

## NAME

Log::Message::Item - Message objects for Log::Message

## SYNOPSIS

```
# Implicitly used by Log::Message to create Log::Message::Item objects

print "this is the message's id: ",    $item->id;

print "this is the message stored: ",  $item->message;

print "this is when it happened: ",    $item->when;

print "the message was tagged: ",      $item->tag;

print "this was the severity level: ",  $item->level;

$item->remove; # delete the item from the stack it was on

# Besides these methods, you can also call the handlers on
# the object specificallly.
# See the Log::Message::Handlers manpage for documentation on what
# handlers are available by default and how to add your own
```

## DESCRIPTION

Log::Message::Item is a class that generates generic Log items. These items are stored on a Log::Message stack, so see the Log::Message manpage about details how to retrieve them.

You should probably not create new items by yourself, but use the storing mechanism provided by Log::Message.

However, the accessors and handlers are of interest if you want to do fine tuning of how your messages are handled.

The accessors and methods are described below, the handlers are documented in the Log::Message::Handlers manpage.

## Methods and Accessors

### remove

Calling remove will remove the object from the stack it was on, so it will not show up any more in subsequent fetches of messages.

You can still call accessors and handlers on it however, to handle it as you will.

### id

Returns the internal ID of the item. This may be useful for comparing since the ID is incremented each time a new item is created. Therefore, an item with ID 4 must have been logged before an item with ID 9.

### when

Returns the timestamp of when the message was logged

### message

The actual message that was stored

**level**

The severity type of this message, as well as the name of the handler that was called upon storing it.

**tag**

Returns the identification tag that was put on the message.

**shortmess**

Returns the equivalent of a `Carp::shortmess` for this item. See the `Carp` manpage for details.

**longmess**

Returns the equivalent of a `Carp::longmess` for this item, which is essentially a stack trace. See the `Carp` manpage for details.

**parent**

Returns a reference to the `Log::Message` object that stored this item. This is useful if you want to have access to the full stack in a handler.

**SEE ALSO**

*Log::Message*, *Log::Message::Handlers*, *Log::Message::Config*

**AUTHOR**

This module by Jos Boumans <kane@cpan.org>.

**Acknowledgements**

Thanks to Ann Barcomb for her suggestions.

**COPYRIGHT**

This module is copyright (c) 2002 Jos Boumans <kane@cpan.org>. All rights reserved.

This library is free software; you may redistribute and/or modify it under the same terms as Perl itself.