

NAME

AnyEvent::Impl::Glib – AnyEvent adaptor for Glib

SYNOPSIS

```
use AnyEvent;
use Glib;

# this module gets loaded automatically as required
```

DESCRIPTION

This module provides transparent support for AnyEvent. You don't have to do anything to make Glib work with AnyEvent except by loading Glib before creating the first AnyEvent watcher.

Glib is probably the most inefficient event loop that has ever seen the light of the world: Glib not only scans all its watchers (really, ALL of them, whether I/O-related, timer-related or what not) during each loop iteration, it also does so multiple times and rebuilds the poll list for the kernel each time again, dynamically even. Newer versions of libglib fortunately do not call malloc/free on every single watcher invocation, though.

Glib also enforces certain undocumented behaviours, for example, you cannot always remove active child watchers, and the conditions on when it is valid to do so are not documented. Of course, if you get it wrong, you get “GLib-CRITICAL” messages. This makes it extremely hard to write “correct” glib programs, as you have to study the source code to get it right, and hope future versions don't change any internals.

AnyEvent implements the necessary workarounds, at a small performance cost.

On the positive side, and most importantly, when it works, Glib generally works correctly, no quarrels there.

If you create many watchers (as in: more than two), you might consider one of the Glib::EV, EV::Glib or Glib::Event modules that map Glib to other, more efficient, event loops.

This module uses the default Glib main context for all its watchers.

SEE ALSO

AnyEvent, Glib.

AUTHOR

Marc Lehmann <schmorp@schmorp.de>
<http://anyevent.schmorp.de>