the throughput times of each case. Throughput time is defined as the interval between the start of the first event and the completion of the last event.

**Usage**

```
durations(eventlog, units)
```

```
## S3 method for class 'eventlog'
durations(eventlog, units = "days")
```

**Arguments**

eventlog	The event log to be used. An object of class eventlog.
units	The time unit in which the throughput times should be reported.

**Methods (by class)**

- eventlog: Compute durations from eventlog

---

eventlog	<i>Eventlog</i>
----------	-----------------

---

**Description**

A function to instantiate an object of class eventlog by specifying a data.frame or tbl\_df and appropriate case, activity and timestamp classifiers.

**Usage**

```
eventlog(eventlog, case_id = NULL, activity_id = NULL,
         activity_instance_id = NULL, lifecycle_id = NULL, timestamp = NULL,
         resource_id = NULL)
```

```
ieventlog(eventlog)
```

**Arguments**

eventlog	The data object to be used as event log. This can be a data.frame or tbl_df.
case_id	The case classifier of the event log. A character vector containing variable names of length 1 or more.
activity_id	The activity classifier of the event log. A character vector containing variable names of length 1 or more.
activity_instance_id	The activity instance classifier of the event log.
lifecycle_id	The life cycle classifier of the event log.
timestamp	The timestamp of the event log. Should refer to a Date or POSIXct field.
resource_id	The resource identifier of the event log. A character vector containing variable names of length 1 or more.

**See Also**

[case\\_id](#), [activity\\_id](#), [activity\\_instance\\_id](#), [lifecycle\\_id](#), [timestamp](#)

## Examples

```
## Not run:
data <- data.frame(case = rep("A",5),
  activity_id = c("A","B","C","D","E"),
  activity_instance_id = 1:5,
  lifecycle_id = rep("complete",5),
  timestamp = 1:5,
  resource = rep("resource 1", 5))
eventlog(data,case_id = "case",
  activity_id = "activity_id",
  activity_instance_id = "activity_instance_id",
  lifecycle_id = "lifecycle_id",
  timestamp = "timestamp",
  resource_id = "resource")

## End(Not run)
```

---

filter\_attributes      *Generic filter function for eventlog*

---

## Description

Generic filter function for eventlog

## Usage

```
filter_attributes(eventlog, ...)

## S3 method for class 'eventlog'
filter_attributes(eventlog, ...)

## S3 method for class 'grouped_eventlog'
filter_attributes(eventlog, ...)
```

## Arguments

eventlog	Eventlog object
...	Filter conditions

## Methods (by class)

- eventlog: Filter eventlog using attributes
- grouped\_eventlog: Filter grouped eventlog using attributes

---

group_by_activity	<i>Group event log on activity id</i>
-------------------	---------------------------------------

---

**Description**

Group an event log by activity identifier

**Usage**

```
group_by_activity(eventlog)
```

```
## S3 method for class 'eventlog'  
group_by_activity(eventlog)
```

**Arguments**

eventlog	Eventlog
----------	----------

**Methods (by class)**

- eventlog: Group eventlog on activity identifier

---

group_by_activity_instance	<i>Group event log on activity instance id</i>
----------------------------	--

---

**Description**

Group an event log by activity instance identifier

**Usage**

```
group_by_activity_instance(eventlog)
```

```
## S3 method for class 'eventlog'  
group_by_activity_instance(eventlog)
```

**Arguments**

eventlog	Eventlog
----------	----------

**Methods (by class)**

- eventlog: Group eventlog on activity instance identifier

---

group_by_case	<i>Group event log on case id</i>
---------------	-----------------------------------

---

**Description**

Group an event log by case identifier

**Usage**

```
group_by_case(eventlog)

## S3 method for class 'eventlog'
group_by_case(eventlog)
```

**Arguments**

eventlog      Eventlog

**Methods (by class)**

- eventlog: Group eventlog on case identifier

---

group_by_resource	<i>Group event log on resource id</i>
-------------------	---------------------------------------

---

**Description**

Group an event log by resource identifier

**Usage**

```
group_by_resource(eventlog)

## S3 method for class 'eventlog'
group_by_resource(eventlog)
```

**Arguments**

eventlog      Eventlog

**Methods (by class)**

- eventlog: Group eventlog on resource identifier

---

group\_by\_resource\_activity  
*Group event log on resource and activity id*

---

**Description**

Group an event log by resource and activity identifier

**Usage**

```
group_by_resource_activity(eventlog)

## S3 method for class 'eventlog'
group_by_resource_activity(eventlog)
```

**Arguments**

eventlog      Eventlog

**Methods (by class)**

- eventlog: Group an event log by resource and activity identifier

---

lifecycle\_id      *Life cycle classifier*

---

**Description**

Get the lifecycle\_id of an object of class eventlog

**Usage**

```
lifecycle_id(x)

## S3 method for class 'eventlog'
lifecycle_id(x)

## S3 method for class 'eventlog_mapping'
lifecycle_id(x)
```

**Arguments**

x                      An eventlog of eventlog\_mapping

**Methods (by class)**

- `eventlog`: Retrieve lifecycle identifier from eventlog
- `eventlog_mapping`: Retrieve lifecycle identifier from eventlog mapping

**See Also**

Other Eventlog classifiers: [activity\\_id](#), [activity\\_instance\\_id](#), [case\\_id](#), [mapping](#), [resource\\_id](#), [timestamp](#)

---

mapping

*Mapping*

---

**Description**

Prints the mapping of an event log object.

**Usage**

```
mapping(eventlog)
```

```
## S3 method for class 'eventlog'  
mapping(eventlog)
```

**Arguments**

`eventlog`      The event log to be used. An object of class eventlog.

**Methods (by class)**

- `eventlog`: Retrieve identifier mapping from eventlog

**See Also**

Other Eventlog classifiers: [activity\\_id](#), [activity\\_instance\\_id](#), [case\\_id](#), [lifecycle\\_id](#), [resource\\_id](#), [timestamp](#)



---

n_activities	<i>n_activities</i>
--------------	---------------------

---

**Description**

Returns the number of activities in an event log

**Usage**

```
n_activities(eventlog)

## S3 method for class 'eventlog'
n_activities(eventlog)

## S3 method for class 'grouped_eventlog'
n_activities(eventlog)
```

**Arguments**

eventlog      The event log to be used. An object of class eventlog.

**Methods (by class)**

- eventlog: Count the number of activities in an event log
- grouped\_eventlog: Count the number of activities for a grouped event log

**See Also**

Other Eventlog count functions: [n\\_activity\\_instances](#), [n\\_cases](#), [n\\_events](#), [n\\_resources](#), [n\\_traces](#)

---

n_activity_instances	<i>n_activity_instances</i>
----------------------	-----------------------------

---

**Description**

Returns the number of activity instances in an event log

**Usage**

```
n_activity_instances(eventlog)

## S3 method for class 'eventlog'
n_activity_instances(eventlog)

## S3 method for class 'grouped_eventlog'
n_activity_instances(eventlog)
```

**Arguments**

eventlog            The event log to be used. An object of class eventlog.

**See Also**

Other Eventlog count functions: [n\\_activities](#), [n\\_cases](#), [n\\_events](#), [n\\_resources](#), [n\\_traces](#)

---

n_cases	<i>n_cases</i>
---------	----------------

---

**Description**

Returns the number of cases in an event log

**Usage**

```
n_cases(eventlog)

## S3 method for class 'eventlog'
n_cases(eventlog)

## S3 method for class 'grouped_eventlog'
n_cases(eventlog)
```

**Arguments**

eventlog            The event log to be used. An object of class eventlog.

**Methods (by class)**

- eventlog: Count number of cases for eventlog
- grouped\_eventlog: Count number of cases for grouped eventlog

**See Also**

Other Eventlog count functions: [n\\_activities](#), [n\\_activity\\_instances](#), [n\\_events](#), [n\\_resources](#), [n\\_traces](#)

---

n_events	<i>n_events</i>
----------	-----------------

---

**Description**

Returns the number of events in an event log

**Usage**

```
n_events(eventlog)

## S3 method for class 'eventlog'
n_events(eventlog)

## S3 method for class 'grouped_eventlog'
n_events(eventlog)
```

**Arguments**

eventlog      The event log to be used. An object of class eventlog.

**Methods (by class)**

- eventlog: Count number of resources in eventlog
- grouped\_eventlog: Count number of resource in eventlog

**See Also**

Other Eventlog count functions: [n\\_activities](#), [n\\_activity\\_instances](#), [n\\_cases](#), [n\\_resources](#), [n\\_traces](#)

---

n_resources	<i>n_resources</i>
-------------	--------------------

---

**Description**

Returns the number of resources in an event log

**Usage**

```
n_resources(eventlog)

## S3 method for class 'eventlog'
n_resources(eventlog)

## S3 method for class 'grouped_eventlog'
n_resources(eventlog)
```

**Arguments**

`eventlog`            The event log to be used. An object of class `eventlog`.

**Methods (by class)**

- `eventlog`: Count number of resources in `eventlog`
- `grouped_eventlog`: Count number of resources in grouped `eventlog`

**See Also**

Other Eventlog count functions: [n\\_activities](#), [n\\_activity\\_instances](#), [n\\_cases](#), [n\\_events](#), [n\\_traces](#)

---

`n_traces`

*n\_traces*

---

**Description**

Returns the number of traces in an event log

**Usage**

```
n_traces(eventlog)
```

```
## S3 method for class 'eventlog'
n_traces(eventlog)
```

```
## S3 method for class 'grouped_eventlog'
n_traces(eventlog)
```

**Arguments**

`eventlog`            The event log to be used. An object of class `eventlog`.

**Methods (by class)**

- `eventlog`: Count number of traces for `eventlog`
- `grouped_eventlog`: Count number of traces for grouped `eventlog`

**See Also**

Other Eventlog count functions: [n\\_activities](#), [n\\_activity\\_instances](#), [n\\_cases](#), [n\\_events](#), [n\\_resources](#)

---

print.eventlog	<i>Generic print function for eventlog</i>
----------------	--

---

**Description**

Generic print function for eventlog

**Usage**

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

**Arguments**

x	Eventlog object
...	Additional Arguments

---

print.eventlog_mapping	<i>Generic print function for eventlog_mapping</i>
------------------------	--

---

**Description**

Generic print function for eventlog\_mapping

**Usage**

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

**Arguments**

x	Eventlog mapping object
...	Additional Arguments

---

 resources

*Resources*


---

**Description**

Returns a tbl\_df containing a list of all resources in the event log, with there absolute and relative frequency

**Usage**

```
resources(eventlog)

## S3 method for class 'eventlog'
resources(eventlog)

## S3 method for class 'grouped_eventlog'
resources(eventlog)
```

**Arguments**

eventlog            The event log to be used. An object of class eventlog.

**Methods (by class)**

- eventlog: Generate resource list for eventlog
- grouped\_eventlog: Generate resource list for grouped eventlog

**See Also**

[resource\\_id, eventlog](#)

---

resource\_id

*Resource classifier*


---

**Description**

Get the resource classifier of an object of class eventlog.

**Usage**

```
resource_id(x)

## S3 method for class 'eventlog'
resource_id(x)

## S3 method for class 'eventlog_mapping'
resource_id(x)
```

**Arguments**

x                    An eventlog of eventlog\_mapping

**Methods (by class)**

- eventlog: Retrieve resource identifier from eventlog
- eventlog\_mapping: Retrieve resource identifier from eventlog mapping

**See Also**

[eventlog](#), [mapping](#)

Other Eventlog classifiers: [activity\\_id](#), [activity\\_instance\\_id](#), [case\\_id](#), [lifecycle\\_id](#), [mapping](#), [timestamp](#)

---

resource\_labels            *Get vector of resource labels*

---

**Description**

Retrieve a vector containing all unique resource labels

**Usage**

```
resource_labels(eventlog)

## S3 method for class 'eventlog'
resource_labels(eventlog)
```

**Arguments**

eventlog            Eventlog

**Methods (by class)**

- eventlog: Retrieve resource labels from eventlog

---

re_map	<i>Re map</i>
--------	---------------

---

**Description**

Construct and eventlog using an existing mapping.

**Usage**

```
re_map(eventlog, eventlog_mapping)
```

**Arguments**

eventlog	The event log data to be used.
eventlog_mapping	An existing eventlog mapping created by the mapping function

---

simple_eventlog	<i>Simple Eventlog</i>
-----------------	------------------------

---

**Description**

A function to instantiate an object of class eventlog by specifying a data.frame or tbl\_df and the minimally required case identifier, activity identifier and timestamp

**Usage**

```
simple_eventlog(eventlog, case_id = NULL, activity_id = NULL,
  timestamp = NULL)
```

```
isimple_eventlog(eventlog)
```

**Arguments**

eventlog	The data object to be used as event log. This can be a data.frame or tbl_df.
case_id	The case classifier of the event log.
activity_id	The activity classifier of the event log.
timestamp	The timestamp of the event log.

**See Also**

[eventlog](#), [case\\_id](#), [activity\\_id](#), [activity\\_instance\\_id](#), [lifecycle\\_id](#), [timestamp](#)



**Examples**

```
## Not run:
data <- data.frame(case = rep("A",5),
  activity_id = c("A","B","C","D","E"),
  timestamp = 1:5,
  simple_eventlog(data,case_id = "case",
  activity_id = "activity_id",
  timestamp = "timestamp")

## End(Not run)
```

---

summary.eventlog	<i>Generic summary function for eventlog class</i>
------------------	--

---

**Description**

Generic summary function for eventlog class

**Usage**

```
## S3 method for class 'eventlog'
summary(object, ...)

## S3 method for class 'grouped_eventlog'
summary(object, ...)
```

**Arguments**

object	Eventlog object
...	Additional Arguments

**Methods (by class)**

- grouped\_eventlog: Summary of grouped event log

---

timestamp	<i>Timestamp classifier</i>
-----------	-----------------------------

---

**Description**

Get the timestamp classifier of an object of class eventlog

**Usage**

```
timestamp(x)

## S3 method for class 'eventlog'
timestamp(x)

## S3 method for class 'eventlog_mapping'
timestamp(x)
```

**Arguments**

x                    An eventlog of eventlog\_mapping

**Methods (by class)**

- eventlog: Retrieve timestamp identifier from eventlog
- eventlog\_mapping: Retrieve timestamp identifier from eventlog mapping

**See Also**

[eventlog](#), [mapping](#)

Other Eventlog classifiers: [activity\\_id](#), [activity\\_instance\\_id](#), [case\\_id](#), [lifecycle\\_id](#), [mapping](#), [resource\\_id](#)

---

traces

*Traces*

---

**Description**

traces computes the different activity sequences of an event log together with their absolute and relative frequencies. Activity sequences are based on the start timestamp of activities.

**Usage**

```
traces(eventlog, ...)
```

```
## S3 method for class 'eventlog'
traces(eventlog, ...)
```

```
## S3 method for class 'grouped_eventlog'
traces(eventlog, ...)
```

**Arguments**

eventlog            The event log to be used. An object of class eventlog.

...                 Deprecated arguments

**Methods (by class)**

- eventlog: Construct traces list for eventlog
- grouped\_eventlog: Construct list of traces for grouped eventlog

**See Also**

[cases](#), [eventlog](#)

---

trace_list	<i>Trace list</i>
------------	-------------------

---

**Description**

Construct trace list

**Usage**

```
trace_list(eventlog)
```

```
## S3 method for class 'eventlog'  
trace_list(eventlog)
```

**Arguments**

eventlog      Eventlog object

**Methods (by class)**

- eventlog: Construct trace list for event log

# Index

act\_collapse, [6, 7](#)  
act\_recode, [6, 6, 7](#)  
act\_unite, [6, 7, 7](#)  
activities, [2](#)  
activities\_to\_eventlog, [3](#)  
activity\_id, [3, 4, 5, 7, 9, 11, 16, 23, 24, 26](#)  
activity\_instance\_id, [3, 4, 4, 9, 11, 16, 23, 24, 26](#)  
activity\_labels, [5](#)

bupaR, [8](#)  
bupaR-package (bupaR), [8](#)

case\_id, [4, 5, 9, 11, 16, 23, 24, 26](#)  
case\_labels, [9](#)  
case\_list, [10](#)  
cases, [8, 27](#)

durations, [10](#)

eventlog, [3, 4, 7, 9, 11, 22–24, 26, 27](#)

filter\_attributes, [12](#)

group\_by\_activity, [13](#)  
group\_by\_activity\_instance, [13](#)  
group\_by\_case, [14](#)  
group\_by\_resource, [14](#)  
group\_by\_resource\_activity, [15](#)

ieventlog (eventlog), [11](#)  
isimple\_eventlog (simple\_eventlog), [24](#)

lifecycle\_id, [4, 5, 9, 11, 15, 16, 23, 24, 26](#)

mapping, [4, 5, 9, 16, 16, 23, 26](#)

n\_activities, [17, 18–20](#)  
n\_activity\_instances, [17, 17, 18–20](#)  
n\_cases, [17, 18, 18, 19, 20](#)  
n\_events, [17, 18, 19, 20](#)

n\_resources, [17–19, 19, 20](#)  
n\_traces, [17–20, 20](#)

print.eventlog, [21](#)  
print.eventlog\_mapping, [21](#)

re\_map, [24](#)  
resource\_id, [4, 5, 9, 16, 22, 22, 26](#)  
resource\_labels, [23](#)  
resources, [22](#)

simple\_eventlog, [24](#)  
summary.eventlog, [25](#)  
summary.grouped\_eventlog  
(summary.eventlog), [25](#)

timestamp, [4, 5, 9, 11, 16, 23, 24, 25](#)  
trace\_list, [27](#)  
traces, [26](#)