NAME

ExtUtils::Constant - generate XS code to import C header constants

SYNOPSIS

```
    use ExtUtils::Constant qw (WriteConstants);
    WriteConstants(
        NAME => 'Foo',
        NAMES => [qw(FOO BAR BAZ)],
    );
    # Generates wrapper code to make the values of the constants FOO BAR
BAZ
    #  available to perl
```

DESCRIPTION

ExtUtils::Constant facilitates generating C and XS wrapper code to allow perl modules to AUTOLOAD constants defined in C library header files. It is principally used by the `h2xs` utility, on which this code is based. It doesn't contain the routines to scan header files to extract these constants.

USAGE

Generally one only needs to call the `WriteConstants` function, and then

```
    #include "const-c.inc"
```

in the C section of `Foo.xs`

```
    INCLUDE: const-xs.inc
```

in the XS section of `Foo.xs`.

For greater flexibility use `constant_types()`, `C_constant` and `XS_constant`, with which `WriteConstants` is implemented.

Currently this module understands the following types. h2xs may only know a subset. The sizes of the numeric types are chosen by the `Configure` script at compile time.

IV

signed integer, at least 32 bits.

UV

unsigned integer, the same size as *IV*

NV

floating point type, probably `double`, possibly `long double`

PV

NUL terminated string, length will be determined with `strlen`

PVN

A fixed length thing, given as a [pointer, length] pair. If you know the length of a string at compile time you may use this instead of *PV*

SV

A **mortal** SV.

YES

Truth. (`PL_sv_yes`) The value is not needed (and ignored).

NO

Defined Falsehood. (`PL_sv_no`) The value is not needed (and ignored).

UNDEF

`undef`. The value of the macro is not needed.

FUNCTIONS

constant_types

A function returning a single scalar with `#define` definitions for the constants used internally between the generated C and XS functions.

XS_constant PACKAGE, TYPES, XS_SUBNAME, C_SUBNAME

A function to generate the XS code to implement the perl subroutine *PACKAGE*::constant used by *PACKAGE*::AUTOLOAD to load constants. This XS code is a wrapper around a C subroutine usually generated by `C_constant`, and usually named `constant`.

TYPES should be given either as a comma separated list of types that the C subroutine `constant` will generate or as a reference to a hash. It should be the same list of types as `C_constant` was given. [Otherwise `XS_constant` and `C_constant` may have different ideas about the number of parameters passed to the C function `constant`]

You can call the perl visible subroutine something other than `constant` if you give the parameter *XS_SUBNAME*. The C subroutine it calls defaults to the name of the perl visible subroutine, unless you give the parameter *C_SUBNAME*.

autoload PACKAGE, VERSION, AUTOLOADER

A function to generate the AUTOLOAD subroutine for the module *PACKAGE VERSION* is the perl version the code should be backwards compatible with. It defaults to the version of perl running the subroutine. If *AUTOLOADER* is true, the AUTOLOAD subroutine falls back on AutoLoader::AUTOLOAD for all names that the constant() routine doesn't recognise.

WriteMakefileSnippet

WriteMakefileSnippet ATTRIBUTE => VALUE [, ...]

A function to generate perl code for Makefile.PL that will regenerate the constant subroutines. Parameters are named as passed to `WriteConstants`, with the addition of `INDENT` to specify the number of leading spaces (default 2).

Currently only `INDENT`, `NAME`, `DEFAULT_TYPE`, `NAMES`, `C_FILE` and `XS_FILE` are recognised.

WriteConstants ATTRIBUTE => VALUE [, ...]

Writes a file of C code and a file of XS code which you should `#include` and `INCLUDE` in the C and XS sections respectively of your module's XS code. You probably want to do this in your `Makefile.PL`, so that you can easily edit the list of constants without touching the rest of your module. The attributes supported are

NAME

Name of the module. This must be specified

DEFAULT_TYPE

The default type for the constants. If not specified `IV` is assumed.

BREAKOUT_AT

The names of the constants are grouped by length. Generate child subroutines for each group with this number or more names in.

NAMES

An array of constants' names, either scalars containing names, or hashrefs as detailed in *C_constant*.

PROXYSUBS

If true, uses proxy subs. See *ExtUtils::Constant::ProxySubs*.

C_FH

A filehandle to write the C code to. If not given, then *C_FILE* is opened for writing.

C_FILE

The name of the file to write containing the C code. The default is `const-c.inc`. The `-` in the name ensures that the file can't be mistaken for anything related to a legitimate perl package name, and not naming the file `.c` avoids having to override Makefile.PL's `.xs` to `.c` rules.

XS_FH

A filehandle to write the XS code to. If not given, then *XS_FILE* is opened for writing.

XS_FILE

The name of the file to write containing the XS code. The default is `const-xs.inc`.

XS_SUBNAME

The perl visible name of the XS subroutine generated which will return the constants. The default is `constant`.

C_SUBNAME

The name of the C subroutine generated which will return the constants. The default is *XS_SUBNAME*. Child subroutines have _ and the name length appended, so constants with 10 character names would be in `constant_10` with the default *XS_SUBNAME*.

AUTHOR

Nicholas Clark <nick@ccl4.org> based on the code in `h2xs` by Larry Wall and others