

New Conventions for LITTLE comments;  
Use of Semi-colon as statement terminator

The conventions for writing comments in LITTLE have been revised and extended, so that comments may now be expressed in three ways:

- a) By use of COMM ... CMND to delimit comment text, e.g.  
COMM THIS IS A COMMENT CMND
- b) By use of /\* ... \*/ to delimit comment text, e.g.  
/\* THIS IS A COMMENT \*/
- c) By use of a distinguished special character, nominally the dollar-sign (\$), which marks the rest of the current line as a comment, e.g.  
A = B \$ THIS IS A COMMENT

All comment forms are recognized at the lexical-token level, so that comments are "null" tokens. The following is a brief summary of the comment-recognizer, and should serve to answer questions about structure of valid comments:

- A. Start formation of new token; let char be current character.
- B. If char is a blank, set char = next\_character, and go to step A.
- C. If char is \$, set char=first\_character\_next\_line, and go to step A.
- D. If char is /, and next\_character is \*, then skip succeeding characters until first instance of \* followed by /; set char = char\_following\_slash, and go to step A.
- E. Otherwise, form token in usual way. If token obtained if COMM, of type name, then ignore the token, skip succeeding characters until 'CMND' is seen, delimited by characters which are neither letters nor numbers; set char = right\_delimiter, and go to step A.

Since comments are recognized when tokens are formed, and since all macro activity takes place above this level, macros can not be used to generate comments, comment delimiters, or parts of comments. Note also that all comment forms are processed by first noticing a comment "opener" and then skipping succeeding characters until a comment "ender" is seen, so that comments may contain statement terminators, quote marks, etc.

Use of semi-colon as statement terminator.

The semicolon ";" is now used to terminate LITTLE statements. To provide compatibility with the previous use of ".," to terminate statements, the lexical process will convert any instance of "." immediately followed by "," to an instance of ";". The check for an instance of ".," is made when forming tokens, so that instances of ".," within quoted strings or comments will not be converted to ";".

Preferred conventions; conversion to new conventions

The use of /\*...\*/ for comments is preferred to COMM...CMND; and ";" is preferred to ".,".

To convert LITTLE source text to the preferred form, use the editor macros available on the permanent file NEWLITTLELEXICALMACROS. Instructions for the use of these macros may be obtained from Dave Shields.

Note that the punch-file produced by the LITTLE compiler will punch ";" instead of ".,", but that comments may not currently be punched, and so are lost. The facility to punch comments may be available shortly to facilitate conversion to new conventions.