```
    scalar context return from CODE(0x182528): undef
    Package lib/Config.pm.
    in  $=Config::TIEHASH('Config') from lib/Config.pm:628
    out $=Config::TIEHASH('Config') from lib/Config.pm:628
    scalar context return from Config::TIEHASH:   empty hash
    in  $=Exporter::import('Config', 'myconfig', 'config_vars') from /dev/null:0
     in  $=Exporter::export('Config', 'main', 'myconfig', 'config_vars') from lib/Exporter.pm:171
     out $=Exporter::export('Config', 'main', 'myconfig', 'config_vars') from lib/Exporter.pm:171
     scalar context return from Exporter::export: ''
    out $=Exporter::import('Config', 'myconfig', 'config_vars') from /dev/null:0
    scalar context return from Exporter::import: ''
```

In all cases shown above, the line indentation shows the call tree. If bit 2 of `frame` is set, a line is printed on exit from a subroutine as well. If bit 4 is set, the arguments are printed along with the caller info. If bit 8 is set, the arguments are printed even if they are tied or references. If bit 16 is set, the return value is printed, too.

When a package is compiled, a line like this

```
    Package lib/Carp.pm.
```

is printed with proper indentation.

66.4 Debugging regular expressions

There are two ways to enable debugging output for regular expressions.

If your perl is compiled with −DDEBUGGING, you may use the **-Dr** flag on the command line.

Otherwise, one can use `re 'debug'`, which has effects at compile time and run time. It is not lexically scoped.

66.4.1 Compile-time output

The debugging output at compile time looks like this:

```
  Compiling REx '[bc]d(ef*g)+h[ij]k$'
  size 45 Got 364 bytes for offset annotations.
  first at 1
  rarest char g at 0
  rarest char d at 0
     1: ANYOF[bc](12)
    12: EXACT <d>(14)
    14: CURLYX[0] {1,32767}(28)
    16:   OPEN1(18)
    18:     EXACT <e>(20)
    20:     STAR(23)
    21:       EXACT <f>(0)
    23:     EXACT <g>(25)
    25:   CLOSE1(27)
    27:   WHILEM[1/1](0)
    28: NOTHING(29)
    29: EXACT <h>(31)
    31: ANYOF[ij](42)
    42: EXACT <k>(44)
    44: EOL(45)
    45: END(0)
  anchored 'de' at 1 floating 'gh' at 3..2147483647 (checking floating)
```