

C U S I N G -KERMIT

S E C O N D E D I T I O N

Communication Software

Expanded
and
Updated
for
Version 6.0

FRANK da CRUZ • CHRISTINE M. GIANONE

Using *C-Kermit*

Communication Software

Second Edition

Frank da Cruz

and


Christine M. Gianone

Digital Press

Boston • Oxford • Johannesburg • Melbourne • New Delhi • Singapore

For information on obtaining a hardcopy of this book, and on all Digital Press publications available, contact our World Wide Web home page at: <http://www.bh.com/digitalpress>

©1997 Butterworth-Heinemann

 A member of the Reed Elsevier group

All rights reserved.

Digital Press™ is an imprint of Butterworth-Heinemann.

All trademarks found herein are property of their respective owners.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher.

Library of Congress Cataloging-in-Publication Data

Da Cruz, Frank, 1944-

Using C-Kermit / Frank da Cruz, Christine M. Gianone. – 2nd ed.
p. cm.

Includes bibliographic references and index.

ISBN 1-55558-164-1

1. C-Kermit. 2. Communication software. I. Gianone, Christine M. II. Title

TK5105.9.D33 1997

005.7'13—dc20

96-38354

CIP

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library.

The publisher offers special discounts on bulk orders of this book.

For information, please contact:

Manager of Special Sales

Butterworth-Heinemann

225 Wildwood Avenue

Woburn, MA 01801-2041

Tel: 781-904-2500

Fax: 781-904-2620

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America

Contents

Preface *xvii*

Chapter 1 Introduction	1
Why Kermit?	2
How Kermit Works	4
Capabilities of C-Kermit	8
Capabilities of Popular Kermit Programs	10
Kermit Software Versions	11
How to Get Kermit Software and Information	11

Chapter 2 Running C-Kermit	19
Starting C-Kermit	19
Exiting from C-Kermit	21
Entering Interactive Commands	21
Command Files	31
Describing Kermit's Commands	33
Some Basic C-Kermit Commands	39
The C-Kermit Initialization File	43
Commands for Controlling Commands	46
Summary Tables	49

Chapter 3 Making the Connection	51
Using C-Kermit in Local Mode	54
A Test Drive	54
Overview of Connection Establishment	57
Making a Serial Connection	57
Direct Serial Connections	63
Dialed Serial Connections	65
Troubleshooting Dialed Connections	74
Command Summary	80
Chapter 4 Configuring Modems	81
What If My Modem Is Not on Kermit's List?	83
Important Settings	84
Changing Things	85
Modifying Modem Commands	87
Adding a New Modem Type	90
Where to Put Modem Commands	92
Modem Command List	93
Modem-Related Variables	94
Chapter 5 Using the Dialing Directory	95
Dialing Directory Format	96
Looking up and Dialing Numbers	101
Adding Prefixes and Suffixes	102
About Country and Area Codes	102
Using Portable Entries	103
Exceptions to the Rules and How to Handle Them	110
Long-Distance Carriers and Calling-Card Numbers	112
Traveling Tips	114
Old-Format Directories	115
Command Summary	115
Chapter 6 Using Networks	117
Making Network Connections	118
TCP/IP Networks	120
Making LAT Connections	128
X.25 Networks	129
Named Pipes	133
NETBIOS	134
Using the Network Directory	135
Command-Line Options for Network Connections	138
Command Summary	139

Chapter 7 Using The Services Directory	141
Login Macros	142
Connection Details	144
Sample Services Directory	145
Chapter 8 Terminal Connection	147
The CONNECT Command	148
Closing the Connection	151
CONNECT-Mode Keyboard Escape Commands	152
Setting Terminal Parameters	157
Coordinating Screen Dimensions	160
Key Mapping	161
C-Kermit in the Middle	162
Automatic Actions While in CONNECT Mode	164
Logging and Debugging Your Terminal Session	165
Command Summary	168
Chapter 9 The Basics of File Transfer	171
Getting Started	171
Basic File Transfer Commands	172
Easy File Transfer Examples	177
Local-Mode File Transfer	182
Interrupting a File Transfer	186
Transferring Text Files	189
Transferring Binary Files	190
Building a Send List	191
File Names	193
Transferring Files between Similar Systems	194
Directory Names	194
Filename Collisions	196
Incomplete File Transfers	197
Recovering from Interrupted File Transfers	197
File Attributes	200
Odds and Ends	203
Keeping a Record of Your File Transfers	204
Summary	205
Chapter 10 Solving File Transfer Problems	207
Parity	208
Speed and Flow Control in the Full Duplex Environment	210
Half Duplex Communication	213
The Pause that Refreshes	214
Noise and Interference	214

Timeouts	216	
Transparency Problems	217	
IBM Mainframe Linemode Communication	219	
IBM Mainframe Full-Screen Communication	221	
For X.25 Users Only	223	
If Files Are Corrupt after Transfer	224	
Collecting the Evidence	225	
Chapter 11 Using a Kermit Server	227	
Configuring the Server	228	
Starting the Server	231	
Sending Commands to Kermit Servers	233	
Transmission of Variables	242	
Turning the Tables	242	
Command Summary	243	
Chapter 12 High-Speed Kermit File Transfer	245	
Overview of the Kermit Protocol	246	
A Word about Efficiency	248	
Analyzing Kermit's Performance	249	
Data Compression	252	
Long Packets	254	
Sliding Windows	256	
Single and Locking Shifts	260	
Control Character (Un)Prefixing	262	
Case Study: Achieving the Best Transfer Rate	266	
Summary and Conclusion	270	
Chapter 13 Automatic File Transfer and Command Execution	273	
Remote-Control File Transfer	273	
Automatic Uploading	274	
Automatic Uploading, Part Deux	275	
Automatic Command Execution	278	
Chapter 14 Using External Protocols	281	
Sending Files	283	
Receiving Files	283	
Things to Watch Out for	284	
Command Summary	284	

Chapter 15	Transferring Files without a Protocol	285
	Downloading to C-Kermit	286
	Uploading from C-Kermit	288
	Encoding 8-Bit Data Files for Transmission	293
	Command Summary	294
Chapter 16	International Character Sets	295
	Proprietary Character Sets	296
	Standard Character Sets	297
	International Characters in Commands	300
	International Characters in Terminal Emulation	301
	Transferring International Text Files	306
	Translating without Transferring	323
	One-Sided Translation	324
	Labor-Saving Devices	325
	Command Summary	326
Chapter 17	Command Files, Macros, and Variables	327
	Command Files Revisited	328
	Macros	332
	Macro Arguments	339
	A Macro Sampler	343
	Variables	347
Chapter 18	Programming Commands	365
	The IF Command	365
	The STOP and END Commands	375
	The GOTO Command	377
	Structured Programming	380
	Built-in Functions	386
	Can We Talk?	401
	Using Escape Sequences	404
	Reading and Writing Files and Commands	408
	Programming Considerations	411
	Summary of Built-in Functions	413
Chapter 19	Script Programming	415
	Automated Connection Establishment	416
	Synchronization Commands	419
	Constructing a Login Script for VMS	428
	A UNIX Login Script	436
	An IBM Mainframe Linemode Login Script	437
	An IBM Mainframe Fullscreen Login Script	438

Login Scripts for Commercial Data Services	440
A Directory of Services	443
Unattended File Transfer	447
Passwords and Security versus Automation	449
Automatic File Transfer Recovery	453
Calling an Alphanumeric Pager	454
Some Things We Didn't Tell You	457
The SCRIPT Command	459
Appendix I Starting and Stopping C-Kermit	461
Program Termination	463
Command-Line Options	465
Command-Line Examples	475
Environment Variables	478
Appendix II A Condensed Guide to Serial Data Communications	479
Character Format and Parity	479
Modems	482
Cables and Connectors	492
Summary	496
Appendix III UNIX C-Kermit	497
Installation	497
Using UNIX C-Kermit	504
Command Summary	520
Appendix IV VMS C-Kermit	521
Preparing Your VMS Session for C-Kermit	521
Using VMS C-Kermit	523
Command Summary	542
Appendix V AOS/VS C-Kermit	543
Using C-Kermit in AOS/VS	544
Appendix VI Other C-Kermit Versions	549
Amiga C-Kermit	549
OS-9 C-Kermit	552
Others	554
Appendix VII Character Set Tables	555
The ASCII and ISO 646 IRV Character Set	557
7-Bit Control Characters	558

7-Bit Roman Character Sets	559
West European Character Sets	560
East European Character Sets	564
Cyrillic Character Sets	566
Hebrew Character Sets	570
Appendix VIII Country Codes	573
Appendix IX Hexification Programs	577
Appendix X Shift-Out/Shift-In Filter	579
Acronyms and Abbreviations	581
References	595
Trademarks and Copyrights	601
Index	605

Illustrations

Figure 1-1	Remote and Local Computers	4
Figure 1-2	Connecting the Local and Remote Computers	5
Figure 1-3	Logging in to the Remote Computer	6
Figure 1-4	Transferring a File	7
Figure 3-1	C-Kermit in Remote Mode	52
Figure 3-2	C-Kermit in Local Mode	52
Figure 3-3	C-Kermit in the Middle	53
Figure 3-4	A Direct Connection	64
Figure 3-5	A Dialed Connection	66
Figure 8-1	Terminal Connection	148
Figure 8-2	Command and Terminal Bytesize	158
Figure 9-1	Upload and Download	176
Figure 9-2	Uploading a File	176
Figure 9-3	Downloading a File	179
Figure 9-4	C-Kermit's Fullscreen File Transfer Display	184
Figure 9-5	Kermit Text File Conversion	189
Figure 10-1	Character Formats	208
Figure 10-2	Hardware Flow Control	211
Figure 10-3	IBM Mainframe Linemode Example	220
Figure 12-1	Kermit Packet Format	246
Figure 12-2	Stop-and-Wait Packet Exchange	247
Figure 12-3	Sliding Windows	257
Figure 16-1	Structure of an 8-Bit Latin Alphabet	299

Figure 16-2	Terminal Character Set Translation	304
Figure 16-3	International Text File Transfer	307
Figure 16-4	Linguini Transfer	311
Figure 18-1	Returning from Nested Command Files	376
Figure 18-2	Mass Mailing Script	412
Figure 19-1	The VMSLOGIN Macro	433
Figure 19-2	Sample IBM 3270 Login Screen	439
Figure 19-3	The TAPMSG Macro	456
Figure II-1	Character Formats	480
Figure II-2	Asynchronous Character Transmission Format	480
Figure II-3	Computers Connected by Modems	483
Figure II-4	Asynchronous Modem Cable Schematic	492
Figure II-5	Data Connectors	493
Figure II-6	Asynchronous Null Modem Schematics	494
Figure VII-1	Structure of a Standard 8-Bit Character Set	556

Tables

Table 1-1	Kermit Software Features as of September 1996	10
Table 1-2	Kermit Software, Listed by Computer Type	12
Table 2-1	Special Characters in C-Kermit Commands	30
Table 2-2	C-Kermit Initialization File Name	44
Table 2-3	Basic C-Kermit Commands	49
Table 2-4	Summary of Backslash Codes	49
Table 3-1	Sample Dialout Device Names	56
Table 3-2	Modem Types Known to C-Kermit	68
Table 4-1	Modem Capabilities	83
Table 4-2	Modem Variables	94
Table 5-1	Telephone Number Modifiers for Hayes Modems	99
Table 5-2	Dialing Category Codes	100
Table 5-3	C-Kermit Dial Status Codes	116
Table 6-1	Commonly Used Assigned TCP Port Numbers	121
Table 6-2	X.3 Numeric PAD Parameters and Values	137
Table 8-1	C-Kermit CONNECT-Mode Escapes	152
Table 9-1	Special Characters in C-Kermit File Specifications	173
Table 9-2	Kermit Packet Types	186
Table 12-1	Dangerous Control Characters	265
Table 12-2	V.34 ZIP-File Transfer Performance	267
Table 12-3	Kermit File Transfer Feature Summary	271
Table 14-1	SET PROTOCOL Command Defaults	282
Table 16-1	Decimal Character Codes for Accented Capital Letter A	296

Table 16-2	7-Bit National Character Sets, Differences from ASCII	298
Table 16-3	The ISO Latin Alphabets	299
Table 16-4	Right Half of Latin Alphabet 1	300
Table 16-5	DEC Multinational Character Set	300
Table 16-6	ANSI Escape Sequence Formats	303
Table 16-7	C-Kermit File Character Sets	307
Table 16-8	Language-Specific Transliteration Rules	320
Table 17-1	Built-in Variables	361
Table 18-1	\Feval() Operators	395
Table 18-2	Selected VT100 Escape Sequences	405
Table 18-3	Built-in Functions	413
Table 19-1	Notation for SCRIPT Command	460
Table I-1	C-Kermit Return Codes	463
Table I-2	C-Kermit Command-Line Options	476
Table I-3	Environment Variables Meaningful to C-Kermit	478
Table II-1	RS-232 / V.24 Modem Signals and Pins	484
Table II-2	Selected Hayes Smartmodem 2400 Commands	487
Table II-3	Modem Modulation Techniques	488
Table II-4	Modem Error Correction and Compression Techniques	490
Table III-1	Setting Your Terminal Type in UNIX	505
Table III-2	UNIX Terminal Control Characters	506
Table IV-1	VMS Kermit SET FILE Commands	540
Table IV-2	PC – VMS File Transfer	541
Table V-1	AOS/VS Template Characters	548
Table VII-1	Character Codes of ASCII and ISO 646 IRV	557
Table VII-2	7-Bit C0 Control Characters	558
Table VII-3	7-Bit National Character Sets, Differences from ASCII	559
Table VII-4	West European Character Sets	560
Table VII-5	East European Character Sets	564
Table VII-6	Cyrillic Character Sets	566
Table VII-7	Hebrew Character Sets	570
Table VIII-1	Country Codes	573

Preface

“Who Is Kermit and Why Is He in My Computer?” asked a (now not so) recent headline [60].¹ Kermit is not a “he” at all, but rather an inanimate, genderless, yet friendly computer software package that lets just about any two computers in the world communicate effectively with each other, no matter how they may differ in size, age, appearance, location, power, architecture, manufacture, or nationality.

This book describes *C-Kermit*, quite possibly the world’s most portable communications software program, for UNIX computer systems (hundreds of different ones); Digital Equipment Corporation (Open)VMS on both VAX and Alpha; PCs with Windows 95, Windows NT, or OS/2; Data General AOS/VS, Stratus VOS, the Commodore Amiga, the Atari ST, and computers with the QNX and OS-9 realtime operating systems. The UNIX version of C-Kermit runs on all known implementations of UNIX (see page 16) and on computers ranging from PCs to large mainframes and supercomputers.

C-Kermit software offers you terminal connection, error-free file transfer and management, script programming, and comprehensive support for national and international character sets, over a wide variety of communication methods including direct and dialed serial connections and (in most versions) TCP/IP, X.25, or other networks. C-Kermit’s full-featured script programming language operates consistently across all of C-Kermit’s platforms and over all types of connections. It allows routine, complex, or time-consuming communications tasks to be executed for you automatically.

¹Numbers in brackets refer to entries in the References on page 595.

C-Kermit transfers text and binary files faithfully and efficiently with any other kind of computer. The Kermit file transfer protocol takes care of synchronization, error detection and correction, file format and character set conversion, and myriad details you should never have to worry about. It was designed to work in even the most hostile communication environments, where other protocols fail. C-Kermit's Kermit implementation, along with that of MS-DOS Kermit, with which it was codeveloped, is the premiere and definitive rendition of the Kermit protocol.

The Kermit file transfer protocol was originally designed in 1981 by Frank da Cruz and Bill Catchings at Columbia University, which has been “Kermit headquarters” ever since, and extended over the years by the authors and others — principally Joe Doupnik of Utah State University and John Chandler of the Harvard / Smithsonian Astronomical Observatory — to meet the evolving needs of the people who depend on it. Because the Kermit protocol is well documented [21], easy to implement, robust, extensible, and adaptable to almost any style of communication and any computer architecture, it has long since taken its place as a worldwide de facto standard for reliable data transfer.

Acknowledgments

C-Kermit was written by Frank da Cruz of Columbia University with contributions from hundreds of other developers and testers, all of whom have our deepest thanks, with our sincere apologies to anyone else we might have overlooked (U = University, locations are in the USA unless otherwise indicated, and note that affiliations or locations might have changed since the contribution was made):

Chris Adie (Edinburgh U, Scotland); Robert Adsett (U of Waterloo, Canada); Larry Afrin (Clemson U); Jeffrey Altman (Columbia U); Greg Andrews (Telebit Corp); Barry Archer (U of Missouri); Bengt Andersson (ABC-Klubben, Sweden); Robert Andersson (International Systems A/S, Oslo, Norway); Chris Armstrong (Brookhaven National Laboratory); William Bader (Software Consulting Services, Nazareth, PA); Fuat Baran (Columbia U); Stan Barber (Rice U); Jim Barbour (U of Colorado); Donn Baumgartner (Dell Computer Corp); Nelson Beebe (U of Utah); Karl Berry (UMB); Mark Berryman (SAIC); Dean W Bettinger (State U of New York); Gary Bilkus; Peter Binderup (Denmark); David Bolen (Advanced Networks and Services, Inc.) Marc Boucher (U of Montreal, Canada); Charles Brooks (EDN); Bob Brown; Mike Brown (Purdue U); Rodney Brown (COCAM, Australia); Jack Bryans (California State U at Long Beach); Mark Buda (DEC); A. Butrimenko (ICSTI, Moscow); Fernando Cabral (Padrão IX, Brasília, Brazil); Björn Carlsson (Stockholm U Computer Centre QZ, Sweden); Bill Catchings (formerly of Columbia U); Bob Cattani (formerly of Columbia U); Davide Cervone (Rochester U, NY); Seth Chaiklin (Denmark); John Chandler (Harvard U/Smithsonian Astronomical Observatory, Cambridge, MA); Bernard Chen (UCLA); Andrew A Chernov (RELCOM

Team, Moscow); John L Chmielewski (AT&T, Lisle, IL); Howard Chu (U of Michigan); Bill Coalson (McDonnell Douglas); Kenneth Cochran; Bertie Coopersmith (London, England); Chet Creider (U of Western Ontario, Canada); Alan Crosswell (Columbia U); Jeff Damens (formerly of Columbia U); Mark Davies (Bath U, England); Sin-itirou Dezawa (Fujifilm, Japan); Clarence Dold (Pope Valley & Napa, CA); Joe R. Doupnik (Utah State U); Frank Dreano (US Navy); John Dunlap (U of Washington); Alex Dupuy (SMART.COM), Jean Dutertre (DEC France and *Club Kermit*); David Dyck (John Fluke Mfg Co.); Stefaan Eeckels (Statistical Office of the European Community, CEC, Luxembourg); Paul Eggert (Twin Sun, Inc.); Bernie Eiben (DEC); Peter Eichhorn (assyst, Gesellschaft für Automatisierung, Software und Systeme mbH, Kirchheim bei München, Germany); Kristoffer Eriksson (Peridot Konsult AB, Örebro, Sweden); John Evans (IRS, Kansas City); Glenn Everhart (DEC); Vincent Fatica (Syracuse U); Charlie Finan (Cray Research, Darien, CT); Herm Fischer (Encino, CA); Carl Fongheiser (CWRU); Mike Freeman (Bonneville Power Authority); Marcello Frutig (Catholic U, São Paulo, Brazil); Hirofumi Fujii (Japan National Laboratory for High Energy Physics, Tokyo); Chuck Fuller (Westinghouse); Andy Fyfe (Caltech); Christine M. Gianone (Columbia U); Joseph (Yossi) Gil (Technion, Haifa, Israel); John Gilmore (UC Berkeley); Madhusudan Giyyarpuram (HP France); Rainer Glaschick (Siemens AG, Paderborn); William H. Glass; Hunter Goatley (Western Kentucky U); Malka Gold (Columbia U); German Goldszmidt (IBM); Chuck Goodheart (NASA); Alistair Gorman (New Zealand); Juri Gonastaev (ICSTI, Moscow); Richard Gratton (Australian Defence Force Academy); Chris Green (Essex U, England); Alan Grieg (Dundee Tech, Scotland); Volkmar Grote (Hamburg, Germany); Valdemar Gunnarson (Iceland); Yekta Gursel (MIT); Jim Guyton (Rand Corp); Vesa Gynther (Finland); Michael Haertel; Marion Hakanson (ORST); John Hamilston (Iowa State U); Steen Hammerum (U of København, Denmark); Simon Hania (Netherlands); Darryl Hankerson (Auburn University); Stan Hanks (Rice U); Ken Harrenstein (SRI); Eugenia Harris (Data General); David Harrison (Kingston Warren Corporation); James Harvey (Indiana/Purdue U); Rob Healey; Chuck Hedrick (Rutgers U); Ron Heiby (Motorola Computer Group); Steve Hemminger (Tektronix); Christian Hemsing (Rheinisch-Westfälisch Technische Hochschule, Aachen, Germany); Andrew Herbert (Monash U, Australia); Mike Hickey (ITI); Dan Hildebrand (QNX Software Systems Inc., Ontario); R.E. Hill; Bill Homer (Cray Research); Ray Hunter (The ex-Wollongong Group); Randy Huntziger (US National Library of Medicine); Larry Jacobs (Transarc); Xander Jansen (SURFnet, Utrecht, Netherlands); Graham Jenkins (TABCORP, Melbourne, Australia); Steve Jenkins (Lancaster U, England); Bo Johansson (Sweden); Dave Johnson (Gradient Technologies); Mark Johnson (Apple Computer); Jyke Jokinen (Tampere U of Technology, Finland); Eric Jones (AT&T); Luke Jones (AT&T); Peter Jones (U of Quebec, Montreal, Canada); Phil Julian (SAS Institute); Peter Kabal (U of Quebec); Mic Kaczmarczik (U of Texas at Austin); Sergey Kartashoff (Institute of Precise Mechanics & Computer Equipment, Moscow); Howie Kaye (Columbia U); Rob Kedoin (Linotype Co., Hauppauge, NY); Phil Keegstra; Mark Kennedy (IBM); Terry Kennedy (St Peter's College, Jersey City, NJ); Carlo Kid (Technical U of Delft, Netherlands); Tim

Kientzle; Ted Kilgore (Auburn U); Paul Kimoto (Cornell U); Douglas Kingston; Lawrence Kirby (Wiltshire, England); John Klensin (United Nations University); Kurt Klingbeil (Province of Alberta); Tom Kloos (Sequent Computer Systems); Jim Knutson (U of Texas at Austin); John T. Kohl (BSDI); Scott Kramer (SRI International); John Kraynack (US Postal Service); David Kricker (Encore Computer); Thomas Krueger (U of Wisconsin at Milwaukee); Bo Kullmar (Central Bank of Sweden, Kista, and ABC-Klubben, Stockholm); R. Brad Kummer (AT&T Bell Labs, Atlanta, GA); John Kunze (UC Berkeley); David Lane (Stratus Computer Inc); Russell Lang (Monash U, Australia); Bob Larson (USC); Bert Laverman (Groningen U, Netherlands); Steve Layton; David Lawyer (UC Irvine); David LeVine (National Semiconductor Corp.); Daniel S. Lewart (UIUC); S.O. Lidie (Lehigh U); Tor Lillqvist (Helsinki U, Finland); Robert Lipe (Arnet Corp); Benny Löfgren (DIAB, Sweden); Dean Long; Mike Long (Analog Devices); Kevin Lowey (U of Saskatchewan, Canada); Andy Lowry (Columbia U); James Lummel (Caprica Telecomputing Resources); David MacKenzie (Environmental Defense Fund, U of Maryland); John Mackin (U of Sidney, Australia); Martin Maclaren (Bath U, England); Chris Maio (formerly of Columbia U); Montserrat Mané (HP France); Fulvio Marino (Olivetti, Ivrea, Italy); Arthur Marsh (DIRCSA, Australia); Peter Mauzey (AT&T); Tye McQueen (Utah State U); Ted Medin (NOSC); Ajay Mehta (DEC); Melissa Metz (Columbia U); Hellmuth Michaelis (Hanseatischer Computerservice GmbH, Hamburg, Germany); Leslie Mikesell (American Farm Bureau); Gary Mills (U of Manitoba, Canada); Martin Minow (DEC); Pawan Misra (Bellcore); Ken Mizialko (IBM, Manassas, VA); Ray Moody (Purdue U); Bruce J. Moore (Allen-Bradley Co); Steve Morley (Convex); Peter Mossel (Columbia U); Tony Movshon (NYU); Lou Muccioli (Swanson Analysis Systems); Dan Murphy; Neal P. Murphy (Harsco Systems, Wonder Lake, IL); Gary Mussar (Bell Northern Research); John Nall (Florida State U); Jack Nelson (U of Pittsburgh); Jim Noble (PRC, Inc.); Ian O'Brien (Bath U, England); John Owens; Michael Pins (Iowa Computer Aided Engineering Network); André Pirard (U of Liège, Belgium); Paul Placeway (Ohio State U); Piet Plomp (Groningen U, Netherlands); Ken Poulton (HP Labs); Manfred Prange (Oakland U); Christopher Pratt (APV Baker, UK); Frank Prindle (NADC); Tony Querubin (U of Hawaii); Phil Race (ICL, Manchester, England); Jean-Pierre Radley; Anton Rang; Ruth Raphaeli (Columbia U); Scott Ribe; Alan Robiette (Oxford U, England); Michel Robitaille (U of Montreal, Canada); Huw Rogers (Schweizerische Kreditanstalt, Zürich); Kai Uwe Rommel (Technische Universität München, Germany); Judith Rosenhouse (Technion, Haifa, Israel); Larry Rosenman (Irving, TX); Jay Rouman (U of Michigan); Jack Rouse (SAS Institute); Stew Rubenstein (Harvard U); Cory Sane (Medical U of SC); Bill Schilit (Columbia U); Ulli Schlüter (RWTH Aachen, Germany); Michael Schmidt (U of Paderborn, Germany); Eric Schnoebelen (Convex); Benn Schreiber (DEC); Dan Schullman (DEC); John Schultz (3M); Steven Schultz (GTE Government Systems Corp); APPP Scorer (Leeds Polytechnic, England); Gordon Scott (Micro Focus, Newbury, England); Jay Sekora (Princeton U); Gisbert W. Selke (Wissenschaftliches Institut der Ortskrankenkassen, Bonn, Germany); David Singer (IBM Almaden Research Labs); David Sizeland (U of London Medical

School, England); Friðrik Skulason (Iceland); Rick Sladkey; Dave Slate; Bradley Smith (UCLA); Fred Smith (Merk); Richard Smith (California State U); Ryan Stanifer (UNT); Bertil Stenström (Stockholm U Computer Centre QZ, Sweden); James Sturdevant (CAP GEMINI AMERICA, Minneapolis, MN, and Medtronic, Inc., Fridley, MN); Margarita Suarez (Columbia U); Peter Svanberg (Kungl. Tekniska Högskolan, Sweden); James Swenson (Accu-Weather, Inc., State College, PA); Chris Sylvain (U of Maryland); Andy Tanenbaum (Vrije U, Amsterdam, Netherlands); Tim Theisen (U of Wisconsin); Glen Thobe; Lee Tibbert (DEC); Markku Toijala (Helsinki U of Technology, Finland); Teemu Torma (Helsinki U of Technology, Finland); Linus Torvalds (Helsinki, Finland); Rick Troxel (US National Institutes of Health); Warren Tucker (Tridom Corp, Mountain Park, GA); Dave Tweten (NASA); G. Uddeborg (Sweden); Walter Underwood (Ford Aerospace); Pieter Van Der Linden (Centre Mondial, Paris, France); Ge van Geldorp (Netherlands); Fred van Kempen (MINIX User Group, Voorhout, Netherlands); Johan van Wingen (Leiden, Netherlands); Wayne Van Pelt (General Electric Corporate Research and Development); Mark Vasoll (Oklahoma State U); Don Vickers (DECUS); Konstantin Vinogradov (ICSTI, Moscow); Paul Vixie (DEC); Bernie Volz (Process Software); Eduard Vopicka (Prague School of Economics, Czech Republic); Dimitri Vulis (D&M Consulting Services, NYC); Roger Wallace (Raytheon); Stephen Walton (California State U at Northridge); Jamie Watson (Adasoft, Switzerland); Rick Watson (U of Texas); Robert Weiner (Programming Plus, New York City); Lauren Weinstein (Vortex Technology); Clark Wierda (Illuminati Online, Austin TX); David Wexelblat (AT&T Bell Labs); Bill Whitney (DEC); Joachim Wiesel (U of Karlsruhe, Germany); Lon Willett (U of Utah); Michael Williams (UCLA); Nate Williams (U of Montana); David Wilson; Joellen Windsor (U of Arizona); Patrick Wolfe (Kuck & Associates, Inc.); Gregg Wonderly (Oklahoma State U); Farrell Woods (Concurrent); David Woolley (London, England); Jack Woolley (SCT Corp); Frank Wortner; Ken Yap (U of Rochester, NY); John Zeeff (Ann Arbor, MI); Martin Zinser (Gesellschaft für Schwerionenforschung GSI, Darmstadt).

The second edition brings *Using C-Kermit* up to date with C-Kermit 6.0. The changes are too numerous to list, but noteworthy among them is the new “intelligent and portable” dialing directory (Chapter 5), whose design required an education in the art of dialing telephones. For their help with this work, grateful thanks to Pat Townsend for moderating the comp.dcom.telecom newsgroup, a goldmine of telephony information and expertise, and to those who assisted directly, especially: Toby Nixon (Program Manager, Windows Telephony, Microsoft Corporation); Dave Kramer (Head of International Consumer Markets, Sprint International); Bernard Lyons (IS Specialist, Claris Ireland); Ken Levitt (Informed Computer Solutions, Wayland, MA); Mark Brader (SoftQuad Inc., Toronto); Martin Kealey (Auckland, New Zealand); and David Woolley (London, England).

Big thanks, as always, to the “Kermitees,” who do such an amazing job of technical and customer support while running our day-to-day production operations so smoothly: Max Evarts, Andy Newcomb, and the generations who came before, especially Bob Tschudi, Peter Howard, Lucy Lee, and Ken Suh.

A special note of appreciation to Jeffrey Altman, for some years a (prodigious) volunteer contributor to the Kermit Project, and now a full-time developer on the Kermit team, for massive contributions to C-Kermit 6.0 — and not just code, but energy, enthusiasm, and great ideas too.

Thanks to everyone who helped in the production of this book: Liz McCarthy, Karen Pratt, and Mike Cash at Digital Press / Butterworth Heinemann / Reed Elsevier; to the former team of the original Digital Press, wherever they are now, who produced the first edition of this book as well as our other Kermit books; to Marjan Baće and Lee Fitzpatrick of Manning Software, the publisher of Kermit 95, for which this book serves as technical reference; and to Guy Steele of Sun Microsystems for permission to reproduce his *Telnet Song* and to Deborah Cotton of the ACM for the right to do so.

And finally, thanks to our management and colleagues at Columbia University for their encouragement and support, especially Vaçe Kundakçı, Deputy Vice President for Academic Information Systems, and Elaine Sloan, Vice President for Information Services and University Librarian; to Bruce Gilchrist and Howard Eskin, directors of our organization during the early days of Kermit; to Alan Crosswell and the AcIS Systems Group for taking care of our well-known server, `kermit.columbia.edu`, and for help in many other forms; and to Lee Lidofsky, a Great Teacher, for a timely push in a good direction, a long time ago.

*Frank da Cruz and Christine M. Gianone
The Kermit Project, Columbia University
New York City, September 1996*

fdc@columbia.edu, cmg@columbia.edu

IMPORTANT:

This edition of *Using C-Kermit* is current as of C-Kermit version 6.0.192. Any changes made after this edition was published, but before the next edition is published, are listed in an online file that accompanies the C-Kermit software. The name of the file depends on the operating system and distribution method, but will generally begin or end with the characters “UPD” — `CKERMIT.UPD`, `CKCKER.UPD`, `UPDATES.DOC`. Be sure to consult this file to learn about new features.

Other supplemental online documentation files include “beware” files, whose names end with “.BWR”, which contain current information about bugs and restrictions, with suggested workarounds, for C-Kermit in general (`CKCKER.BWR`), and for particular implementations (`ckuker.bwr` for UNIX, `CKVKER.BWR` for VMS, etc). There are also files containing detailed installation instructions for each operating system (`CK*INS.DOC`), plus a configuration guide (`CKCCFG.DOC`) and a program logic manual (`CKCPLM.DOC`).

A special note of appreciation to Jeffrey Altman, for some years a (prodigious) volunteer contributor to the Kermit Project, and now a full-time developer on the Kermit team, for massive contributions to C-Kermit 6.0 — and not just code, but energy, enthusiasm, and great ideas too.

Thanks to everyone who helped in the production of this book: Liz McCarthy, Karen Pratt, and Mike Cash at Digital Press / Butterworth Heinemann / Reed Elsevier; to the former team of the original Digital Press, wherever they are now, who produced the first edition of this book as well as our other Kermit books; to Marjan Baće and Lee Fitzpatrick of Manning Software, the publisher of Kermit 95, for which this book serves as technical reference; and to Guy Steele of Sun Microsystems for permission to reproduce his *Telnet Song* and to Deborah Cotton of the ACM for the right to do so.

And finally, thanks to our management and colleagues at Columbia University for their encouragement and support, especially Vaçe Kundakçı, Deputy Vice President for Academic Information Systems, and Elaine Sloan, Vice President for Information Services and University Librarian; to Bruce Gilchrist and Howard Eskin, directors of our organization during the early days of Kermit; to Alan Crosswell and the AcIS Systems Group for taking care of our well-known server, `kermit.columbia.edu`, and for help in many other forms; and to Lee Lidofsky, a Great Teacher, for a timely push in a good direction, a long time ago.

*Frank da Cruz and Christine M. Gianone
The Kermit Project, Columbia University
New York City, September 1996*

fdc@columbia.edu, cmg@columbia.edu

IMPORTANT:

This edition of *Using C-Kermit* is current as of C-Kermit version 6.0.192. Any changes made after this edition was published, but before the next edition is published, are listed in an online file that accompanies the C-Kermit software. The name of the file depends on the operating system and distribution method, but will generally begin or end with the characters “UPD” — `CKERMIT.UPD`, `CKCKER.UPD`, `UPDATES.DOC`. Be sure to consult this file to learn about new features.

Other supplemental online documentation files include “beware” files, whose names end with “.BWR”, which contain current information about bugs and restrictions, with suggested workarounds, for C-Kermit in general (`CKCKER.BWR`), and for particular implementations (`ckuker.bwr` for UNIX, `CKVKER.BWR` for VMS, etc). There are also files containing detailed installation instructions for each operating system (`CK*INS.DOC`), plus a configuration guide (`CKCCFG.DOC`) and a program logic manual (`CKCPLM.DOC`).