

NAME

"IO::Async::Signal" – event callback on receipt of a POSIX signal

SYNOPSIS

```
use IO::Async::Signal;

use IO::Async::Loop;
my $loop = IO::Async::Loop->new;

my $signal = IO::Async::Signal->new(
    name => "HUP",

    on_receipt => sub {
        print "I caught SIGHUP\n";
    },
);

$loop->add( $signal );

$loop->run;
```

DESCRIPTION

This subclass of IO::Async::Notifier invokes its callback when a particular POSIX signal is received.

Multiple objects can be added to a Loop that all watch for the same signal. The callback functions will all be invoked, in no particular order.

EVENTS

The following events are invoked, either using subclass methods or CODE references in parameters:

on_receipt

Invoked when the signal is received.

PARAMETERS

The following named parameters may be passed to new or configure:

name => STRING

The name of the signal to watch. This should be a bare name like TERM. Can only be given at construction time.

on_receipt => CODE

CODE reference for the on_receipt event.

Once constructed, the Signal will need to be added to the Loop before it will work.

AUTHOR

Paul Evans <leonerd@leonerd.org.uk>