Perl
“Source Filters”
Source Filters

- What are “Source Filters”
- How to use them
- What can you use them for
- Some examples of CPAN modules which use them
- Things to watch out for when using Source Filters
What are Source Filters?

- Source filters alter the program text of a module before Perl sees it.
- Similar to C Preprocessor.
- The original purpose of source filters was to let you encrypt your program source to prevent casual piracy.
How to use Source Filters

- Three ways:
  1. You can write it in C,
  2. Use an external program as a filter
  3. Write the filter in Perl

- I will deal with writing a filter in Perl
Filter::Util::Call

- Filter::Util::Call is the old way

package Whatever;
use Filter::Util::Call;
sub import {
    filter_add();
}
sub filter {
    .. change $_[..]
}
;
Filter::Simple is the new better way
package MyFilter;
use Filter::Simple sub { .. alter $._ .. } 
or
package MyFilter;
use Filter::Simple;
FILTER {
.. do something with $._ ..
}
Case Study - Trace.pm

A tracing Module which prints out tracing info based on the contents of $ENV{TRACE}

package Trace;

use Env qw/ $TRACE /;
use vars qw($VERSION $AUTOLOAD);

sub AUTOLOAD
{
    $AUTOLOAD =~ /.*::\w+/ or die "No such subroutine: $AUTOLOAD";
    output_message("Trace::$1", @_) if trace_this($1);
}

sub trace_this {
    my ($trace_what) = @_; 
    return 0 unless defined $TRACE;
    return $TRACE =~ m/$trace_what/bi;
}

sub output_message {
    my ($trace, @message) = @_; 
    print join(":", @_), "\n";
}

#!/usr/bin/env perl
use Trace;

Trace::WHATEVER("A Message");
What can go wrong?

Consider:

```perl
foreach (1 .. 10000) {
    Trace::SOMETHING(“a message”);
}
```

Each time through the loop, the contents of $ENV{TRACE} is examined to decide whether to out a message or not.
package Trace;
use strict;
use warnings;
use Env qw/ $TRACE /;
$TRACE =~ s/,/\|/g;  # replace commas with pipe for regex
use Filter::Simple sub {
    s/Trace::/#race::/g;
    s/#race::($TRACE)/Trace::$1/g;
}

Now all the Trace::WHATEVER lines are commented out apart from those in $ENV{TRACE}
CPAN Modules which use Source Filters

- Acme-Abuse
- Acme-Aspartame
- Acme-Beatnik
- Acme-BottomsUp
- Acme-Brainfuck
- Acme-ButFirst
- Acme-Canadian
- Acme-ComeFrom
- Acme-Comment
- Acme-Currency
- Acme-Echo
- Acme-Filter-Kenny
- Acme-Gosub
- Acme-Hyperindex
- Acme-Lingua-NIGERIAN
- Acme-Lingua-Pirate-Perl
- Acme-Lingua-String-Perl
- Acme-Metification
- Acme-No
- Acme-Pythonic
- Acme-Rant
- Acme-Sub-Parms
- Acme-Tiroler
- Acme-Ukrop
- Acme-Valid
- Acme-Your
- Apache-AntiSpam
- Apache-Clickable
- Apache-XPP
- Attribute-Handlers-Prospective
- AxKit
- Backtick-AutoChomp
- Blossom-Debug
- Blossom-Include
- Cac
- Cache-Cascade
- Call-Immediate
- CatalystX-Imports
- Class-HPLOO
- Class-Root
- Class-Workflow
- Code-Dumper
- Config-Vars
- Crypt-License
- Crypt-Util
- Data-Dumper-Simple
- DateTime-Event-Klaxon
- Debug-Mixin
- Decode-Source
- Devel-StealthDebug
- Devel-XRay
- Easy-Log
- Embperl
- Error-TryCatch
- ExtUtils-configPL
- For-Else
- Gimp-ScriptFu-Client
- IfLoop
- Inline-Files
- Inline-YAML
- Language-Indonesia
- Lexical-Attributes
- Lingua-Romana-Perligata
- Lingua-Shakespeare
- Lingua-Sinica-PerlYuYan
- Lingua-Tolkien-BURL
- Log-Easy
- Logic
- MOP
- Macrame
- Macro
- Mail-FilterXML
- Markup-Perl
- Module-Compile
- Moose
- MooseX-POE
- Net-IP-Match
- Net-SIP
- OOPS
- ObjectivePerl
- PDL
- PDL-NiceSlice
- PHP-Include
- POE
- POE-Component-Client-FTP
- POE-Component-Jabber
- POE-Component-OSCAR
- POE-Filter-XML
- Parse-Native
- Perl-Unsafe-Signals
- Perl6-Attributes
- Perl6-Binding
- Perl6-Export
- Perl6-Parameters
- Perl6-Perldoc-v0.0.2
- Perl6-Perldoc-v0.0.3
- Perl6-Perldoc-v0.0.4
- Perl6-Perldoc-v0.0.5
- Perl6-Placeholders
- Perl6-Pugs
- Perl6-Rules
- Perl6-Subs
- Perl6-Variables
- Plagger
- PlusPlus
- Poem
- Querylet
- Ruby
- SQL-Interpolate
- SQL-Preproc
- SWISH-Filter
- Semi-Semicolons
- Shell-POSIX-Select
- SiePerl
- Smart-Comments
- Smart-Comments-v1.0.2
- Smart-Comments-v1.0.3
- Spiffy
- String-ReccStack
- Sub-Lambda
- Sub-Lexical
- Sub-Methodical
- Sub-Quotelike
- Sub-Regexp
- Sub-Signatures
- SuperPython
- Switch
- Test-AtRuntime
- Test-Cookbook
- Test-HTTP
- Test-LectroTest
- Test-MultiFork
- Test-Standalone
- Text-RewriteRules
- Thread-Synchronized
- Tie-VecArray
- Toolkit
- Unicode-Overload
- Weed
- XML-Filter-Essex
- XML-XSH
- XML-XSH2
- Yada-Yada-Yada
- fields-aliased
- forks
- htpl
- ifdef
- interface
- kurila
- namespace-clean
Things to watch out for

- Line numbers
- Introducing syntax errors in your filter