## NAME

EV::MakeMaker − MakeMaker glue for the C−level EV API

## SYNOPSIS

This allows you to access some libevent functionality from other perl modules.

## DESCRIPTION

For optimal performance, hook into EV at the C−level. You'll need to make changes to your `Makefile.PL`, load EV in your `pm` file and add code to your `xs` / `c` file(s).

## HOW TO

### Makefile.PL

```
use EV::MakeMaker qw(ev_args);

# ... set up %args ...

WriteMakefile (ev_args (%args));
```

### extension.pm

```
use EV (); # imports optional
```

### extension.xs

```
#include "EVAPI.h"

[...]

BOOT:
  I_EV_API (HvNAME (GvSTASH (CvGV (cv))));
```

## API

See the EVAPI.h <http://cvs.schmorp.de/EV/EV/EVAPI.h> header, which you should include instead of *ev.h*.

In short, all the functions and macros from *ev.h* should work, except that the trailing underscore macros (`EV_A_`, `EV_DEFAULT_`) are not available (except `EV_P_` :).

Multiplicity is enabled.

The `data` member in each watcher is of type `SV *` and not `void *` (this might change at some point).

## EXAMPLE

The EV::Glib, EV::ADNS and Glib::EV modules all give nice examples on how to use this module.

Here are some *.xs* fragments taken from EV::ADNS that should get you going:

```
#include "EVAPI.h"

static ev_prepare pw;
static ev_idle iw;

static void
idle_cb (EV_P_ ev_idle *w, int revents)
{
  ev_idle_stop (EV_A, w);
}

MODULE = ...

BOOT:
{
  I_EV_API ("EV::ADNS");
  ev_prepare_init (&pw, prepare_cb);
```

```
  ev_init (&iw, idle_cb); ev_set_priority (&iw, EV_MINPRI);
  ev_idle_start (EV_DEFAULT, &iw);
}
```