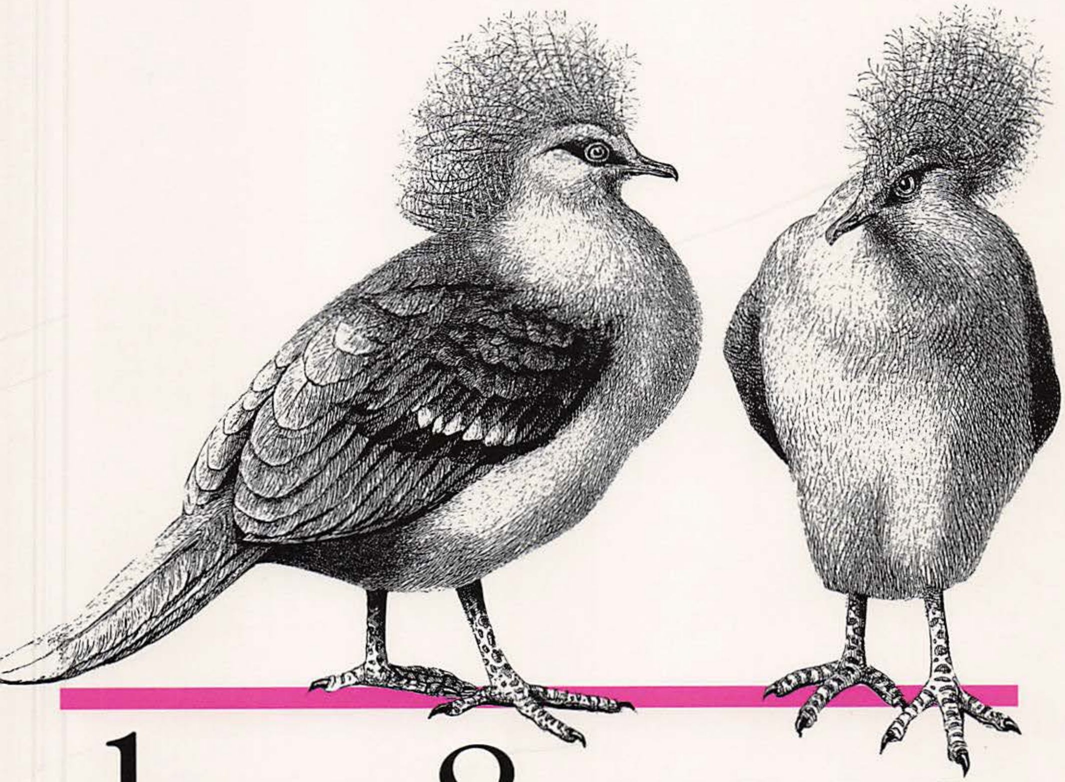
*UNIX Programming Tools*



# lex & yacc

John R. Levine

***lex & yacc***
by John R. Levine, Tony Mason and Doug Brown

***Editor:*** Dale Dougherty

***Printing History:***

| | |
|---|---|
| May 1990: | First Edition. |
| January 1991: | Minor corrections. |
| October 1992: | Second Edition. Major revisions. |
| February 1995: | Minor corrections. |

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks and The Java Series is a trademark of O'Reilly & Associates, Inc. Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly and Associates, Inc. was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher assumes no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

ISBN: 1-56592-000-7 [1/00]

## *Availability of Lex and Yacc*

Lex and yacc were both developed at Bell Laboratories in the 1970s. Yacc was the first of the two, developed by Stephen C. Johnson. Lex was designed by Mike Lesk and Eric Schmidt to work with yacc. Both lex and yacc have been standard UNIX utilities since 7th Edition UNIX. System V and older versions of BSD use the original AT&T versions, while the newest version of BSD uses flex (see below) and Berkeley yacc. The articles written by the developers remain the primary source of information on lex and yacc.

The GNU Project of the Free Software Foundation distributes *bison*, a yacc replacement; bison was written by Robert Corbett and Richard Stallman. The bison manual, written by Charles Donnelly and Richard Stallman, is excellent, especially for referencing specific features. Appendix D discusses bison.

BSD and GNU Project also distribute *flex* (*F*ast *Lex*ical Analyzer Generator), "a rewrite of lex intended to right some of that tool's deficiencies," according to its reference page. Flex was originally written by Jef Poskanzer; Vern Paxson and Van Jacobson have considerably improved it and Vern currently maintains it. Appendix E covers topics specific to flex.

There are at least two versions of lex and yacc available for MS-DOS and OS/2 machines. MKS (Mortice Kern Systems Inc.), publishers of the MKS Toolkit, offers lex and yacc as a separate product that supports many PC C compilers. MKS lex and yacc comes with a very good manual. Appendix F covers MKS lex and yacc. Abraxas Software publishes PCYACC, a version of lex and yacc which comes with sample parsers for a dozen widely used programming languages. Appendix G covers Abraxas' version lex and yacc.

## *Sample Programs*

The programs in this book are available free from UUNET (that is, free except for UUNET's usual connect-time charges). If you have access to UUNET, you can retrieve the source code using UUCP or FTP. For UUCP, find a machine with direct access to UUNET, and type the following command:

```
uucp uunet\!~/nutshell/lexyacc/progs.tar.Z yourhost\!~/yourname/
```

The backslashes can be omitted if you use the Bourne shell (*sh*) instead of the C shell (*csh*). The file should appear some time later (up to a day or more) in the directory */usr/spool/uucppublic/*yourname. If you don't have an account but would like one so that you can get electronic mail, then contact UUNET at 703-204-8000.

To use *ftp*, find a machine with direct access to the Internet. Here is a sample session, with commands in boldface.

```
% ftp ftp.oreilly.com
Connected to ftp.oreilly.com.
220 FTP server (Version 5.99 Wed May 23 14:40:19 EDT 1990) ready.
Name (ftp.oreilly.com:yourname): anonymous
331 Guest login ok, send ident as password.
Password: ambar@ora.com (use your user name and host here)
230 Guest login ok, access restrictions apply.
ftp> cd published/oreilly/nutshell/lexyacc
250 CWD command successful.
ftp> binary (you must specify binary transfer for compressed files)
200 Type set to I.
ftp> get progs.tar.Z
200 PORT command successful.
150 Opening BINARY mode data connection for progs.tar.Z.
226 Transfer complete.
ftp> quit
221 Goodbye.
%
```

The file is a compressed tar archive. To extract files once you have retrieved the archive, type:

```
% zcat progs.tar.Z | tar xf -
```

System V systems require the following tar command instead:

```
% zcat progs.tar.Z | tar xof -
```

## *Conventions Used in This Handbook*

The following conventions are used in this book:

**Bold** is used for statements and functions, identifiers, and program names.

*Italic* is used for file, directory, and command names when they appear in the body of a paragraph as well as for data types and to emphasize new terms and concepts when they are introduced.

`Constant Width` is used in examples to show the contents of files or the output from commands.

**`Constant Bold`** is used in examples to show command lines and options that you type literally.

Quotes are used to identify a code fragment in explanatory text. System messages, signs, and symbols are quoted as well.

% is the Shell prompt.

[] surround optional elements in a description of program syntax. (Don't type the brackets themselves.)

## *Acknowledgments*

This first edition of this book began with Tony Mason's MGL and SGL compilers. Tony developed most of the material in this book, working with Dale Dougherty to make it a "Nutshell." Doug Brown contributed Chapter 8, *Yacc Ambiguities and Conflicts.* Dale also edited and revised portions of the book. Tim O'Reilly made it a better book by withholding his editorial blessing until he found what he was looking for in the book. Thanks to Butch Anton, Ed Engler, and Mike Loukides for their comments on technical content. Thanks also to John W. Lockhart for reading a draft with an eye for stylistic issues. And thanks to Chris Reilley for his work on the graphics. Finally, Ruth Terry brought the book into print with her usual diligence and her sharp eye for every editorial detail. Though she was trying to work odd hours to also care for her family, it seemed she was caring for this book all hours of the day.

For the second edition, Tony rewrote chapters 1 and 2, and Doug updated Chapter 8. John Levine wrote Chapters 3, 5, 6, 7, and most of the appendices, and edited the rest of the text. Thanks to the technical reviewers, Bill Burke, Warren Carithers, Jon Mauney, Gary Merrill, Eugene Miya, Andy Oram, Bill Torcaso, and particularly Vern Paxson whose detailed page-by-page suggestions made the fine points much clearer. Margaret Levine Young's blue pencil (which was actually pink) tightened up the text and gave the book editorial consistency. She also compiled most of the index. Chris Reilly again did the graphics, and Donna Woonteiler did the final editing and shepherded the book through the production process.

# Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

## Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

## Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

## Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

## API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

**LAW FIRMS**

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

**FINANCIAL INSTITUTIONS**

Litigation and bankruptcy checks for companies and debtors.

**E-DISCOVERY AND LEGAL VENDORS**

Sync your system to PACER to automate legal marketing.

**WHAT WILL YOU BUILD?** | **sales@docketalarm.com** | **1-866-77-FASTCASE**