NAME

ppmkpat - make a pattern (pattern prepass, optimizer and compiler)

SYNOPSIS

ppmkpat [patname] [options] < patdefinition

DESCRIPTION

Ppmkpat creates (makes) common pattern package patterns from pattern definitions and previously defined (predefined) patterns.

The following is a brief description of the command line options:

+d The next argument is a pattern directory to be added to the directory search order.

+fo Create the pattern in object format.

+fs Create the pattern in standard format (this is the default if **+fo** is not present).

+ipok Output to stderr the characters **IP** when ppmkpat starts and the characters **OK** when ppmkpat finishes if the pattern was created with no errors.

+p Perform only the prepass on the pattern definition and output the results to stdout. This is the same as performing the cc(1) command with the -E option.

-p The prepass is not performed on the pattern definition.

+r Restrict the definition to a subset of the normally allowed built-in patterns. This is used by the ppsccsgp(3L) subroutine to prevent the use of special built-in patterns (e.g., frepeat and eofrpt). The pattern compiler source header files should be consulted for a list of the specific built-in patterns which are affected.

+t Perform a translation by mapping lower case string characters into upper case characters. This translation is only performed on string argument characters.

-s Do not include pattern source definition in the compiled output. This can be used to save space, and increase security.

+s Include pattern source definition in the compiled output (done by default).

-D Similar to the -D option of cc(1). May be used as -Dsymbol or -Dsymbol=value, where **value** is a number, or an unquoted

string. The #if - #endif, and #ifdef - #endif preprocessor directives will recognize these **-D** options. Multiple **-D** options are allowed.

Ppmkpat reads the the pattern definition <patdefinition> from standard input <stdin>. The predefined patterns are read from files. The directories which are searched for the predefined patterns are controlled by **+d** options in the command line. If no **+d** options are specified then the following directory search order is used:

/keypat	**builtin pattern/keyword/primitives directory**
.	**present working directory**
/compat	**common pattern directory**
/usr/pat	**common user pattern directory**

If one or more **+d** options are present in the command line, then all of the default directories will be removed from the search order except **/keypat**. The keyword directory (/keypat) may never be removed from the search order.

The argument directly following a **+d** (**+d** <dirname>) is the path name of a directory. This path name is added to the end of the search order list.

The following example should explain the [redacted] **+d** option discussed above:

ppmkpat +d . +d /type01/pat +d /compat +d /usr/pat

For the command line above the directory search order will be:

/keypat
. **(present working directory)**
/type01/pat
/compat
/usr/pat

Ppmkpat creates a pattern with one of several formats.

Standard format - This form will be produced by default (i.e. no **+fo** in the command line).

Object format - This form will be produced if the **+fo** option is used in the command line.

Ppmkpat puts its compiled output (the pattern) into a file. If **patname** is specified in the command line, then some characters are appended to the end of patname and used as the name of the

pattern file. If **patname** is not specified, then a default name (PPDFLTNAM as defined in the **/usr/include/ppsubs.h** header file) is appended with two characters and used as the name of the file. The characters which are appended to the file name are **.p** for standard format and **.o** for object format.

The object formated file may be link loaded into an **a.out** file like any other object file.

FILES

/keypat	builtin pattern/keyword/primitives directory
/compat	common pattern directory
/usr/pat	common user pattern directory
temp.p	default standard output pattern file
temp.o	default object format (both types) pattern file
/tmp/ppsrc<prid>.c	temporary file for prepass; <prid> = process ID

SEE ALSO

a.out(5), pattern(5L) ppdpat(1L)

DIAGNOSTICS

The diagnostics produced by **ppmkpat** are intended to be self explanatory. Ppmkpat exits with value PPSYNTAXERR if one or more errors were found in the definition.

BUGS

Ppmkpat will accept a long (over 60 characters in length) **patname**. However the object formats use the first seven characters after the last / character in **patname** as the **C** symbol name of the pattern (e.g. if **patname** = **/compat/sctab012**, then **sctab01** will be the symbol name of the pattern when it is part of an **a.out** file).

FILES

/keypat	builtin pattern/keyword/primitives directory
/compat	common pattern directory
/usr/pat	common user pattern directory
temp.p	default standard output pattern file
temp.o	default object format (both types) pattern file
/tmp/ppsrc<prid>.c	temporary file for prepass; <prid> = process ID

SEE ALSO

a.out(5), pattern(5L) ppdpat(1L)

DIAGNOSTICS

The diagnostics produced by **ppmkpat** are intended to be self explanatory. Ppmkpat exits with value PPSYNTAXERR if one or more errors were found in the definition.

BUGS

Ppmkpat will accept a long (over 60 characters in length) **patname**. However the object formats use the first seven characters after the last / character in **patname** as the C symbol name of the pattern (e.g. if **patname** = **/compat/sctab012**, then **sctab01** will be the symbol name of the pattern when it is part of an **a.out** file).