

NAME

Net::DBus::Callback – a callback for receiving reactor events

SYNOPSIS

```
use Net::DBus::Callback;

# Assume we have a 'terminal' object and its got a method
# to be invoked every time there is input on its terminal.
#
# To create a callback to invoke this method one might use
my $cb = Net::DBus::Callback->new(object => $terminal,
                                method => "handle_stdin");

# Whatever is monitoring the stdin channel, would then
# invoke the callback, perhaps passing in a parameter with
# some 'interesting' data, such as number of bytes available
$cb->invoke($nbytes)

#... which results in a call to
# $terminal->handle_stdin($nbytes)
```

DESCRIPTION

This module provides a simple container for storing details about a callback to be invoked at a later date. It is used when registering to receive events from the Net::DBus::Reactor class. NB use of this module in application code is no longer necessary and it remains purely for backwards compatibility. Instead you can simply pass a subroutine code reference in any place where a callback is desired.

METHODS

```
my $cb = Net::DBus::Callback->new(method => $name, [args => \@args])
    Creates a new callback object, for invoking a plain old function. The method parameter should be the fully qualified function name to invoke, including the package name. The optional args parameter is an array reference of parameters to be pass to the callback, in addition to those passed into the invoke method.
```

```
my $cb = Net::DBus::Callback->new(object => $object, method => $name, [args => \@args])
    Creates a new callback object, for invoking a method on an object. The method parameter should be the name of the method to invoke, while the object parameter should be a blessed object on which the method will be invoked. The optional args parameter is an array reference of parameters to be pass to the callback, in addition to those passed into the invoke method.
```

```
$cb->invoke(@args)
    Invokes the callback. The argument list passed to the callback is a combination of the arguments supplied in the callback constructor, followed by the arguments supplied in the invoke method.
```

AUTHOR

Daniel P. Berrange.

COPYRIGHT

Copyright (C) 2004–2011 Daniel P. Berrange

SEE ALSO

Net::DBus::Reactor