NAME

O - Generic interface to Perl Compiler backends

SYNOPSIS

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

DESCRIPTION

This is the module that is used as a frontend to the Perl Compiler.

If you pass the `-q` option to the module, then the STDOUT filehandle will be redirected into the variable `$O::BEGIN_output` during compilation. This has the effect that any output printed to STDOUT by BEGIN blocks or use'd modules will be stored in this variable rather than printed. It's useful with those backends which produce output themselves (`Deparse`, `Concise` etc), so that their output is not confused with that generated by the code being compiled.

The `-qq` option behaves like `-q`, except that it also closes STDERR after deparsing has finished. This suppresses the "Syntax OK" message normally produced by perl.

CONVENTIONS

Most compiler backends use the following conventions: OPTIONS consists of a comma-separated list of words (no white-space). The `-v` option usually puts the backend into verbose mode. The `-ofile` option generates output to **file** instead of stdout. The `-D` option followed by various letters turns on various internal debugging flags. See the documentation for the desired backend (named `B::Backend` for the example above) to find out about that backend.

IMPLEMENTATION

This section is only necessary for those who want to write a compiler backend module that can be used via this module.

The command-line mentioned in the SYNOPSIS section corresponds to the Perl code

```
use O ("Backend", OPTIONS);
```

The `O::import` function loads the appropriate `B::Backend` module and calls its `compile` function, passing it OPTIONS. That function is expected to return a sub reference which we'll call CALLBACK. Next, the "compile-only" flag is switched on (equivalent to the command-line option `-c`) and a CHECK block is registered which calls CALLBACK. Thus the main Perl program mentioned on the command-line is read in, parsed and compiled into internal syntax tree form. Since the `-c` flag is set, the program does not start running (excepting BEGIN blocks of course) but the CALLBACK function registered by the compiler backend is called.

In summary, a compiler backend module should be called "B::Foo" for some foo and live in the appropriate directory for that name. It should define a function called `compile`. When the user types

```
perl -MO=Foo,OPTIONS foo.pl
```

that function is called and is passed those OPTIONS (split on commas). It should return a sub ref to the main compilation function. After the user's program is loaded and parsed, that returned sub ref is invoked which can then go ahead and do the compilation, usually by making use of the `B` module's functionality.

BUGS

The `-q` and `-qq` options don't work correctly if perl isn't compiled with PerlIO support : STDOUT will be closed instead of being redirected to `$O::BEGIN_output`.

AUTHOR

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