- `char *yytext' holds the text of the current token. It may be modified but not lengthened (you cannot append characters to the end).

- `int yyleng' holds the length of the current token.

- `FILE *yyin' is the file which by default `flex' reads from. It may be redefined but doing so only makes sense before scanning begins or after an EOF has been encountered. Changing it in the midst of scanning will have unexpected results since `flex' buffers its input; use `yyrestart()' instead. Once scanning terminates because an end-of-file has been seen, you can assign `yyin' at the new input file and then call the scanner again to continue scanning.

- `void yyrestart( FILE *new_file )' may be called to point `yyin' at the new input file. The switch-over to the new file is immediate (any previously buffered-up input is lost). Note that calling `yyrestart()' with `yyin' as an argument thus throws away the current input buffer and continues scanning the same input file.

- `FILE *yyout' is the file to which `ECHO' actions are done. It can be reassigned by the user.

When the scanner receives an end-of-file indication from YY_INPUT, it then checks the `yywrap()' function. If `yywrap()' returns false (zero), then it is assumed that the function has gone ahead and set up `*yyin*' to point to another input file, and scanning continues. If it returns true (non-zero), then the scanner terminates, returning 0 to its caller. Note that in either case, the start condition remains unchanged; it does *not* revert to `*INITIAL*'.

If you do not supply your own version of `yywrap()', then you must either use `%option noyywrap' (in which case the scanner behaves as though `yywrap()' returned 1), or you must link with `*-lfl*' to obtain the default version of the routine, which always returns 1.