
Effective Tcl Tk Programming Writing Better Programs In Tcl And Tk

Recognizing the pretension ways to get this ebook **Effective Tcl Tk Programming Writing Better Programs In Tcl And Tk** is additionally useful. You have remained in right site to start getting this info. get the Effective Tcl Tk Programming Writing Better Programs In Tcl And Tk colleague that we allow here and check out the link.

You could purchase lead Effective Tcl Tk Programming Writing Better Programs In Tcl And Tk or acquire it as soon as feasible. You could quickly download this Effective Tcl Tk Programming Writing Better Programs In Tcl And Tk after getting deal. So, when you require the ebook swiftly, you can straight acquire it. Its so categorically simple and hence fats, isnt it? You have to favor to in this tone

Effective Tcl
Tk
Programming
Writing
Better
Programs In
Tcl And Tk

Downloaded
from
ssm.nvherald.com
by guest

HESS JUSTICE

A Developer's Guide Elsevier
Introduction to
Network
Simulator NS2
is a primer
providing
materials for
NS2
beginners,
whether
students,
professors, or
researchers
for
understanding
the
architecture of
Network
Simulator 2
(NS2) and for
incorporating
simulation
modules into
NS2. The
authors

discuss the
simulation
architecture
and the key
components
of NS2
including
simulation-
related
objects,
network
objects,
packet-related
objects, and
helper objects.
The NS2
modules
included
within are
nodes, links,
SimpleLink
objects,
packets,
agents, and
applications.
Further, the
book covers
three helper
modules:
timers,
random
number

generators,
and error
models. Also
included are
chapters on
summary of
debugging,
variable and
packet
tracing, result
compilation,
and examples
for extending
NS2. Two
appendices
provide the
details of
scripting
language Tcl,
OTcl and AWK,
as well object
oriented
programming
used
extensively in
NS2.
*A Developer's
Guide* Packt
Publishing Ltd
A guide to
completing
Python

projects for those ready to take their skills to the next level Python Projects is the ultimate resource for the Python programmer with basic skills who is ready to move beyond tutorials and start building projects. The preeminent guide to bridge the gap between learning and doing, this book walks readers through the "where" and "how" of real-world Python programming with practical,

actionable instruction. With a focus on real-world functionality, Python Projects details the ways that Python can be used to complete daily tasks and bring efficiency to businesses and individuals alike. Python Projects is written specifically for those who know the Python syntax and lay of the land, but may still be intimidated by larger, more complex projects. The

book provides a walk-through of the basic set-up for an application and the building and packaging for a library, and explains in detail the functionalities related to the projects. Topics include: *How to maximize the power of the standard library modules *Where to get third party libraries, and the best practices for utilization *Creating, packaging, and reusing libraries within

and across projects
 *Building multi-layered functionality including networks, data, and user interfaces
 *Setting up development environments and using virtualenv, pip, and more
 Written by veteran Python trainers, the book is structured for easy navigation and logical progression that makes it ideal for individual, classroom, or corporate training. For Python

developers looking to apply their skills to real-world challenges, Python Projects is a goldmine of information and expert insight.
The Art of UNIX Programming
 "O'Reilly Media, Inc."
 As an open operating system, Unix can be improved on by anyone and everyone: individuals, companies, universities, and more. As a result, the very nature of Unix has been altered over

the years by numerous extensions formulated in an assortment of versions. Today, Unix encompasses everything from Sun's Solaris to Apple's Mac OS X and more varieties of Linux than you can easily name. The latest edition of this bestselling reference brings Unix into the 21st century. It's been reworked to keep current with the broader state of Unix in today's world and highlight

the strengths of this operating system in all its various flavors. Detailing all Unix commands and options, the informative guide provides generous descriptions and examples that put those commands in context. Here are some of the new features you'll find in Unix in a Nutshell, Fourth Edition Solaris 10, the latest version of the SVR4-based operating system, GNU/Linux,

and Mac OS X Bash shell (along with the 1988 and 1993 versions of ksh) tsch shell (instead of the original Berkeley csh) Package management programs, used for program installation on popular GNU/Linux systems, Solaris and Mac OS X GNU Emacs Version 21 Introduction to source code management systems Concurrent system Subversion version control system

GDB debugger As Unix has progressed, certain commands that were once critical have fallen into disuse. To that end, the book has also dropped material that is no longer relevant, keeping it taut and current. If you're a Unix user or programmer, you'll recognize the value of this complete, up-to-date Unix reference. With chapter overviews, specific examples, and detailed command.

<p><i>Tcl/Tk in a Nutshell</i> Pearson Education India What others in the trenches say about The Pragmatic Programmer... “The cool thing about this book is that it’s great for keeping the programming process fresh. The book helps you to continue to grow and clearly comes from people who have been there.” —Kent Beck, author of <i>Extreme Programming Explained: Embrace</i></p>	<p>Change “I found this book to be a great mix of solid advice and wonderful analogies!” —Martin Fowler, author of <i>Refactoring and UML Distilled</i> “I would buy a copy, read it twice, then tell all my colleagues to run out and grab a copy. This is a book I would never loan because I would worry about it being lost.” —Kevin Ruland, Management Science, MSG-Logistics “The wisdom and practical experience of</p>	<p>the authors is obvious. The topics presented are relevant and useful.... By far its greatest strength for me has been the outstanding analogies—tracer bullets, broken windows, and the fabulous helicopter-based explanation of the need for orthogonality, especially in a crisis situation. I have little doubt that this book will eventually become an excellent source of useful</p>
---	--	---

information for journeymen programmers and expert mentors alike.” —John Lakos, author of Large-Scale C++ Software Design “This is the sort of book I will buy a dozen copies of when it comes out so I can give it to my clients.” —Eric Vought, Software Engineer “Most modern books on software development fail to cover the basics of what makes a great software developer, instead

spending their time on syntax or technology where in reality the greatest leverage possible for any software team is in having talented developers who really know their craft well. An excellent book.” —Pete McBreen, Independent Consultant “Since reading this book, I have implemented many of the practical suggestions and tips it contains. Across the

board, they have saved my company time and money while helping me get my job done quicker! This should be a desktop reference for everyone who works with code for a living.” —Jared Richardson, Senior Software Developer, iRenaissance, Inc. “I would like to see this issued to every new employee at my company....” —Chris Cleeland, Senior Software Engineer,

Object Computing, Inc. “If I’m putting together a project, it’s the authors of this book that I want. . . . And failing that I’d settle for people who’ve read their book.” —Ward Cunningham
 Straight from the programming trenches, The Pragmatic Programmer cuts through the increasing specialization and technicalities of modern software development to examine the core

process-- taking a requirement and producing working, maintainable code that delights its users. It covers topics ranging from personal responsibility and career development to architectural techniques for keeping your code flexible and easy to adapt and reuse. Read this book, and you'll learn how to Fight software rot; Avoid the trap of duplicating knowledge; Write flexible, dynamic, and

adaptable code; Avoid programming by coincidence; Bullet-proof your code with contracts, assertions, and exceptions; Capture real requirements; Test ruthlessly and effectively; Delight your users; Build teams of pragmatic programmers; and Make your developments more precise with automation. Written as a series of self-contained sections and filled with entertaining

anecdotes, thoughtful examples, and interesting analogies, The Pragmatic Programmer illustrates the best practices and major pitfalls of many different aspects of software development. Whether you're a new coder, an experienced programmer, or a manager responsible for software projects, use these lessons daily, and you'll quickly see improvements in personal productivity, accuracy, and

job satisfaction. You'll learn skills and develop habits and attitudes that form the foundation for long-term success in your career. You'll become a Pragmatic Programmer. **Python GUI Programmin g with Tkinter** Morgan Kaufmann No-nonsense and practical, yet with wit and charm. A joy to read." - Dan Sanderson, Software Developer, Amazon.com ""Shows style, not just facts-

valuable."" - Brian Downs, former Training Director, Lucent Technologies ""Brilliant, never tedious-highly recommended !"" -Jon Allen, Maintainer of perldoc.perl.org ""You could have chosen no better primer than this book."" - Damian Conway, from the Foreword Perl is a complex language that can be difficult to master. Perl advocates boast that ""There's More Than One Way To Do It,"" but

do you really want to learn several ways of saying the same thing to a computer? To make Perl more accessible, Dr. Tim Maher has over the years designed and taught an essential subset of the language that is smaller, yet practical and powerful. With this engaging book you can now benefit from ""Minimal Perl,"" even if all you know about Unix is grep. You will learn how to write simple Perl commands-

many just one-liners-that go far beyond the limitations of Unix utilities, and those of Linux, MacOS/X, etc. And you'll acquire the more advanced Perl skills used in scripts by capitalizing on your knowledge of related Shell resources. Sprinkled throughout are many Unix-specific Perl tips. This book is especially suitable for system administrators , webmasters, and software developers.

Programming Ruby Pearson Education
Freely available source code, with contributions from thousands of programmers around the world: this is the spirit of the software revolution known as Open Source. Open Source has grabbed the computer industry's attention. Netscape has opened the source code to Mozilla; IBM supports Apache; major database vendors have ported their

products to Linux. As enterprises realize the power of the open-source development model, Open Source is becoming a viable mainstream alternative to commercial software. Now in Open Sources, leaders of Open Source come together for the first time to discuss the new vision of the software industry they have created. The essays in this volume offer insight into how the Open Source

movement works, why it succeeds, and where it is going. For programmers who have labored on open-source projects, Open Sources is the new gospel: a powerful vision from the movement's spiritual leaders. For businesses integrating open-source software into their enterprise, Open Sources reveals the mysteries of how open development builds better software, and how

businesses can leverage freely available software for a competitive business advantage. The contributors here have been the leaders in the open-source arena: Brian Behlendorf (Apache) Kirk McKusick (Berkeley Unix) Tim O'Reilly (Publisher, O'Reilly & Associates) Bruce Perens (Debian Project, Open Source Initiative) Tom Paquin and Jim Hamerly (mozilla.org, Netscape) Eric

<p>Raymond (Open Source Initiative) Richard Stallman (GNU, Free Software Foundation, Emacs) Michael Tiemann (Cygnus Solutions) Linus Torvalds (Linux) Paul Vixie (Bind) Larry Wall (Perl) This book explains why the majority of the Internet's servers use open- source technologies for everything from the operating system to Web serving and email. Key</p>	<p>technology products developed with open- source software have overtaken and surpassed the commercial efforts of billion dollar companies like Microsoft and IBM to dominate software markets. Learn the inside story of what led Netscape to decide to release its source code using the open-source mode. Learn how Cygnus Solutions builds the world's best compilers by</p>	<p>sharing the source code. Learn why venture capitalists are eagerly watching Red Hat Software, a company that gives its key product -- Linux -- away. For the first time in print, this book presents the story of the open- source phenomenon told by the people who created this movement. Op en Sources will bring you into the world of free software and show you the revolution. <i>Design</i></p>
--	---	--

<i>Patterns</i>	Chapter 8:	Extensions
Addison-	Procedure	and Packages
Wesley	Techniques	Chapter 20:
Professional	Chapter 9:	Programming
Machine	Namespaces	Tools Chapter
generated	Chapter 10:	21: Debugging
contents note:	Basic TclOO	and
Chapter 1:	Chapter 11:	Optimization
Tcl/Tk	Advanced	techniques
Features	TclOO Chapter	Chapter 22:
Chapter 2:	12: Packages	Tips and
The Mechanics	and modules	Techniques .
of Using the	Chapter 13:	Programmin
Tcl and Tk	Introduction to	g in Lua
Interpreters	Tk Graphics	"O'Reilly
Chapter 3:	Chapter 14:	Media, Inc."
Introduction to	Overview of	You need a
the Tcl	the canvas	graphical user
Language	Widget	interface, and
Chapter 4: File	Chapter 15:	it needs to run
System, Disk	The text	on multiple
I/O and	widget and	platforms. You
Sockets	htmlib	don't have
Chapter 5:	Chapter 16:	much time,
Using Strings	Themed	and you're not
and Lists	Widgets	a wizard with
Chapter 6:	Chapter 17: Tk	X/Motif, the
Basic list,	Megawidgets	Win32 GUI, or
array and dict	Chapter 18:	the Mac GUI.
Chapter 7:	Writing a Tcl	The project
Advanced List,	Extension	seems
array and dict	Chapter 19:	impossible,

but with Tcl/Tk it's simple and fun. The Tcl scripting language and the Tk toolkit create a powerful programming environment for building graphical user interfaces. With two lines of code you can create a simple button; with two hundred lines of code, a desktop calculator; and with a thousand lines of code, an industrial-strength groupware calendar and appointment minder. Your applications

run on all of the major platforms: UNIX, Windows 95/NT, and Macintosh. You can even embed your programs in a Web page to make them available online. Mark Harrison and Michael McLennan, two noted Tcl/Tk experts, combine their extensive experience in this practical programming guide. It is ideal for developers who are acquainted with the basics of Tcl/Tk and are

now moving on to build real applications. *Effective Tcl/Tk Programming* shows you how to build Tcl/Tk applications effectively and efficiently through plenty of real-world advice. It clarifies some of the more powerful aspects of Tcl/Tk, such as the packer, the canvas widget, and binding tags. The authors describe valuable design strategies and coding techniques

that will make your Tcl/Tk projects successful. You will learn how to: Create interactive displays with the canvas widget Create customized editors with the text widget Create new geometry managers, like tabbed notebooks or paned windows Implement client/server architectures Handle data structures Interface with existing applications Package Tcl/Tk code into reusable libraries

Deliver Tcl/Tk applications that are easy to configure and install Embed applications in a Web page Build applications that will run on multiple platforms Throughout the book, the authors develop numerous applications and a library of reusable components. Learn from their approach, follow their strategies, and steal their code for your own applications! But don't

bother retyping all of the examples. 0201634740B 04062001 [Programming in Python 3](#) IGI Global Find out how to create visually stunning and feature-rich applications by empowering Python's built-in Tkinter GUI toolkit Key Features Explore Tkinter's powerful features to easily design and customize your GUI application Learn the basics of 2D and 3D animation in

GUI applications. Learn to integrate stunning Data Visualizations using Tkinter Canvas and Matplotlib. Book Description Tkinter is a lightweight, portable, and easy-to-use graphical toolkit available in the Python Standard Library, widely used to build Python GUIs due to its simplicity and availability. This book teaches you to design and build graphical user interfaces that

are functional, appealing, and user-friendly using the powerful combination of Python and Tkinter. After being introduced to Tkinter, you will be guided step-by-step through the application development process. Over the course of the book, your application will evolve from a simple data-entry form to a complex data management and visualization tool while maintaining a clean and robust design.

In addition to building the GUI, you'll learn how to connect to external databases and network resources, test your code to avoid errors, and maximize performance using asynchronous programming. You'll make the most of Tkinter's cross-platform availability by learning how to maintain compatibility, mimic platform-native look and feel, and build executables for

deployment across popular computing platforms. By the end of this book, you will have the skills and confidence to design and build powerful high-end GUI applications to solve real-world problems. What you will learn Implement the tools provided by Tkinter to design beautiful GUIs Discover cross-platform development through minor customization s in your existing application Visualize

graphs in real time as data comes in using Tkinter's animation capabilities Use PostgreSQL authentication to ensure data security for your application Write unit tests to avoid regressions when updating code Who this book is for This book will appeal to developers and programmers who would like to build GUI-based applications. Knowledge of Python is a prerequisite.

Tkinter GUI Programming by Example Effective Tcl/Tk Programming Writing Better Programs with Tcl and Tk Leverage the power of Python and its de facto GUI framework to build highly interactive interfaces Key Features The fundamentals of Python and GUI programming with Tkinter. Create multiple cross-platform projects by integrating a host of third-party libraries and tools. Build beautiful

and highly-interactive user interfaces that target multiple devices. Book Description Tkinter is a modular, cross-platform application development toolkit for Python. When developing GUI-rich applications, the most important choices are which programming language(s) and which GUI framework to use. Python and Tkinter prove to be a great combination. This book will

get you familiar with Tkinter by having you create fun and interactive projects. These projects have varying degrees of complexity. We'll start with a simple project, where you'll learn the fundamentals of GUI programming and the basics of working with a Tkinter application. After getting the basics right, we'll move on to creating a project of slightly increased complexity,

such as a highly customizable Python editor. In the next project, we'll crank up the complexity level to create an instant messaging app. Toward the end, we'll discuss various ways of packaging our applications so that they can be shared and installed on other machines without the user having to learn how to install and run Python programs. What you will learn Create a scrollable

frame via
theCanvas
widget Use
the pack
geometry
manager
andFrame
widget to
control layout
Learn to
choose a data
structurefor a
game Group
Tkinter
widgets, such
asbuttons,
canvases, and
labels Create
a highly
customizableP
ython editor
Design and
lay out a chat
window Who
this book is for
This book is
for beginners
to GUI
programming
who haven't
used Tkinter
yet and are

eager to start
building great-
looking and
user-friendly
GUIs. Prior
knowledge of
Python
programming
is expected.
*Unix in a
Nutshell*
Addison-
Wesley
Professional
Practical
Programming
in Tcl/Tk, 4th
edition
Authoritative
coverage of
every Tcl and
Tk command
in the core
toolkits State-
of-the-art Tk
GUI coverage
for Tcl, Perl,
Python, and
Ruby
developers
Covers all key
Tcl 8.4

enhancements
: VFS,
internationaliz
ation and
performance
improvements
, new widgets,
and much
more Covers
multi-
threaded Tcl
applications
and Starkits, a
revolutionary
way to
package and
deploy Tcl
applications
The world's
#1 guide to
Tcl/Tk has
been
thoroughly
updated to
reflect
Tcl/Tk8.4's
powerful
improvements
in
functionality,
flexibility, and
performance!

Brent Welch, Ken Jones, and Jeffrey Hobbs, three of the world's leading Tcl/Tk experts, cover every facet of Tcl/Tk programming, including cross-platform scripting and GUI development, networking, enterprise application integration, and much more. Coverage includes: Systematic explanations and sample code for all Tcl/Tk 8.4 core commands Complete Tk GUI development guidance--

perfect for developers working with Perl, Python, or Ruby Insider's insights into Tcl 8.4's key enhancements : VFS layer, internationalized font/character set support, new widgets, and more Definitive coverage of TclHttpd web server--written by its creator New ways to leverage Tcl/Tk 8.4's major performance improvements Advanced coverage: threading, Safe Tcl, Tcl script library,

regular expressions, and namespaces Whether you're upgrading to Tcl/Tk 8.4, or building GUIs for applications created with other languages, or just searching for a better cross-platform scripting solution, Practical Programming in Tcl and Tk, Fourth Edition delivers all you need to get results! Elements of Reusable Object-Oriented Software (Adobe

Reader) John Wiley & Sons Python is rapidly becoming the de facto standard language for systems integration. Python has a large user and developer-base external to the neuroscience community, and a vast module library that facilitates rapid and maintainable development of complex and intricate systems. In this Research Topic, we highlight recent efforts to develop Python

modules for the domain of neuroscience software and neuroinformatics: - simulators and simulator interfaces - data collection and analysis - sharing, re-use, storage and databasing of models and data - stimulus generation - parameter search and optimization - visualization - VLSI hardware interfacing. Moreover, we seek to provide a representative overview of existing mature Python

modules for neuroscience and neuroinformatics, to demonstrate a critical mass and show that Python is an appropriate choice of interpreter interface for future neuroscience software development.
Tcl and the Tk Toolkit
Manning Publications
A tutorial and reference to the object-oriented programming language for beginning to experienced programmers, updated for version 1.8,

describes the language's structure, syntax, and operation, and explains how to build applications. Original. (Intermediate) TCP/IP Illustrated Pearson Education The Art of UNIX Programming poses the belief that understanding the unwritten UNIX engineering tradition and mastering its design patterns will help programmers of all stripes to become better

programmers. This book attempts to capture the engineering wisdom and design philosophy of the UNIX, Linux, and Open Source software development community as it has evolved over the past three decades, and as it is applied today by the most experienced programmers. Eric Raymond offers the next generation of "hackers" the unique opportunity to learn the connection between UNIX

philosophy and practice through careful case studies of the very best UNIX/Linux programs. *A Complete Introduction to the Python Language* Frontiers Media SA Scripting with Python makes you productive and increases the reliability of your scientific work. Here, the author teaches you how to develop tailored, flexible, and efficient working environments

built from small programs (scripts) written in Python. The focus is on examples and applications of relevance to computational science: gluing existing applications and tools, e.g. for automating simulation, data analysis, and visualization; steering simulations and computational experiments; equipping programs with graphical user interfaces; making computational

Web services; creating interactive interfaces with a Maple/Matlab-like syntax to numerical applications in C/C++ or Fortran; and building flexible object-oriented programming interfaces to existing C/C++ or Fortran libraries. Minimal Perl Addison-Wesley Professional This book addresses the recent developments in systems maintenance research and practices

ranging from technicality of systems evolution to managerial aspects of the topic, including issues such as evolving legacy systems to e-business, applying patterns for reengineering legacy systems to web, architectural recovery of legacy systems, evolving legacy systems into software components. **55 Specific Ways to Improve Your**

Programs and Designs

Springer Science & Business Media
The classic guide to UNIX® programming-completely updated! UNIX application programming requires a mastery of system-level services. Making sense of the many functions-more than 1,100 functions in the current UNIX specification-is a daunting task, so for years programmers have turned to

Advanced UNIX Programming for its clear, expert advice on how to use the key functions reliably. An enormous number of changes have taken place in the UNIX environment since the landmark first edition. In Advanced UNIX Programming, Second Edition, UNIX pioneer Marc J. Rochkind brings the book fully up to date, with all-new, comprehensive coverage including:

POSIX Solaris™ Linux® FreeBSD Darwin, the Mac™ OS X kernel And more than 200 new system calls
Rochkind's fully updated classic explains all the UNIX system calls you're likely to need, all in a single volume!
Interprocess communication, networking (sockets), pseudo terminals, asynchronous I/O, advanced signals, realtime, and threads
Covers the system calls

you'll actually use-no need to plow through hundreds of improperly implemented, obsolete, and otherwise unnecessary system calls! Thousands of lines of example code include a Web browser and server, a keystroke recorder/player, and a shell complete with pipelines, redirection, and background processes. Emphasis on the practical-ensuring portability, avoiding pitfalls, and

much more! Since 1985, the one book to have for mastering UNIX application programming has been Rochkind's Advanced UNIX Programming. Now completely updated, the second edition remains the choice for up-to-the-minute, in-depth coverage of the essential system-level services of the UNIX family of operating systems. [Linux Shell Scripting Cookbook](#) Packt

Publishing Ltd Python 3 is the best version of the language yet: It is more powerful, convenient, consistent, and expressive than ever before. Now, leading Python programmer Mark Summerfield demonstrates how to write code that takes full advantage of Python 3's features and idioms. The first book written from a completely "Python 3" viewpoint, Programming

in Python 3 brings together all the knowledge you need to write any program, use any standard or third-party Python 3 library, and create new library modules of your own. Summerfield draws on his many years of Python experience to share deep insights into Python 3 development you won't find anywhere else. He begins by illuminating Python's "beautiful heart": the

eight key elements of Python you need to write robust, high-performance programs. Building on these core elements, he introduces new topics designed to strengthen your practical expertise—on e concept and hands-on example at a time. This book's coverage includes Developing in Python using procedural, object-oriented, and functional programming paradigms Creating

custom packages and modules Writing and reading binary, text, and XML files, including optional compression, random access, and text and XML parsing Leveraging advanced data types, collections, control structures, and functions Spreading program workloads across multiple processes and threads Programming SQL databases and key-value

DBM files
Utilizing
Python's
regular
expression
mini-language
and module
Building
usable,
efficient, GUI-
based
applications
Advanced
programming
techniques,
including
generators,
function and
class
decorators,
context
managers,
descriptors,
abstract base
classes,
metaclasses,
and more
Programming
in Python 3
serves as both
tutorial and
language
reference, and
it is
accompanied
by extensive
downloadable
example
code—all of it
tested with
the final
version of
Python 3 on
Windows,
Linux, and
Mac OS X.
*A Desktop
Quick
Reference*
Addison-
Wesley
Professional
Over 100
great recipes
to effectively
learn Tcl/Tk
8.5.
Open Sources
Pearson
Education
The Tcl
language and
Tk graphical
toolkit are
simple and
powerful
building
blocks for
custom
applications.
The Tcl/Tk
combination is
increasingly
popular
because it lets
you produce
sophisticated
graphical
interfaces with
a few easy
commands,
develop and
change scripts
quickly, and
conveniently
tie together
existing
utilities or
programming
libraries. One
of the
attractive
features of
Tcl/Tk is the
wide variety of
commands,

many offering a wealth of options. Most of the things you'd like to do have been anticipated by the language's creator, John Ousterhout, or one of the developers of Tcl/Tk's many powerful extensions. Thus, you'll find that a command or option probably exists to provide just what you need. And that's why it's valuable to have a quick

reference that briefly describes every command and option in the core Tcl/Tk distribution as well as the most popular extensions. Keep this book on your desk as you write scripts, and you'll be able to find almost instantly the particular option you need. Most chapters consist of alphabetical listings. Since Tk and mega-widget

packages break down commands by widget, the chapters on these topics are organized by widget along with a section of core commands where appropriate. Contents include: Core Tcl and Tk commands and Tk widgets C interface (prototypes) Expect [incr Tcl] and [incr Tk] Tix TclX BLT Oratcl, SybTcl, and Tclobc