

NAME

lgrindef – LGrind’s language definition data base

NOTE

This man page is not yet much outdated, but might be soon except somebody asks me to work on it. Consider the LaTeX docs the real docs.

SYNOPSIS

texmf-dist/tex/latex/lgrind/lgrindef

DESCRIPTION

lgrindef contains all language definitions for **lgrind**(1). The data base is very similar to **vgrind**(5) and **termcap**(5), and it is upward-compatible with that of **vgrind**(5). Capabilities in **lgrindef** are of two types: Boolean capabilities which indicate that the language has some particular feature and string capabilities which give a regular expression or keyword list. Entries may continue onto multiple lines by giving a \ as the last character of a line. Lines starting with # are comments.

Capabilities

The following table names and describes each capability.

Name	Type	Description
ab	str	Regular expression for the start of an alternate form comment
ae	str	Regular expression for the end of an alternate form comment
bb	str	Regular expression for the start of a block
be	str	Regular expression for the end of a lexical block
cb	str	Regular expression for the start of a comment
ce	str	Regular expression for the end of a comment
cf	bool	(Boolean) Use specialized C function detection
id	str	String giving characters other than letters and digits that may legally occur in identifiers (default ‘_’)
kw	str	A list of keywords separated by spaces
lb	str	Regular expression for the start of a character constant
le	str	Regular expression for the end of a character constant
mb	str	Regular expression for the start of TeX math within a comment
me	str	Regular expression for the end of TeX math within a comment
np	str	Regular expression for a line not containing the start of a procedure
oc	bool	Present means upper and lower case are equivalent
pb	str	Regular expression for start of a procedure
pl	bool	Procedure definitions are constrained to the lexical level matched by the ‘px’ capability
px	str	A match for this regular expression indicates that procedure definitions may occur at the next lexical level. Useful for lisp-like languages in which procedure definitions occur as subexpressions of defuns.
rb	str	Regular expression for the start of a block outside the actual code
sb	str	Regular expression for the start of a string
se	str	Regular expression for the end of a string
rb	str	Regular expression for the end of a block outside a funtion (e. g. records in Pascal and Modula-2)
tb	str	Regular expression for the start of TeX text within a comment
tc	str	Use the named entry as a continuation of this one
te	str	Regular expression for the end of TeX text within a comment
tl	bool	Present means procedures are only defined at the top lexical level
vb	str	Regular expression for the start of typewriter text within a comment
ve	str	Regular expression for the end of typewriter text within a comment
zb	str	Regular expression for the start of program text within a comment
ze	str	Regular expression for the end of program text within a comment

Regular Expressions

lgrind uses regular expressions similar to those of **ex**(1) and **lex**(1). The characters '^', '\$', '|', ':', and '\ are reserved characters and must be 'quoted' with a preceding \ if they are to be included as normal characters. The metasympols and their meanings are:

\$	The end of a line
^	The beginning of a line
	A delimiter (space, tab, newline, start of line)
.....	Matches any string of symbols (like '.' in lex)
	Matches any identifier. In a procedure definition (the 'pb' capability) the string that matches this symbol is used as the procedure name.
()	Grouping
	Alternation
?	Last item is optional
\	Preceding any string means that the string will not match an input string if the input string is preceded by an escape character (\). This is typically used for languages (like C) that can include the string delimiter in a string by escaping it.

Unlike other regular expressions in the system, these match words and not characters. Hence something like '(tramp|steamer)flies?' would match 'tramp', 'steamer', 'trampflies', or 'steamerflies'. Contrary to some forms of regular expressions, **lgrind** alternation binds very tightly. Grouping parentheses are likely to be necessary in expressions involving alternation.

Keyword List

The keyword list is just a list of keywords in the language separated by spaces. If the 'oc' boolean is specified, indicating that upper and lower case are equivalent, then all the keywords should be specified in lower case.

EXAMPLE

The following entry, which describes the C language, is typical of a language entry.

```
C|the C programming language:\
:pb=~d?*?~d?~p~d??:bb={:be=}:cb=/*:ce=*/:\
:sb=":se=e":lb=:le=e':tl:\
:zb=@:ze=@:tb=% %:te=% %:mb=%$:me=$%:vb=%\|:ve=\| %:\
:kw=asm auto break case char continue default do double\
else enum extern float for fortran goto if int long\
register return short sizeof static struct switch typedef\
union unsigned while #define #else #endif #if #ifdef\
#ifndef #include #undef # define else endif if ifdef\
ifndef include undef:
```

Note that the first field is just the language name (and any variants of it). Thus the C language could be specified to **lgrind**(1) as 'c' or 'C', since case is not significant here.

FILES

texmf-dist/tex/latex/lgrind/lgrind file containing terminal descriptions

SEE ALSO

latex(1), **lgrind**(1), **vgrindefs**(5), For full documentation, refer to the package itself; it comes as a .dtx containing both the documentation and the LaTeX-files.