

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

- 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 $TRACE =~ m/\b($trace_what)\b/i;
}

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

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

Trace::WHATEVER("A Message");
```

What can go wrong ?

Consider:

```
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

How can Source Filters help ?

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

Things to watch out for

- Line numbers
- Introducing syntax errors in your filter