NAME

ExtUtils::ParseXS - converts Perl XS code into C code

SYNOPSIS

```
use ExtUtils::ParseXS;

my $pxs = ExtUtils::ParseXS->new;
$pxs->process_file( filename => 'foo.xs' );

$pxs->process_file( filename => 'foo.xs',
                    output => 'bar.c',
                    'C++' => 1,
                    typemap => 'path/to/typemap',
                    hiertype => 1,
                    except => 1,
                    versioncheck => 1,
                    linenumbers => 1,
                    optimize => 1,
                    prototypes => 1,
                  );

# Legacy non-OO interface using a singleton:
use ExtUtils::ParseXS qw(process_file);
process_file( filename => 'foo.xs' );
```

DESCRIPTION

`ExtUtils::ParseXS` will compile XS code into C code by embedding the constructs necessary to let C functions manipulate Perl values and creates the glue necessary to let Perl access those functions. The compiler uses typemaps to determine how to map C function parameters and variables to Perl values.

The compiler will search for typemap files called *typemap*. It will use the following search path to find default typemaps, with the rightmost typemap taking precedence.

```
../../../typemap:../../typemap:../typemap:typemap
```

EXPORT

None by default. `process_file()` and/or `report_error_count()` may be exported upon request. Using the functional interface is discouraged.

METHODS

$pxs->new()

Returns a new, empty XS parser/compiler object.

$pxs->process_file()

This method processes an XS file and sends output to a C file. The method may be called as a function (this is the legacy interface) and will then use a singleton as invocant.

Named parameters control how the processing is done. The following parameters are accepted:

C++

Adds `extern "C"` to the C code. Default is false.

hiertype

Retains `::` in type names so that C++ hierarchical types can be mapped. Default is false.

except

Adds exception handling stubs to the C code. Default is false.

typemap

Indicates that a user-supplied typemap should take precedence over the default typemaps. A single typemap may be specified as a string, or multiple typemaps can be specified in an array reference, with the last typemap having the highest precedence.

prototypes

Generates prototype code for all xsubs. Default is false.

versioncheck

Makes sure at run time that the object file (derived from the `.xs` file) and the `.pm` files have the same version number. Default is true.

linenumbers

Adds `#line` directives to the C output so error messages will look like they came from the original XS file. Default is true.

optimize

Enables certain optimizations. The only optimization that is currently affected is the use of *target*s by the output C code (see *perlguts*). Not optimizing may significantly slow down the generated code, but this is the way **xsubpp** of 5.005 and earlier operated. Default is to optimize.

inout

Enable recognition of `IN`, `OUT_LIST` and `INOUT_LIST` declarations. Default is true.

argtypes

Enable recognition of ANSI-like descriptions of function signature. Default is true.

s

Maintainer note: I have no clue what this does. Strips function prefixes?

$pxs->report_error_count()

This method returns the number of [a certain kind of] errors encountered during processing of the XS file.

The method may be called as a function (this is the legacy interface) and will then use a singleton as invocant.

AUTHOR

Based on xsubpp code, written by Larry Wall.

Maintained by:

- Ken Williams, <ken@mathforum.org>
- David Golden, <dagolden@cpan.org>
- James Keenan, <jkeenan@cpan.org>
- Steffen Mueller, <smueller@cpan.org>

COPYRIGHT

SEE ALSO

perl, ExtUtils::xsubpp, ExtUtils::MakeMaker, *perlxs*, *perlxstut*.