NAME

perlcc - generate executables from Perl programs

SYNOPSIS

```
    $ perlcc hello              # Compiles into executable 'a.out'
    $ perlcc -o hello hello.pl  # Compiles into executable 'hello'

    $ perlcc -O file            # Compiles using the optimised C backend
    $ perlcc -B file            # Compiles using the bytecode backend

    $ perlcc -c file            # Creates a C file, 'file.c'
    $ perlcc -S -o hello file   # Creates a C file, 'file.c',
                                # then compiles it to executable 'hello'
    $ perlcc -c out.c file      # Creates a C file, 'out.c' from 'file'

    $ perlcc -e 'print q//'     # Compiles a one-liner into 'a.out'
    $ perlcc -c -e 'print q//'  # Creates a C file 'a.out.c'

    $ perlcc -I /foo hello # extra headers (notice the space after -I)
    $ perlcc -L /foo hello # extra libraries (notice the space after -L)

    $ perlcc -r hello           # compiles 'hello' into 'a.out', runs
'a.out'.
    $ perlcc -r hello a b c     # compiles 'hello' into 'a.out', runs
'a.out'.
                                # with arguments 'a b c'

    $ perlcc hello -log c       # compiles 'hello' into 'a.out' logs
compile
                                # log into 'c'.
```

DESCRIPTION

perlcc creates standalone executables from Perl programs, using the code generators provided by the *B* module. At present, you may either create executable Perl bytecode, using the `-B` option, or generate and compile C files using the standard and 'optimised' C backends.

The code generated in this way is not guaranteed to work. The whole codegen suite (`perlcc` included) should be considered **very** experimental. Use for production purposes is strongly discouraged.

OPTIONS

-L*ibrary directories*

Adds the given directories to the library search path when C code is passed to your C compiler.

-I*include directories*

Adds the given directories to the include file search path when C code is passed to your C compiler; when using the Perl bytecode option, adds the given directories to Perl's include path.

-o *output file name*

Specifies the file name for the final compiled executable.

-c *C file name*

Create C code only; do not compile to a standalone binary.

-e *perl code*

Compile a one-liner, much the same as `perl -e '...'`

-S

Do not delete generated C code after compilation.

-B

Use the Perl bytecode code generator.

-O

Use the 'optimised' C code generator. This is more experimental than everything else put together, and the code created is not guaranteed to compile in finite time and memory, or indeed, at all.

-v

Increase verbosity of output; can be repeated for more verbose output.

-r

Run the resulting compiled script after compiling it.

-log

Log the output of compiling to a file rather than to stdout.