

#### NAME

h2ph - convert .h C header files to .ph Perl header files

#### **SYNOPSIS**

h2ph [-d destination directory] [-r | -a] [-l] [headerfiles]

#### DESCRIPTION

*h2ph* converts any C header files specified to the corresponding Perl header file format. It is most easily run while in /usr/include:

```
cd /usr/include; h2ph * sys/*

or
  cd /usr/include; h2ph * sys/* arpa/* netinet/*

or
  cd /usr/include; h2ph -r -l .
```

The output files are placed in the hierarchy rooted at Perl's architecture dependent library directory. You can specify a different hierarchy with a **-d** switch.

If run with no arguments, filters standard input to standard output.

#### **OPTIONS**

-d destination dir

Put the resulting .ph files beneath **destination\_dir**, instead of beneath the default Perl library location (\$Config{'installsitearch'}).

-r

Run recursively; if any of **headerfiles** are directories, then run *h2ph* on all files in those directories (and their subdirectories, etc.). **-r** and **-a** are mutually exclusive.

-a

Run automagically; convert **headerfiles**, as well as any **.h** files which they include. This option will search for **.h** files in all directories which your C compiler ordinarily uses. **-a** and **-r** are mutually exclusive.

-1

Symbolic links will be replicated in the destination directory. If **-I** is not specified, then links are skipped over.

-h

Put 'hints' in the .ph files which will help in locating problems with *h2ph*. In those cases when you **require** a **.ph** file containing syntax errors, instead of the cryptic

```
[ some error condition ] at (eval mmm) line nnn
you will see the slightly more helpful
[ some error condition ] at filename.ph line nnn
```

However, the .ph files almost double in size when built using -h.

-D

Include the code from the .h file as a comment in the .ph file. This is primarily used for



debugging h2ph.

-Q

'Quiet' mode; don't print out the names of the files being converted.

### **ENVIRONMENT**

No environment variables are used.

### **FILES**

```
/usr/include/*.h
/usr/include/sys/*.h
etc.
```

# **AUTHOR**

Larry Wall

## **SEE ALSO**

perl(1)

### **DIAGNOSTICS**

The usual warnings if it can't read or write the files involved.

### **BUGS**

Doesn't construct the %sizeof array for you.

It doesn't handle all C constructs, but it does attempt to isolate definitions inside evals so that you can get at the definitions that it can translate.

It's only intended as a rough tool. You may need to dicker with the files produced.

You have to run this program by hand; it's not run as part of the Perl installation.

Doesn't handle complicated expressions built piecemeal, a la:

```
enum {
FIRST_VALUE,
SECOND_VALUE,
    #ifdef ABC
THIRD_VALUE
    #endif
};
```

Doesn't necessarily locate all of your C compiler's internally-defined symbols.