NAME

B::Xref - Generates cross reference reports for Perl programs

SYNOPSIS

perl -MO=Xref[,OPTIONS] foo.pl

DESCRIPTION

The B::Xref module is used to generate a cross reference listing of all definitions and uses of variables, subroutines and formats in a Perl program. It is implemented as a backend for the Perl compiler.

The report generated is in the following format:

```
File filename1
Subroutine subname1
Package package1
object1 line numbers
object2 line numbers
...
Package package2
...
```

Each **File** section reports on a single file. Each **Subroutine** section reports on a single subroutine apart from the special cases ''(definitions)'' and ''(main)''. These report, respectively, on subroutine definitions found by the initial symbol table walk and on the main part of the program or module external to all subroutines.

The report is then grouped by the **Package** of each variable, subroutine or format with the special case ''(lexicals)'' meaning lexical variables. Each **object** name (implicitly qualified by its containing **Package**) includes its type character(s) at the beginning where possible. Lexical variables are easier to track and even included dereferencing information where possible.

The `line numbers` are a comma separated list of line numbers (some preceded by code letters) where that object is used in some way. Simple uses aren't preceded by a code letter. Introductions (such as where a lexical is first defined with `my`) are indicated with the letter ''i''. Subroutine and method calls are indicated by the character ''&''. Subroutine definitions are indicated by ''s'' and format definitions by ''f''.

For instance, here's part of the report from the *pod2man* program that comes with Perl:

```
Subroutine clear_noremap
Package (lexical)
$ready_to_print i1069, 1079
Package main
$& 1086
$. 1086
$0 1086
$1 1087
$2 1085, 1085
$3 1085, 1085
$ARGV 1086
%HTML_Escapes 1085, 1085
```

This shows the variables used in the subroutine `clear_noremap`. The variable `$ready_to_print` is a *my()* (lexical) variable, **i**ntroduced (first declared with *my()*) on line 1069, and used on line 1079. The variable `$&` from the main package is used on 1086, and so on.

A line number may be prefixed by a single letter:

i Lexical variable introduced (declared with *my()*) for the first time.

& Subroutine or method call.

s Subroutine defined.

r Format defined.

The most useful option the cross referencer has is to save the report to a separate file. For instance, to save the report on *myperlprogram* to the file *report*:

```
$ perl -MO=Xref,-oreport myperlprogram
```

OPTIONS

Option words are separated by commas (not whitespace) and follow the usual conventions of compiler backend options.

`-oFILENAME`
: Directs output to `FILENAME` instead of standard output.

`-r`
: Raw output. Instead of producing a human-readable report, outputs a line in machine-readable form for each definition/use of a variable/sub/format.

`-d`
: Don't output the ''(definitions)'' sections.

`-D[tO]`
: (Internal) debug options, probably only useful if `-r` included. The `t` option prints the object on the top of the stack as it's being tracked. The `O` option prints each operator as it's being processed in the execution order of the program.

BUGS

Non-lexical variables are quite difficult to track through a program. Sometimes the type of a non-lexical variable's use is impossible to determine. Introductions of non-lexical non-scalars don't seem to be reported properly.

AUTHOR

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