3rd Edition

Updated & Expanded

UNIX POWER TOOLS

Shelley Powers, Jerry Peek, Tim O'Reilly & Mike Loukides

O'REILLY™

UNIX Power Tools, 3rd Edition

Jerry Peek
Shelley Powers
Tim O'Reilly
Mike Loukides

Editor
Laurie Petrycki

O'Reilly Media

SPECIAL OFFER: Upgrade this ebook with O'Reilly

A Note Regarding Supplemental Files

Supplemental files and examples for this book can be found at
http://examples.oreilly.com/9780596003302/. Please use a standard desktop web browser to access these files, as they may not be accessible from all ereader devices.

All code files or examples referenced in the book will be available online. For physical books that sh with an accompanying disc, whenever possible, we've posted all CD/DVD content. Note that while w provide as much of the media content as we are able via free download, we are sometimes limited by licensing restrictions. Please direct any questions or concerns to booktech@oreilly.com.

How to Use This Book

This section refers to conventions used in the print book and explains how they were modified for the Safari version. The numbers in the following images correspond to the list below.
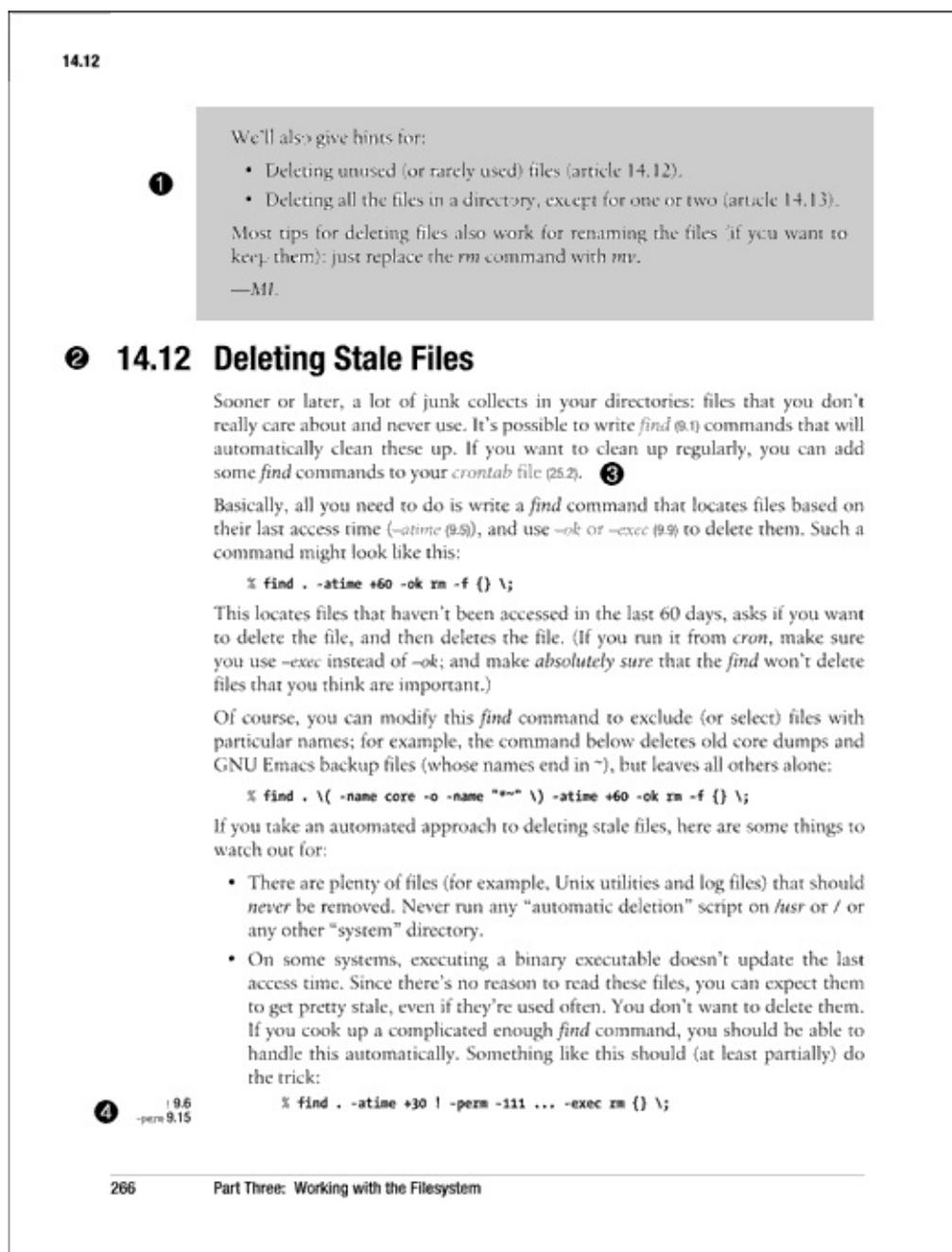
1. **Summary Boxes.** You'll see gray shaded summary boxes all through the book. (On Safari, the Summary Boxes are bordered sidebars with the title "Summary Box.") They summarize a topic and point you to articles with examples and further explanation.

2. **Article/Section Number.** The first two digits indicate in which chapter the article resides; the last two digits indicate the number of the article within that chapter. The article number is used to refer to this article in all cross-references throughout the book. (On Safari, Article numbers correspond to Section numbers.)

3. **Cross-Reference in a Sentence.** To find out more about the topic displayed in gray type (On Safari, this text is displayed in boldface.), see the article referenced by the number in parentheses immediately following the term.

4. **Cross-Reference in a Code Example.** When a cross-reference occurs in an example, the cross-referenced text and related article number appear in the left margin. (On Safari, these cross-references appear above the code example.)

join can do a lot more than this simple example shows. See your online manual page. The GNU version of *join* is on the CD-ROM.

join

—*JP*

21.20 What is (or isn't) Unique?

❺

uniq

uniq reads a file and compares adjacent lines (which means you'll usually want to sort the file first—to be sure identical lines appear next to each other). Here's what *uniq* can do as it watches the input lines stream by:

- With the *–u* option, the output gets only the lines that occur just once (and weren't repeated).
- The *–d* option does the opposite: the output gets a *single* copy of each line that was repeated (no matter how many times it was repeated).

  (The GNU version also has a *–D* option. It's like *–d* except that *all* duplicate lines are output.)

- The default output (with no options) is the union of *–u* and *–d*: only the first occurrence of a line is written to the output file; any adjacent copies of a line (second, third, etc.) are ignored.

- The output with *–c* is like the default, but each line is preceded by a count of how many times it occurred.

❻

Be warned:

```
% uniq file1 file2
```

will *not* print the unique lines from both *file1* and *file2* to standard output. It will *replace* the contents of *file2* with the unique lines from *file1*!

Three more options control how comparisons are done:

- *-n* ignores the first *n* fields of a line and all whitespace before each. A field is defined as a string of non-whitespace characters (separated from its neighbors by whitespace).
- *+n* ignores the first *n* characters. Fields are skipped before characters.
- *-w n* in the GNU version compares no more than *n* characters in each line.
- GNU *uniq* also has *–i* to make comparisons case-insensitive. (Upper and lowercase letters compare equal.)

uniq is often used as a filter. See also *comm* (11.8), *sort* (22.1), and especially *sort –u* (22.6).

So what can you do with all of this?

1. **Globe** If you don't want to type this script into a file yourself, or if we're talking about a C program that isn't shown, you can download it from the book's web site. See the Preface for full details on the content available for download. (Online version available at http://examples.oreilly.com/upt3)

2. **Screw.** Be careful with this feature, or you might get screwed.

Real World-watching generation fail to realize is that
our American forefathers, under the tutelage of Zog,
the wizened master sage from Zeta-Reticuli, had to fight
not only the godless and effete British for our system of
self-determined government, but also avoid the terrors of
hynpo-death from the dark and unclean Draco-Repitilians.

There is one subtlety to *fmt* to be aware of: *fmt* expects sentences to end with either a period, a question mark or an exclaimation point followed by two spaces. If your document isn't marked up according to this convention, *fmt* can't differentiated between sentences and abbreviations. This is a common "gotcha" that appears frequently on Usenet.

❼ On at least one version of Unix, *fmt* is a disk initializer (disk formatter) command. Don't run *that* command accidentally! Check your online manual page and see the *fmt* equivalents below.

There are a few different versions of *fmt*, some fancier than others. In general, the program assumes that:

- Paragraphs have blank lines between them.
- If a line is indented, keep the indentation.
- The output lines should be about 70 characters wide. Some have a command line option to let you set this. For example, fmt -132 (or, on some versions, fmt -l 132) would reformat your file to have lines with no more than 132 characters on each.
- It reads files or standard input. Lines will be written to standard output.

❽ *fmt* The GNU *fmt* is on the CD-ROM. There are also a couple of freely available versions. Many versions of *fmt* have options for other structured data. The –*p* option (21.4) reformats program source code. (If your *fmt* doesn't have –*p*, the *recomment* (21.4) script uses standard *fmt* with *sed* to do the same thing.) The –*s* option breaks long lines at whitespace but doesn't join short lines to form longer ones.

Alternatively, you can make your own (21.3) simple (and a little slower) version with *sed* and *nroff*. If you want to get fancy (and use some *nroff* and/or *tbl* coding), this will let you do automatically formatted text tables, bulleted lists, and much more.

❾ —JP, TOR, and JJ

21.3 Alternatives to fmt

fmt (21.2) is hard to do without once you've learned about it. Unfortunately, it's not available in some versions of Unix. You can get the GNU version from the

1. **Pushpin.** A note to keep in mind, or a helpful tip.
2. **Bomb.** A bomb icon in the margin is a cross-reference to another article that explains the possible trouble you might encounter using the tip or script in the current article. (You can think of the bomb as a cross-referenced screw.) (On Safari, the Bomb appears above the paragraph it refers to.
3. **Author's Initials.** The author's full name is listed in the Preface.

Preface

A Book for Browsing

Technical books can be boring. But this is not an ordinary technical book! This book is like an almanac, a news magazine, and a hypertext database all rolled into one. Instead of trying to put the topics in perfect order — and expecting you to start at the beginning, then read through to the end — we hope that you'll browse. Start anywhere. Read what you want. (That's not quite true. First, you should read this Preface and the pages before it titled *How to Use This Book*. They will help you get the most out of your time with this book. Next, you may want to skim through the Unix fundamentals in [Chapter 1](). *Then* read what you want.)

Like an Almanac

The book is full of practical information. The main purpose isn't to teach you concepts (though they're in here). We've picked a lot of common problems, and we'll show you how to solve them.

Even though it's not designed to be read in strict order, the book is organized into chapters with related subject matter. If you want to find a specific subject, the table of contents is still a good place to start. In addition, several of the chapters contain **shaded boxes**. These are like small tables of contents on a particular subject, which might be even more limited than the scope of the chapter itself. Use the **Index** when you're trying to find a specific piece of information instead of a general group of articles about a topic.

Like a News Magazine

This book has **short articles**. Most show a problem and a solution — in one page or less. The articles are numbered within each chapter. Not all articles are "how-to" tips. Some articles have background information and concepts.

Like a Hypertext Database

Each article doesn't define all the concepts and words used. Instead, it gives you "links" that let you get more information *if you need it*. It's easy to get more information when you need it, but just skip the link if you don't. *Unix Power Tools* uses two kinds of links: those in a sentence and those in the margin. For examples, see the pages before this Preface titled *How to Use This Book*.

Programs on the Web

The book describes scripts and freely available programs that are available on the web site. An article about a program or file that's on the web site will have a globe icon next to it, like this. To get one of these programs, visit the web site:

http://www.oreilly.com/catalog/upt3/

About Unix Versions

There are lots of similarities between different versions of Unix. But it's almost impossible to write a book that covers every detail of every version correctly. Where we know there might be big differences or problems, we'll print a note in the text. Other places, we're forced to use "weasel words" like "Some versions of XXX will do...," without telling you exactly *which* versions. When you see those weasel words, what can you do?

- If the command or feature won't destroy anything when it doesn't work, try it! For instance, don't experiment with *rm*, the command that removes files. But *cat*, a command that shows files, probably won't hurt anything if some feature doesn't work with your version.

- Look at the *online* manual or check your *vendor's* latest printed manuals. However, even these can be wrong. For instance, your system administrator may have installed a local version of a command that works differently — but not updated the online documentation. Be careful with "generic" manuals, the kind you buy at a bookstore; there are lots of versions of Unix, and the manual may not match your version closely enough.

- Ask your system administrator or another "guru" for help before you use a command that might be dangerous.

Cross-References

If a cross-reference is to a single word — for example, a command name like this: *tar* — the cross reference is probably to an article that introduces that command. Cross references to phrases — like this: **from a parent process to child process** — are to an article that explains more about the concept or problem printed in gray.

Cross references don't necessarily give a complete list of all articles about a topic. We've tried to pick one or a few articles that give the best information. For a more complete list, use the Index.

What's New in the Third Edition

There have been some big changes in Unix since we wrote the first edition in the early 1990s, and there's been a surprising number of changes since the second edition, released in the late 1990s. Well over half of the articles have been revised, and we've expanded our coverage of the so-called small Unix flavors: Linux, FreeBSD, Mac OS X's Darwin, and so on.

A major change to this edition was the addition of several new topics relevant to today's connected world, including protecting your machine from attack and several articles related to Internet protocol. We've also added chapters with coverage of two of the more popular languages used in Unix: Perl and Python.

Typefaces and Other Conventions

Italic

Is used for the names of all Unix utilities, switches, directories, and filenames and to emphasize new terms and concepts when they are first introduced. It's also used in programs and examples to explain what's happening or what's been left out at the . . . marks.

Bold

Is used occasionally within text to make words easy to find — just like movie stars' names in the People section of your local newspaper.

`Constant width`

Is used for sample code fragments and examples. A reference in text to a word or item used in an example or code fragment is also shown in constant width font.

`Constant width bold`

Is used in examples to show commands or text that would be typed in literally by the user.

`Constant width italic`, `bold italic`

Are used in code fragments and examples to show variables for which a context-specific substitution should be made. (The variable `filename`, for example, would be replaced by some actual filename.)

function(n)

Is a reference to a manual page in Section *n* of the Unix programmer's manual. For example, *getopt(3)* refers to a page called *getopt* in Section 3.

%

Is the C-shell prompt.

$

Is the Bourne-shell prompt.

:-)

Is a "smiley face" that means "don't take this seriously." The idea started on *Usenet* and spread.

& . . .

Stands for text (usually computer output) that's been omitted for clarity or to save space.

CTRL

Starts a control character. To create CTRL-d, for example, hold down the "control" key and press the "d" key. Control characters are not case sensitive; "d" refers to both the upper- and lowercase letter. The notation ^D also means CTRL-d. Also, you'll sometimes see the key sequence in bold (for example, CTRL-d is used when we want to make it clear exactly what you should type.

·

Is used in some examples to represent a space chara·cter.

`TAB`

Is used in some examples to represent a TAB character.

The Authors

This book is the effort of several authors who have contributed to one edition or another since the first edition was released. Much of the material for the first and second edition came from three authors: Jerry Peek, Tim O'Reilly, and Mike Loukides. Their work is still present, though edited for current times. This third edition brought in four new authors, who edited the previous material, in addition to contributing new articles: Shelley Powers, Steven Champeon, Deborah Hooker, and Joe Johnston.

In addition, we also had several other authors contribute to all three editions — either people who originally posted a good tip to Usenet, authors of Nutshell Handbooks who let us take material from their books, or authors of software packages who let us take a few paragraphs from README files or other documentation.

Here's a list of authors and their initials:

AD	Angus Duggan	JIK	Jonathan I. Kamens
AF	AEleen Frisch	JM	Jeff Moskow
AN	Adrian Nye	JP	Jerry Peek
BA	Brandon S. Allbery	JJ	Joe Johnston
BB	Bruce Barnett	JS	John Strang
BR	Bill Rosenblatt	LK	Lar Kaufman
CT	Chris Torek	LL	Linda Lamb
DC	Debra Cameron	LM	Linda Mui
DD	Dale Dougherty	LW	Larry Wall
DG	Daniel Gilly	MAL	Maarten Litmaath
DH	Dave Hitz	ML	Mike Loukides
DJPH	Deborah Hooker	MS	Mike Stansbery
DL	Don Libes	RS	Randal Schwartz

DR	Daniel Romike	SP	Shelley Powers
DS	Daniel Smith	SG	Simson Garfinkel
EK	Eileen Kramer	SC	Steve Champeon
EP	Eric Pearce	SW	Sun Wu
GS	Gene Spafford	TC	Tom Christiansen
GU	Greg Ubben	TOR	Tim O'Reilly
HS	Henry Spencer	UM	Udi Manber

The Fine Print

Where we show an article from an author on Usenet, that person may not have thought of the idea originally, but may just be passing on something he or she learned. We attribute everything we can.

Request for Comments

Please tell us about any errors you find in this book or ways you think it could be improved. Our U.S.
mail address, phone numbers, and electronic mail address are as follows:

O'Reilly & Associates, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
(800) 998-9938 (in the United States or Canada)
(707) 829-0515 (international/local)
(707) 829-0104 (fax)
bookquestions@oreilly.com (email)

Acknowledgments for the First Edition

This book wouldn't exist without Ron Petrusha. As the technical book buyer at Golden-Lee, a major book distributor, he discovered us soon after we started publishing Nutshell Handbooks in the mid-1980s. He was one of our early boosters, and we owed him one. So when he became an editor at Bantam (whose computer-book publishing operations were later acquired by Random House), we too[k] him seriously when he started asking if there was anything we could do together.

At first nothing seemed to fit, since by that time we were doing pretty well as a publisher. We neede[d] to find something that we could do together that might sell better than something that either compan[y] might do alone. Eventually, Ron suggested that we copublish a Unix book for Bantam's "Power Tool[s" series. This made sense for both of us. It gave Bantam access to our Unix expertise and reputation, an[d] it gave us a chance to learn from Bantam about the mass market bookstore trade, as well as build on their successful "Power Tools" series.

But what would the book contain? There were two features of Bantam's original *DOS Power Tools* th[at] we decided to emulate: its in-depth treatment of under-documented system features and its large collection of freely available scripts and utilities. However, we didn't want to write yet another book that duplicated the format of many others on the market, in which chapters on each of the major Uni[x] tools follow one another in predictable succession. Our goal was certainly to provide essential technical information on Unix utilities, but more importantly, to show how the utilities can be combined and used to solve common (and uncommon) problems.

Similarly, because we were weary of the multitude of endless tutorial books about Unix utilities, we wanted to keep the tone brisk and to the point. The solution I came up with, a kind of "hypertext in print," actually owes a lot to Dale Dougherty. Dale has been working for several years on hypertext and online information delivery, and I was trying to get him to work with me on this project. So I trie[d] to imagine the kind of book that he might like to create. (We have a kind of friendly rivalry, in which we try to leapfrog each other with ideas for new and better books!) Dale's involvement never went fa[r] beyond the early brainstorming stage, but the book still bears his indirect stamp. In some of the first books he wrote for me, he introduced the idea that sidebars — asides that illuminate and expand on the topic under discussion — could be used effectively in a technical book. Well, Dale, here's a book that's nothing but sidebars!

Dale, Mike Loukides, and I worked out the basic outline for the book in a week or two of brainstorming and mail exchanges. We thought we could throw it together pretty quickly by mining many of our existing books for the tips and tricks buried in them. Unfortunately, none of us was ever able to find enough time, and the book looked to be dying a slow death. (Mike was the only one who got any writing done.) Steve Talbott rescued the project by insisting that it was just too good an idea to let go; he recruited Jerry Peek, who had just joined the company as a writer and Unix consultant/tools developer for our production department.

Production lost the resulting tug of war, and Jerry plunged in. Jerry has forgotten more Unix tips and tricks than Mike, Dale, or I ever knew; he fleshed out our outline and spent a solid year writing and collecting the bulk of the book. I sat back in amazement and delight as Jerry made my ideas take shape. Finally, though, Jerry had had enough. The book was just too big, and he'd never signed on to do it all alone! (It was about 1,000 pages at that point, and only half done.) Jerry, Mike, and I spent a week locked up in our conference room, refining the outline, writing and cutting articles, and generally trying to make Jerry feel a little less like Sisyphus.

From that point on, Jerry continued to carry the ball, but not quite alone, with Mike and I playing "tag team," writing and editing to fill in gaps. I'm especially grateful to Mike for pitching in, since he had many other books to edit and this was supposed to be "my" project. I am continually amazed by the breadth of Mike's knowledge and his knack for putting important concepts in perspective.

Toward the end of the project, Linda Mui finished up another book she was working on and joined the project, documenting many of the freely available utilities that we'd planned to include but hadn't gotten around to writing up. Linda, you really saved us at the end!

Thanks also to all the other authors, who allowed us to use (and sometimes abuse!) their material. In particular, we're grateful to Bruce Barnett, who let us use so much of what he's written, even though we haven't yet published his book, and Chris Torek, who let us use many of the gems he's posted to the Net over the years. (Chris didn't keep copies of most of these articles; they were saved and sent in by Usenet readers, including Dan Duval, Kurt J. Lidl, and Jarkko Hietaniemi.)

Jonathan Kamens and Tom Christiansen not only contributed articles but read parts of the book with learned and critical eyes. They saved us from many a "power goof." If we'd been able to give them enough time to read the whole thing, we wouldn't have to issue the standard disclaimer that any errors that remain are our own. H. Milton Peek provided technical review and proofreading. Four sharp-eyed Usenet readers helped with debugging: Casper Dik of the University of Amsterdam, Byron Ratzikis of Network Appliance Corporation, Dave Barr of the Population Research Institute, and Duncan Sinclair.

In addition to all the acknowledged contributors, there are many unacknowledged ones — people who have posted questions or answers to the Net over the years and who have helped to build the rich texture of the Unix culture that we've tried to reflect in this book. Jerry also singles out one major contributor to his own mastery of Unix. He says: "Daniel Romike of Tektronix, Inc. (who wrote Section 28.5 and Section 30.8 in the early 1980s, by the way) led the first Unix workshop I attended. He took the time to answer a ton of questions as I taught myself Unix in the early 1980s. I'm sure some of the insights and neat tricks that I thought I've figured out myself actually came from Dan instead."

James Revell and Bryan Buus scoured "the Net" for useful and interesting free software that we weren't aware of. Bryan also compiled most of the software he collected so we could try it out and gradually winnow down the list.

Thanks also to all of the authors of the software packages we wrote about! Without their efforts, we wouldn't have had anything to write about; without their generosity in making their software free in the first place, we wouldn't be able to distribute hundreds of megabytes of software for the price of a book.

Jeff Moskow of Ready-to-Run Software solved the problem we had been putting off to the end: that of packaging up all the software for the original disk, porting it to the major Unix platforms, and making it easy to install. This was a much bigger job than we'd anticipated, and we could never have done it without Jeff and the RTR staff. We might have been able to distribute source code and binaries for a few platforms, but without their porting expertise, we could never have ported all these programs to every supported platform. Eric Pearce worked with RTR to pre-master the software for CD-ROM duplication, wrote the installation instructions, and made sure that everything came together at the end! (Eric, thanks for pitching in at the last minute. You were right that there were a lot of details that might fall through the cracks.)

Edie Freedman worked with us to design the format of the book — quite an achievement considering everything we wanted the format to do! She met the challenge of presenting thousands of inline cross-

references without distracting the reader or creating a visual monstrosity. What she created is as attractive as it is useful — a real breakthrough in technical book design, and one that we plan to use again and again!

Lenny Muellner was given the frightful task of implementing all of our ideas in *troff* — no mean feat and one that added to his store of grey hair.

Eileen Kramer was the copyeditor, proofreader, and critic who made sure that everything came together. For a thousand-plus page book with multiple authors, it's hard to imagine just how much work that was.

Ellie Cutler wrote the index; Chris Reilley created the illustrations. Additional administrative support was provided by Bonnie Hyland, Donna Woonteiler, and Jane Appleyard.

—Tim O'Reilly

sample content of Unix Power Tools, Third Edition

- download online Whip Hand (Sid Halley, Book 2) online
- *download online Glassperlespillet pdf, azw (kindle)*
- **Unfair Advantage: The Power of Financial Education book**
- Agile Software Architecture: Aligning Agile Processes and Software Architectures pdf, azw (kindle), epub, doc, mobi
- **Man About Town pdf**

- http://tuscalaural.com/library/Whip-Hand--Sid-Halley--Book-2-.pdf
- http://honareavalmusic.com/?books/A-Change-in-Altitude--A-Novel.pdf
- http://www.freightunlocked.co.uk/lib/Surrealist-Painters-and-Poets--An-Anthology.pdf
- http://deltaphenomics.nl/?library/Sweet-Poison--Why-Sugar-Makes-Us-Fat.pdf
- http://fitnessfatale.com/freebooks/Man-About-Town.pdf