NAME

B::C - Perl compiler's C backend

SYNOPSIS

```
perl -MO=C[,OPTIONS] foo.pl
```

DESCRIPTION

This compiler backend takes Perl source and generates C source code corresponding to the internal structures that perl uses to run your program. When the generated C source is compiled and run, it cuts out the time which perl would have taken to load and parse your program into its internal semi-compiled form. That means that compiling with this backend will not help improve the runtime execution speed of your program but may improve the start-up time. Depending on the environment in which your program runs this may be either a help or a hindrance.

OPTIONS

If there are any non-option arguments, they are taken to be names of objects to be saved (probably doesn't work properly yet). Without extra arguments, it saves the main program.

-ofilename

> Output to filename instead of STDOUT

-v

> Verbose compilation (currently gives a few compilation statistics).

--

> Force end of options

-uPackname

> Force apparently unused subs from package Packname to be compiled. This allows programs to use eval "foo()" even when sub foo is never seen to be used at compile time. The down side is that any subs which really are never used also have code generated. This option is necessary, for example, if you have a signal handler foo which you initialise with $SIG{BAR} = "foo". A better fix, though, is just to change it to $SIG{BAR} = \&foo. You can have multiple **-u** options. The compiler tries to figure out which packages may possibly have subs in which need compiling but the current version doesn't do it very well. In particular, it is confused by nested packages (i.e. of the form A::B) where package A does not contain any subs.

-D

> Debug options (concatenated or separate flags like perl -D).

-Do

> OPs, prints each OP as it's processed

-Dc

> COPs, prints COPs as processed (incl. file & line num)

-DA

> prints AV information on saving

-DC

> prints CV information on saving

-DM

> prints MAGIC information on saving

-f

> Force options/optimisations on or off one at a time. You can explicitly disable an option using **-fno-option**. All options default to **disabled**.
>
> **-fcog**
>
>> Copy-on-grow: PVs declared and initialised statically.
>
> **-fsave-data**
>
>> Save package::DATA filehandles (only available with PerlIO).
>
> **-fppaddr**
>
>> Optimize the initialization of op_ppaddr.
>
> **-fwarn-sv**
>
>> Optimize the initialization of cop_warnings.
>
> **-fuse-script-name**
>
>> Use the script name instead of the program name as $0.
>
> **-fsave-sig-hash**
>
>> Save compile-time modifications to the %SIG hash.

-On

> Optimisation level (n = 0, 1, 2, ...). **-O** means **-O1**.
>
> **-O0**
>
>> Disable all optimizations.
>
> **-O1**
>
>> Enable **-fcog**.
>
> **-O2**
>
>> Enable **-fppaddr**, **-fwarn-sv**.

-llimit

> Some C compilers impose an arbitrary limit on the length of string constants (e.g. 2048 characters for Microsoft Visual C++). The **-llimit** options tells the C backend not to generate string literals exceeding that limit.

EXAMPLES

```
perl -MO=C,-ofoo.c foo.pl
perl cc_harness -o foo foo.c
```

Note that `cc_harness` lives in the `B` subdirectory of your perl library directory. The utility called `perlcc` may also be used to help make use of this compiler.

```
perl -MO=C,-v,-DcA,-l2048 bar.pl > /dev/null
```

BUGS

Plenty. Current status: experimental.

AUTHOR

Malcolm Beattie, `mbeattie@sable.ox.ac.uk`