



Rocky Enterprise Linux 9.2 Manual Pages on command 'Log::Any::Adapter::Capture.3pm'

C:\>man Log::Any::Adapter::Capture.3pm

Log::Any::Adapter::Capture(3User Contributed Perl DocumentaLog::Any::Adapter::Capture(3pm)

NAME

Log::Any::Adapter::Capture - Adapter for capturing log messages into an arrayref

VERSION

version 1.708

SYNOPSIS

```
# temporarily redirect arrays of [ $level, $category, $message ] into an array
```

```
Log::Any::Adapter->set( { lexically => \my $scope }, Capture => to => \my @array );
```

```
# temporarily redirect just the text of log messages into an array
```

```
Log::Any::Adapter->set( { lexically => \my $scope }, Capture => text => \my @array );
```

```
# temporarily redirect the full argument list and context of each call, but only for
```

```
# log levels 'info' and above.
```

```
Log::Any::Adapter->set(
```

```
  { lexically => \my $scope },
```

```
  Capture =>
```

```
    format => 'structured',
```

```
    to => \my @array,
```

```
    log_level => 'info'
```

```
);
```

DESCRIPTION

This logging adapter provides a convenient way to capture log messages into a callback or arrayref of your choice without needing to write your own adapter. It

is intended for cases where you want to temporarily capture log messages, such as showing them to a user of your application rather than having them written to a log file.

ATTRIBUTES

to

Specify a coderef or arrayref where the messages will be delivered. The content pushed onto the array or passed to the coderef depends on "format".

format

'messages'

```
sub ( $level, $category, $message_text ) { ... }  
push @to, [ $level, $category, $message_text ];
```

This is the default format. It passes/pushes 3 arguments: the name of the log level, the logging category, and the message text as a plain string.

'text'

```
sub ( $message_text ) { ... }  
push @to, $message_text;
```

This format is the simplest, and only passes/pushes the text of the message.

'structured'

```
sub ( $level, $category, @message_parts, \%context? ) { ... }  
push @to, [ $level, $category, @message_parts, \%context? ];
```

This passes/pushes the full information available about the call to the logging method. The @message_parts are the actual arguments passed to the logging method, and if the final argument is a hashref, it is the combined "context" from the logging proxy and any overrides passed to the logging method.

log_level

Like other logging adapters, this optional argument can filter out any log messages above the specified threshold. The default is to pass through all messages regardless of level.

ATTRIBUTE ALIASES

These are not actual attributes, just shortcuts for others:

text

```
text => $dest
```

is shorthand for

format => 'text', to => \$dest

structured

structured => \$dest

is shorthand for

format => 'structured', to => \$dest

AUTHORS

? Jonathan Swartz <swartz@pobox.com>

? David Golden <dagolden@cpan.org>

? Doug Bell <preaction@cpan.org>

? Daniel Pittman <daniel@rimspace.net>

? Stephen Thirlwall <sdt@cpan.org>

COPYRIGHT AND LICENSE

This software is copyright (c) 2017 by Jonathan Swartz, David Golden, and Doug Bell.

This is free software; you can redistribute it and/or modify it under the same terms as the Perl 5 programming language system itself.

perl v5.30.0

2020-01-24

Log::Any::Adapter::Capture(3pm)