IEEE Standards Interpretations for IEEE Std 1003.2™-1992 IEEE Standard for Information Technology--Portable Operating System Interfaces (POSIX®)--Part 2: Shell and Utilities

Interpretation Request #93
Topic: yacc **Relevant Clauses:** A.3.6.1.3.1, Line 690-691

The applicable condition is when yacc input file has the programs section. Please see Appendix 1 attached to this note. A suggested correction, if applicable: Remove Line 690-691 Reason for the suggested correction There are quite many yacc input files assuming that the program section is copied before yyparse(). If the program section needs to be copied after yyparse(), then many of already written yacc input files need to be modified. The line 936-941 says that "It is unspecified whether the program section precedes or follows the semantic actions in the output file; therefore, if the application contains any macro definitions and declarations intended to apply the code in the semantic actions, it shall place them within %{ ... %} in the declaration section." The above says that "it is unspecified whether the program section precedes or follows the semantic actions in the output file;". Most of yacc implementation places its semantic actions in yyparse() and the document also mentions it. (Line 685-686) Therefore, there should be no problem removing line 690- 691.

Appendix 1: An example: This yacc input file is a little example about the problem. From this input, yacc will generate a parser. The parser calls hello_world() when non-terminal sound is recognized. The parser also calls print_str() when ding is recognized. %{ * * Declaration *\ #include <stdio.h char *hello_world(char *); %} %token DING %token DONG %token DELL %% sound: ding DONG DELL { hello_world("sound recognized\n"); }; ding: DING { print_str("ding recognized\n"); } %% char * print_str(char *s) { printf(s); return (s); } YaccØs output will have the following format. (This is y.tab.c by default.) 1) Declaration section The beginning of y.tab.c is a copy of the input portion surrounded by %{ and %}.

In the above example, #include <stdio.h char *hello_world(char *) are copied. 2) Then bunch of #defines and variables which will be used by yacc generated code comes. *3) Then user defined code will be copied. In this above example, char * print_str(char *s) { printf(s); return (s); } 4) Then tables generated by yacc comes in. 5) Then finally the parser generated by yacc comes. This is yyparse() and in the example above, the generated parser will call print_str() and hellow_world(). The spec says *3) part should be copied after 5). If *3) part is copied after yyparse(), then the C compiler later will complain that print_str() is implicitly declared as int type, but later it is redeclared as char *. So the spec says, that the functions that the parser uses should be declared at the declaration section. %{ * * Declaration * #include <stdio.h char *hello_world(char *); char *print_str(char *); %} Quite many existing yacc input are written like ding.y expecting that print_str() does not need to be declared at the declaration section. (Because user codes are copied before yyparse().)

Interpretation Response
First note that in B) the relevant subclause is A.3.6.3.1, not A.3.6.1.3.1 The standard states behavior for the yacc code file, and conforming implementations must conform to this. However, concerns have been raised about this which are being referred to the sponsor.

Rationale for Interpretation
None.