Perl debugger

Nic Herndon

August 11, 2014

How to start the debugger

- Just like a Perl script can be run with

  ```
  $ script.pl [args]
  ```

  or

  ```
  $ perl script.pl [args]
  ```

- Perl debugger is invoked with

  ```
  $ perl -d script.pl [args]
  ```

- Example

  ```
  $ perl -d ./rename_fa.pl ms_assembly.txt ms_ogs_proteins.fa
  ready_ms_ogs_proteins.fa
  Loading DB routines from perl5db.pl version 1.37
  Editor support available.
  Enter h or 'h h' for help, or 'man perldebug' for more help.
  main::(./rename_fa.pl:8):  my %map;
  DB<1>
  ```

Debugger help

```
$ man perldebug
```

- If you invoke Perl with the **-d** switch, your script runs under the Perl source debugger.
- Your code must first compile correctly for the debugger to work on it.
- This works like an interactive Perl environment, prompting for debugger commands that let you
  - examine source code,
  - set breakpoints,
  - get stack backtraces,
  - change the values of variables, etc.
- The program will halt right before the first run-time executable statement and ask you to enter a debugger command. Contrary to popular expectations, whenever the debugger halts and shows you a line of code, it always displays the line it's about to execute, rather than the one it has just executed.

Debugger help

```
  DB<1> h
List/search source lines:               Control script execution:
  l [ln|sub]  List source code            T           Stack trace
  - or .      List previous/current line  s [expr]    Single step [in expr]
  v [line]    View around line            n [expr]    Next, steps over subs
  f filename  View source in file         <CR/Enter>  Repeat last n or s
  /pattern/ ?patt?   Search forw/backw    r           Return from subroutine
  M           Show module versions        c [ln|sub]  Continue until position
Debugger controls:                        L           List break/watch/actions
  o [...]     Set debugger options        t [n] [expr] Toggle trace [max depth] ][trace expr]
  <[<]|{[{]|>[>] [cmd] Do pre/post-prompt b [ln|event|sub] [cnd] Set breakpoint
  ! [N|pat]   Redo a previous command     B ln|*      Delete a/all breakpoints
  H [-num]    Display last num commands   a [ln] cmd  Do cmd before line
  = [a val]   Define/list an alias        A ln|*      Delete a/all actions
  h [db_cmd]  Get help on command         w expr      Add a watch expression
  h h         Complete help page          W expr|*    Delete a/all watch exprs
  |[|]db_cmd  Send output to pager        ![!] syscmd Run cmd in a subprocess
  q or ^D     Quit                        R           Attempt a restart
Data Examination:     expr     Execute perl code, also see: s,n,t expr
  x|m expr       Evals expr in list context, dumps the result or lists methods.
  p expr         Print expression (uses script's current package).
  S [[!]pat]     List subroutine names [not] matching pattern
  V [Pk [Vars]]  List Variables in Package.  Vars can be ~pattern or !pattern.
  X [Vars]       Same as "V current_package [Vars]".  i class inheritance tree.
  y [n [Vars]]   List lexicals in higher scope <n>.  Vars same as V.
  e     Display thread id     E Display all thread ids.
For more help, type h cmd_letter, or run man perldebug for all docs.
  DB<1>
```

Arrows issue

"When I use the perl debugger, the up/down/left/right keys don't work. I get characters like these '^[[A^[[B^[[D^[[C' when I use those keys" (http://www.perlmonks.org/?node_id=743947).

```
$ sudo cpan
cpan[1]> install Term::ReadLine::Perl
cpan[2]> install Term::ReadKey
```