C-Kermit 8.0

We love our city -- as it was and as it is.

Network and Serial Communications for Unix, VMS, and other platforms.

Version: 8.0.201
Date: 8 February 2002

This page last updated: Tue Apr 2 14:22:57 2002

- LICENSING
- DOCUMENTATION
- SUMMARY OF NEW FEATURES
- CASE STUDIES AND TUTORIALS
- DOWNLOAD
- BINARIES

C-Kermit is a combined network and serial communication software package offering a consistent, medium-independent, cross-platform approach to connection establishment, terminal sessions, file transfer, character-set translation, numeric and alphanumeric paging, and automation of communication tasks. C-Kermit includes:

- Along with Kermit 95, the fastest and most advanced implementation of the Kermit file transfer protocol available anywhere.
- A powerful, portable, easy-to-use script programming language to automate all your routine

communications tasks.
- Consistent operation over serial connections (direct or dialed) and network connections (TCP/IP and in some cases also LAT or X.25) -- on a huge selection of hardware and software platforms.
- Secure authentication and strong encryption.
- Built-in FTP and HTTP clients plus an SSH interface
- Configurability as an SSH Subsystem <-- **New!**
- Character-set translation in both file transfer *and* online sessions, for Western- and Eastern-European languages, Cyrillic, Greek, Hebrew, and Japanese, now including Unicode.
- Ability to send numeric and alphanumeric pages.

And lots more. C-Kermit is:

- "Year-2000 compliant" as of version 6.0 (1996).
- "Euro compliant" as of version 7.0 (2000).

C-Kermit 8.0 is available for practically every known variation and version of Unix, past and present, on every architecture, and for DEC / Compaq VMS / OpenVMS on VAX and Alpha. Earlier releases of C-Kermit remain available for other platforms and operating systems:

- Data General AOS/VS
- Stratus VOS
- Microware OS-9
- Plan 9 from Bell Labs
- The Commodore Amiga
- The Atari ST

In Unix, C-Kermit can be thought of as a user-friendly and powerful alternative to cu, tip, minicom, uucp, ftp, ftpd, telnet, ktelnet, rlogin, ssh, find, grep, iconv, recode, expect, wget, sendpage, bc, and to some extent even Lisp, your shell, and/or Perl; a single package for both network and serial communications, offering automation, helpfulness, and language features not found in most of the other packages, and with most of the same features available on all its non-Unix platforms (such as VMS), as well as in Kermit 95 on Windows 9x/NT/2000/XP.

Links:

- Announcement
- Specifications
- C-Kermit FAQ
- C-Kermit Script Library
- Reviews
- G-Kermit
- Kermit 95
- Other Kermit Programs
- Kermit Project Home Page

LICENSING

As of version 7.0, C-Kermit has a **new license**, CLICK HERE to read it. Summary:

1. **Personal or Internal Use:** C-Kermit is freely downloadable by individuals for their own use and by organizations (companies, universities, government agencies, hospitals, etc) for *internal use*. In this case, we ask (but do not require) you to also purchase the user manual to *(a)* let you get the most out of the software; *(b)* reduce the load on our help desk; and *(c)* contribute some revenue to the nonprofit and entirely self-supporting Kermit Project.

2. **Free Unix Distributions:** C-Kermit may be included in "free Unix" distributions such as GNU/Linux, FreeBSD, NetBSD, and OpenBSD. See the license for details.

3. **Redistribution:** Redistribution of C-Kermit except as in (2) **must be licensed**. CLICK HERE for **NEW, EASIER, LOWER-COST** terms and conditions.

[Top] [Download] [Binaries] [Kermit Home]

DOCUMENTATION

The **user manual** for C-Kermit 8.0 is (still) the book *Using C-Kermit*, Second Edition [Amazon], which is current with C-Kermit 6.0. Features new to version 7.0 are documented in the online Supplement to *Using C-Kermit*, Second Edition, for C-Kermit 7.0, and those new to 8.0 in C-Kermit 8.0 Supplement (see links just below), which should serve until the Third Edition is ready.

If you don't have the manual, please order it. It explains how to use C-Kermit, how to make connections, how to troubleshoot connection and file-transfer problems, how to handle character-set translation, how to write script programs, and lots more. It lets you get the most out of your software, it reduces the load on our help desk, and sales of the manual are the primary source of funding for continued C-Kermit development and support.

Online Information Resources:

- **C-Kermit 7.0 Manual Supplement**
- **C-Kermit 8.0 Manual Supplement - New!**
- **TUTORIAL** (html) **- New!**
- **TUTORIAL** (pdf) - New
- **TUTORIAL** (nroff == manpage) - New
- **Kermit Security Reference** - Updated
- **Telnet Client Reference** - Updated
- **Internet Kermit Service User Documentation** - Updated
- **Internet Kermit Service Administrator Documentation** - Updated
- **General C-Kermit Hints, Tips, Patches** - Updated and HTMLized
- **Unix C-Kermit Hints and Tips** - Updated and HTMLized
- **Unix C-Kermit Installation Instructions** - Updated and HTMLized
- **VMS C-Kermit Hints and Tips** - Updated and HTMLized
- **VMS C-Kermit Installation Instructions** - Updated and HTMLized
- **Compilation / Configuration Options** - Updated and HTMLized
- **Program Logic Manual** - Updated and HTMLized

[Top] [Download] [Binaries] [Kermit Home]

SUMMARY OF NEW FEATURES

New Terms and Conditions for Commercial Redistribution

CLICK HERE for details.

Built-In FTP Client (Unix only)

- Wouldn't it be nice if all the power, flexibility, and ease of use of the Kermit command and scripting language were available in an FTP client, not to mention features like security, character-set translation, recovery, update, directory tree transfer, atomic file movement, automatic text/binary mode switching, and almost everything else C-Kermit can do? Now they are. CLICK HERE for an overview. *The built-in FTP client is available only in the Unix version of C-Kermit 8.0, and will also be included in Kermit 95 1.1.21; anybody who is interested in adapting it to VMS or other operating systems, please contact us*.

Built-In HTTP 1.1 Client (Unix only)

- HTTP connections can be open at the same time as other connections.
- Both persistent and "one-shot" connections are supported.
- Fully scriptable.
- SSL/TLS security is configurable.
- CLICK HERE for documentation.

Security

- **SSH:** Unix C-Kermit now includes an interface to the external SSH program, so you can use all of C-Kermit's features (command language, scripting, file transfer, character-set translation) on SSH connections.
- Due to relaxations in USA export law, secure versions of C-Kermit are available in source-code form, supporting **Kerberos IV**, **Kerberos V**, **SSL/TLS**, and **SRP**.
- **ZLIB compression** has been added to SSL/TLS for improved peformance.
- Kermit's SSL/TLS security method can now use not only X.509 certificates but also Kerberos 5 credentials.
- CLICK HERE for the new Kermit Security Reference.

Dialing

- Fast generic AT dialing is now the default.
- Many new modem types added.
- C-Kermit now supports RFC 2217: **Telnet Com Port Control**. When dialing out from an access (reverse terminal) server that supports this option (such as a Cisco 2509 or a Linux-based sredird server), and that was contacted using Telnet protocol, C-Kermit can now set and examine the port's serial communications parameters: speed, flow control, modem signals, etc, and can send BREAK signals from the server's serial port.

Terminal Sessions

- Timestamped session log now available.
- Now 8-bit clean by default rather than by request.
- The pseudo-keepalive feature previously found only in K95 is now available also in most Unix

versions of C-Kermit. This lets you have Kermit automatically time out when the terminal session has been inactive for a specified amount of time and take any of several desired action upon an idle timeout: return to command mode, hangup, exit, or send the character, string, or Telnet negotiation of your choice.

- CLICK HERE for details.

Automatic Text/Binary Mode Switching

- C-Kermit did this in version 7.0 by matching filenames against lists of well-known filetypes (like `.exe`, `.zip`, `.txt`, etc), which is not foolproof; for example, is `foo.doc` a text file or a (binary) Microsoft Word document?. Now the *contents* of each file is quickly prescanned to determine its type. This occurs when sending files with Kermit, Zmodem, or FTP. CLICK HERE for details.

Proxy support

- Kermit can now make terminal or FTP connections through Proxy Servers supporting the HTTP CONNECT command.

Scripting Improvements

- **Learned scripts** (automatic script writing)
- **LISP-like S-Expressions** (primarily for arithmetic)
- Date-time and timezone conversions and arithmetic
- Script-level Ctrl-C trapping
- Functions to convert between strings and arrays, with grouping
- Ability to pass arrays as arguments to macros
- Lots of new built-in variables and functions
- Cleaner passing of arguments to command files
- Group/pattern UNDEFINE
- More flexible MINPUT
- Faster script execution and improved troubleshooting

Command Parsing Improvements

- Better handling of file and directory names that contain spaces
- Natural use of doublequotes
- A way to choose how to handle unknown commands
- Improved file and directory name completion
- SHOW and SAVE (command) HISTORY

New or Improved Commands

- SET TERMINAL IDLE-TIMEOUT and IDLE-ACTION
- SET FILE STRINGSPACE and LISTSIZE
- CHMOD (Unix only)
- CLEAR KEYBOARD-BUFFER
- IF KBHIT and other new IF conditions
- COPY, RENAME, and TRANSLATE now operate on file groups
- New DELETE switches: /DIRECTORIES, /RECURSIVE, /TREE, /SUMMARY
- GREP, HEAD, TAIL
- SET ROOT (similar to Unix chroot)
- Improved character-set handling in the TYPE command

Bugs Fixed

- **Memory leaks and buffer attack vulnerabilities plugged.**
- Expect-like scripting of local programs via SET HOST /PTY was broken on some platforms.
- Solaris hardware flow control was broken.
- FreeBSD 4.0 + ncurses 5.0 problem.
- NetBSD modem signal sensing and control.
- SCO OSR5 and Unixware modem signal sensing and control.
- Transparent printing in Unix didn't always work.
- ANSWER 0 (wait forever for incoming call) broken.
- Problems with RETURN from right-recursive functions.
- SEND /COMMAND didn't work.
- Streaming file transfers couldn't be canceled.
- Improper interaction of SEND-LIST with GET.
- The MAIL command was broken.
- "`kermit -s *`" might have skipped some files.
- CP1251 erroneously treated like Latin-1.
- UTF-8 conversion didn't handle C1 control area properly.
- Lots of other smaller problems.

New to 8.0.201

- Installability as an SSH v2 subsystem
- A more natural user interface for the FTP client
- Up/Down Arrow keys for command recall
- SET ESCAPE now lets you choose 8-bit characters
- A more-flexible SHOW VARIABLES command
- and fixes a few bugs that surfaced since the 8.0 release

CLICK HERE to see the list of features that were new to C-Kermit 7.0.

[Top] [Download] [Binaries] [Kermit Home]

CASE STUDIES AND TUTORIALS

CLICK HERE to view the **new C-Kermit 8.0 tutorial**.

Case studies and tutorials are posted from time to time to the newsgroup news:comp.protocols.kermit.misc to showcase some of the new features of C-Kermit 7.0 and 8.0, with each posting also available on the website. Here's the index:

1. Cleaning Out Beta-Test Binaries (file management)
2. Kerbang Scripts (script construction and argument passing)
3. Autodownload (file transfer)
4. Automatic Text/Binary Mode Switching (file transfer)
5. Directory Recursion (file transfer and management)
6. Streaming (file transfer)
7. Internet Kermit Service (file transfer and management)
8. Unicode (character sets)

9. Printing (file transfer and management)
10. Atomic File Movement (file transfer and management)
11. C-Kermit Meets SSH (networking and security)
12. C-Kermit's Telnet Client (networking)
13. PPP Dialing (networking)
14. Character Sets (file transfer)
15. File Selection (file transfer and management)
16. Bootstrapping (installation)
17. Fun with Dates and Times (script programming)
18. File Timestamps and Permissions (file transfer)
19. An Easy Way to Get Files from IKSD (file transfer)
20. Lynx/Kermit Coordination (Part I)
21. Secure Telnet Using C-Kermit 7.1 With Stunnel
22. More Fun with Dates
23. Modem Pool Statistics
24. *(Check back later for more as we add new ones)*

[Top] [Kermit Home]

DOWNLOAD

[Source Code] [Individual Text Files] [Individual Binaries]

You can download C-Kermit source and/or binaries in tar, zip, or other archive format or as separate files. INSTALL PACKAGES ARE NOT AVAILABLE YET. THE C-KERMIT 8.0 CDROM IS NOT AVAILABLE YET but the C-Kermit 7.0 CDROM is still available.

> Install packages are contributed by users. So far, no C-Kermit 8.0 packages have been contributed. CLICK HERE for a general discussion of C-Kermit package making. HOWEVER, the Unix version of C-Kermit 8.0 includes its own new installation procedure, built into the makefile. See Section 5 of the Unix C-Kermit installation guide.

[Top] [Kermit Home]

Download and Build from Source Code

[Unix Build] [VMS Build]

You can **download C-Kermit 8.0 source and text files** directly from the Kermit Project in any of several archive formats by clicking on the following FTP links. The **complete archives** contain source code, build procedure, license, initialization files, CA certificates, manual page or help topic, initialization files, and plain-text information files extracted from the Web pages listed in the Documentation section: a complete distribution. The **text archives** contain everything but the source code; these are to be used when you download a prebuilt binary. The size of each archive file in megabytes is shown in each cell.

Archive	**Zip**	**Tar**	**Tar.Z**	**Tar.gz**
Unix Complete	2.7 `cku201.zip`	10.7 `cku201.tar`	3.9 `cku201.tar.Z`	2.7 `cku201.tar.gz`
Unix Text Only	0.7 `cku201txt.zip`	2.4 `cku201txt.tar`	0.9 `cku201txt.tar.Z`	0.7 `cku201txt.tar.gz`
VMS Complete	2.7 `ckv201.zip`			
VMS Text Only	0.6 `ckv201txt.zip`			

Besides those, we also have C-Kermit 7.0 archives for the following platforms that have not yet been updated to C-Kermit 8.0:

`ckl197.zip`
: Source code and text files for **Stratus VOS**, ZIP archive (text files in DOS format) (Version 7.0).

`ck9197.zip`
: Source code and text files for **Microware OS-9/68k**, ZIP archive (text files in DOS format). (Version 7.0).

Source code and text files are also available separately in the `kermit/f/` directory. These include files for platforms other than Unix and VMS, such as Stratus VOS, Data General AOS/VS, OS-9, the Amiga, etc. All files in this directory are text files; transfer them in text mode. See the `ckaaaa.txt` file for details. **NOTE:** The Unix, VMS, VOS, Amiga, OS-9, and AOS/VS source files are all at 8.0 level (even if only the Unix and VMS versions have actually been built). The others (Macintosh, Atari ST) have not yet been updated in a long while due to lack or programmers and/or platforms; **volunteers welcome.** And of course anybody interested in porting C-Kermit to new platforms are more than welcome to contact us about it; we'll be happy to get you started.

Trouble with FTP?

- **CLICK HERE** for FTP hints and tips; or:
- Forget FTP -- Use Kermit!

[Top] [Hints & Tips] [Kermit Home]

Unix Build Instructions

[Unix Hints & Tips] [Unix Installation Instructions]

1. **Make a fresh directory** and 'cd' to it. Example:

```
$ mkdir kermit
$ cd kermit
```

2. **Download** the appropriate source code archive.

3. If you downloaded a compressed tar file, **uncompress** it. Examples:

```
$ gunzip cku201.tar.gz
$ uncompress cku201.tar.Z
```

4. If you downloaded a tar archive, "**un-tar**" it. Example:

   ```
   $ tar xvf cku201.tar
   ```

   If you downloaded a Zip archive, unzip it. Example:

   ```
   $ unzip -a cku201.zip
   ```

5. Now you can **delete the tar archive** (or zip archive) if you wish:

   ```
   $ rm cku201.tar                 (or rm cku201.zip)
   ```

6. Read the comments at the top of the makefile to find out which target is appropriate for your computer and operating system, and then **give the appropriate "make" command.** Examples:

   ```
   $ make linux                    (Linux, almost any version)
   $ make freebsd44                (FreeBSD 4.4)
   $ make solaris8                 (Solaris 8 with cc)
   $ make solaris8g                (Solaris 8 with gcc)
   $ make irix62                   (IRIX 6.2)
   ```

7. To **install** the version of Kermit you have just built, read the installation instructions, especially if you had trouble with any of these steps, or if you will be using C-Kermit to dial out. The most convenient way to install is to the use makefile's `install` target ("make install"); read about it HERE. See the configuration options guide for information about compile-time options for customizing the feature set.

You can also download **individual Unix binaries** from the C-Kermit binaries table, but in case of library or other version mismatches, it is better to build from source code if you can.

[Top] [Hints & Tips] [Kermit Home]

VMS Build Instructions

[VMS Hints & Tips] [VMS Installation Instructions]

If you want to **install a prebuilt VMS binary**, then fetch the most appropriate **VMS binary** from the C-Kermit binaries table. Pick a VAX binary for a VAX or an Alpha binary for an Alpha. The VMS version number for the binary must be less than or equal to your VMS version. If you want to make TCP/IP connections, pick the binary for the appropriate TCP/IP product (TGV Multinet, DEC UCX, Process Software TCPware, etc), again with a version number less than or equal to yours; if none can be found, then try a UCX version (since most non-DEC TCP products include built-in UCX emulation). If you downloaded a prebuilt binary, also download the VMS C-Kermit text-file archive. Then read the installation instructions for VMS.

If you want to **build from source code**, fetch the VMS complete archive above if you have VMS-based unpacking tools, otherwise get the source files and text individually as described just below. NOTE: Unzip the Zip file with '`unzip -a`'.

1. **Make a fresh directory** and SET DEFAULT to it. Example:

   ```
   $ create/directory kermit
   $ set default [.kermit]
   ```

2. **Download** the VMS source code Zip archive.

3. **Unpack** the Zip archive. Example:

   ```
   $ unzip -a ckv201.zip
   ```

4. **Run the build procedure**:

   ```
   $ @ckvker
   ```

   If you have a pre-5.0 VMS release, use the "old" build procedure:

   ```
   $ @ckvold
   ```

   If you experience any trouble, read the comments at the top of the build procedure.

5. This makes a `WERMIT.EXE` file in your current directory. Start it with:

   ```
   $ r wermit
   ```

6. To **install** the version of Kermit you have just built, read the installation instructions.

[Top] [Hints & Tips] [Kermit Home]

Individual Binaries

[Binaries Table]

The C-Kermit binaries table is HERE. Before visiting the table, you should read this section.

When you download a prebuilt-Kermit binary, you should also download the C-Kermit text files, unpack them if necessary, and install them as desired. NOTE: In Unix, you can still use "make install", even if if you did not use the makefile to build your Kermit binary (the makefile as well as all the text files you need are in the text archive).

In the binaries table, filenames start with "ck" for C-Kermit, then one letter or digit to indicate the platform ("u" for Unix, "d" for Data General AOS/VS, "v" for VMS, "i" for Amiga, "9" OS-9, "p" for Plan 9, etc). After that comes a three-digit edit number:

188: Version `5A(188)`, November 1992 through September 1993.
189: Version `5A(189)`, September 1993 through October 1994.
190: Version `5A(190)`, October 1994 through September 1996.
192: Version `6.0.192`, September 1996 through December 1999.

193: Version `6.1.193`, November 1996 through June 1998.
194: Version `6.1.194`, June 1998 through December 1998.
195: Version `7.0.195`, January 1999 through August 1999.
196: Version `7.0.196`, September 1999 through final release 1 Jan 2000.
197: Version `7.0.197`, January-February 2000.
200: Version `8.0.200`, December 2001.
201: Version `8.0.201`, February 2002.

Then a possible test-version designator: "a" for Alpha or "b" for Beta, followed by the 2-digit test number. Examples:

```
cku200.xxx        C-Kermit 8.0.200 final release
cku200b04.xxx     C-Kermit 8.0.200 Beta.04
cku197.xxx        C-Kermit 7.0.197 final release
```

Test versions are included here only for platforms that do not have a final build available (usually because the machine disappeared or had an OS upgrade before the final C-Kermit release).

Note that edits 193, 194, 195, 198, and 199 were never formally released (191 was only for OS/2).

The rest of the name is platform-dependent; in Unix it's the name of the makefile target, optionally followed by specific hardware platform and/or OS version, when it makes a difference. In VMS it's the platform ("axp" (i.e. Alpha) or "vax"), then the VMS version number (e.g. "vms73"), and then TCP/IP product and version number (or "nonet" if TCP/IP support is not built in). And so on. VMS TCP/IP product codes are as follows:

`ucx`	DEC TCP/IP
`tgv`	TGV MultiNet
`pst`	Process Software TCPware
`twg`	The Wollongong Group WIN/TCP or PathWay
`cmu`	Carnegie-Mellon University CMU/IP

REMEMBER: It's often better to build your own binary than to run a prebuilt one, due to the ever-increasing likelihood of OS and/or library version mismatch.

After downloading, rename to "`kermit`" or "`kermit.exe`" (etc), as appropriate for your operating system and, if necessary, give execute permission, e.g. (in Unix):

```
$ mv cku201.linux-i386-rh7.2 kermit
$ chmod +x kermit
```

Also remember that before C-Kermit can be used to dial out from Unix, it will probably also be necessary to give the Kermit executable a certain owner and group, and to set it suid and/or sgid bits, to allow it access to the dialout device and/or lockfile directory (the same as any other dialout software, such as cu or minicom). Read Sections 10 and 11 of the Unix C-Kermit installation guide.

Notes on the Binaries:

1. "curses" refers to the fullscreen file-transfer display, used when transferring files over dialout or network connections. It's nice but it adds size and sometimes causes problems so if a "curses" version

gives you trouble, try a no-curses (NC) version. On platforms that have a choice between "traditional curses" and "new curses" (ncurses), you might also have a choice of binaries -- if one doesn't work, try the other.

2. Likewise, some binaries come in TCP/IP and non-TCP/IP versions. If you don't need to make TCP/IP connections with C-Kermit, choose the non-TCP/IP version (if available). A TCP/IP-enabled binary might not run on platforms that don't have TCP/IP installed.

3. And some binaries come in optimized and non-optimized versions; this refers to compile-time optimization. Choose the optimized version, but if it gives you trouble, try the corresponding non-optimized one, if available (some optimizers have bugs). When optimization is marked in the table, O means normal optimization, O- means no optimization, O+ means extra optimization.

4. Some binaries are available in gcc and non-gcc versions; that is, versions built using two different compilers. If one gives you trouble, try the other if available. Some compilers have bugs; some compilers support features that other ones don't.

5. Some HP-UX binaries (notably, the HP-UX 7.00 ones) are built for long filename (255 chars) file systems, others for short-filename (14 chars) systems; these are marked with SF and LF. Choose one that is appropriate for your file system.

6. Some Linux binaries are linked with libc, some with glibc. Pick the one that is appropriate for your Linux system. Better yet, just build from source ("make linux" should work on any Linux system).

> ***If you are able to make a binary not listed table***
> *(or that is listed, but is not current"), please contribute it to the archive.*

CLICK HERE to visit the C-Kermit binaries table.

Questions? Comments? -- Send e-mail.

[Top] [Kermit Home]

C-Kermit 8.0 / Columbia University / `kermit@columbia.edu` / 8 February 2002