

**NAME**

Yacc – an LALR(1) parser generator

**SYNOPSIS**

**yacc** [ **-dlrtv** ] [ **-b** *file\_prefix* ] [ **-p** *symbol\_prefix* ] *filename*

**DESCRIPTION**

*Yacc* reads the grammar specification in the file *filename* and generates an LR(1) parser for it. The parsers consist of a set of LALR(1) parsing tables and a driver routine written in the C programming language. *Yacc* normally writes the parse tables and the driver routine to the file *y.tab.c*.

The following options are available:

- b** *file\_prefix*  
The **-b** option changes the prefix prepended to the output file names to the string denoted by *file\_prefix*. The default prefix is the character *y*.
- d**  
The **-d** option causes the header file *y.tab.h* to be written.
- l**  
If the **-l** option is not specified, *yacc* will insert The generated code to the user's original code. If the **-l** option is specified, *yacc* will not insert the *.TP* **-p** *symbol\_prefix*  
The **-p** option changes the prefix prepended to yacc-generated symbols to the string denoted by *symbol\_prefix*. The default prefix is the string *yy*.
- r**  
The **-r** option causes *yacc* to produce separate files for code and tables. The code file is named *y.code.c*, and the tables file is named *y.tab.c*.
- t**  
The **-t** option changes the preprocessor directives generated by *yacc* so that debugging statements will be incorporated in the compiled code.
- v**  
The **-v** option causes a human-readable description of the generated parser to be written to the file *y.output*.

If the environment variable TMPDIR is set, the string denoted by TMPDIR will be used as the name of the directory where the temporary files are created.

**FILES**

*y.code.c*  
*y.tab.c*  
*y.tab.h*  
*y.output*  
*/tmp/yacc.aXXXXXX*  
*/tmp/yacc.tXXXXXX*  
*/tmp/yacc.uXXXXXX*

**DIAGNOSTICS**

If there are rules that are never reduced, the number of such rules is reported on standard error. If there are any LALR(1) conflicts, the number of conflicts is reported on standard error.

**NAME**

yacc – yet another compiler-compiler

**SYNOPSIS**

yacc [ **-vd** ] grammar

**DESCRIPTION**

*Yacc* converts a context-free grammar into a set of tables for a simple automaton which executes an LR(1) parsing algorithm. The grammar may be ambiguous; specified precedence rules are used to break ambiguities.

The output file, *y.tab.c*, must be compiled by the C compiler to produce a program *yyparse*. This program must be loaded with the lexical analyzer program, *yylex*, as well as *main* and *yyerror*, an error handling routine. These routines must be supplied by the user; *Lex(1)* is useful for creating lexical analyzers usable by *yacc*.

If the **-v** flag is given, the file *y.output* is prepared, which contains a description of the parsing tables and a report on conflicts generated by ambiguities in the grammar.

If the **-d** flag is used, the file *y.tab.h* is generated with the *define* statements that associate the *yacc*-assigned ‘token codes’ with the user-declared ‘token names’. This allows source files other than *y.tab.c* to access the token codes.

**FILES**

y.output	
y.tab.c	
y.tab.h	defines for token names
yacc.tmp, yacc.acts	temporary files
/usr/lib/yaccpar	parser prototype for C programs
/lib/liby.a	library with default ‘main’ and ‘yyerror’

**SEE ALSO**

*lex(1)*

*LR Parsing* by A. V. Aho and S. C. Johnson, Computing Surveys, June, 1974.

*YACC – Yet Another Compiler Compiler* by S. C. Johnson.

**DIAGNOSTICS**

The number of reduce-reduce and shift-reduce conflicts is reported on the standard output; a more detailed report is found in the *y.output* file. Similarly, if some rules are not reachable from the start symbol, this is also reported.

**BUGS**

Because file names are fixed, at most one *yacc* process can be active in a given directory at a time.