



Kermit for Microsoft Windows

Version 0.80

General Topics

[Introduction](#)

[Getting Started](#)

[Using Kermit](#)

[Starting a New Kermit Session](#)

[Saving a Kermit Session](#)

[Opening a Saved Kermit Session](#)

[Configuration Options / Session Settings](#)

[Configuration Options / Terminal Emulation Settings](#)

[Configuration Options / Communications Settings](#)

[Configuration Options / Device Settings](#)

[Configuration Options / Protocol Settings](#)

[Configuration Options / Font Settings](#)

[About Kermit](#)

Introduction

This is a fairly complete implementation of the KERMIT communications protocol for Microsoft Windows. It includes most extensions to the protocol as well (Long Packets, Windowing, Server Mode, Attribute Packets, and Enhanced Error Checking).

Additionally, a fairly full featured terminal emulation capability is included. The terminal emulations are encapsulated in Dynamic Link Libraries which allow new emulations to be added with ease. A modem dialer is also included.

As in most MS Windows programs, cooperative multitasking is fully supported. You may even have two copies of the program running at one time on different serial ports!

Local Operation:	Yes
Remote Operation:	No
Login Scripts:	No*
Transfer text files:	Yes
Wildcard send:	Yes (plus directory listings)
File transfer interruption:	Yes
Filename collision avoidance:	No*
Can time out:	Yes
Eighth-bit prefixing:	Yes
Repeat count prefixing:	Yes
Alternate block checks:	Yes
Terminal emulation:	ANSI, VT-102, TTY
Communication settings:	Baud, Parity, Flow, etc.
Transmit BREAK:	Yes
Support for dialout modems:	Yes
IBM mainframe communication:	Yes
Transaction logging:	No*
Debug logging:	No*
Session logging:	No*
Raw file transmit:	No*
Act as server:	Yes
Talk to server:	Yes
Advanced server functions:	No*
Local file management:	No (available from Windows)
Command/Init files:	No*
Command macros:	No*
File attribute packets:	Yes
Extended packets:	Yes
Sliding Windows:	Yes

Those entries marked with an asterisk (*) are features which are expected to be available in Release 1.00. Unfortunately, the author has extremely limited time for this project and version 1.00 is going to take a while. New releases are distributed anytime significant improvements have been made to the application.

Note: Users are encouraged to review the enclosed ASCII text file README.TXT for complete file lists and update/version information.

See also...

[Getting Started](#)

[Using Kermit](#)

Getting Started

System Requirements

Kermit for Windows requires any of the existing flavors of MS Windows Version 3.1. These flavors include MS Windows, MS Windows for Workgroups, and MS Windows NT. Only the Intel variation of Windows NT is supported at this time. Kermit for Windows NT appears to run fine under the current beta release of Windows95 as well. Although Windows95 and Windows NT will run 16 bit applications (i. e., the standard 16-bit version of Kermit for Windows), it is preferable to run the NT (32 bit version).

Kermit for Windows supports both standard asynchronous (modem style) and TCP/IP (Telnet style) communications.

For asynchronous connectivity, Kermit for Windows relies upon the serial port drivers included with Windows. I have been told that aftermarket serial drivers for Windows work fine, but have not personally tested this. Support for Hayes compatible modems is also provided when using Asynchronous communications.

For TCP/IP connectivity, Kermit for Windows relies upon the WinSock library. Most TCP/IP software vendors now support the WinSock API, so this should not be much of a constraint. In theory all WinSock implementations should be compatible, but I have only tested the Microsoft WinSock libraries. Microsoft includes a WinSock library with Windows NT. They have also just released a free version of WinSock for Windows for Workgroups. This free WinSock implementation can be downloaded from the Microsoft Software Library on CompuServe or from <ftp.microsoft.com> on the Internet.

The memory and hard disk requirements of Kermit for Windows are extremely small and should almost never be a problem.

Installation/Setup

No special installation program is provided with this software. Installation is accomplished by simply creating a suitable directory and copying all files to the directory. The original release is distributed as a compressed file. Use PKUNZIP to decompress the files.

Starting with release 0.77, it is now also necessary to install a file in your Windows System directory. If you are using the 16-bit version, you need to move the file CTL3DV2.DLL to your Windows System (usually C:\WINDOWS\SYSTEM) directory. If you are using the 32-bit version, you need to move the file CTL3D32.DLL to your Windows NT System Directory (usually C:\WINNT\SYSTEM32). If you already have a copy of the DLL in the target directory, check the dates and only copy the one distributed with Kermit for Windows if its date is more recent than the existing copy. Either way, you **MUST** delete the DLL file from the directory you installed Kermit for Windows. If you don't you will get a nasty message every time you start Kermit indicating that the DLL file is incorrectly installed.

Note: Users are encouraged to review the enclosed ASCII text file README.TXT for complete file lists and update/version information.

An icon can easily be added to the Windows Program Manager to provide a menu entry for invoking Kermit. Follow the instructions provided with Windows for adding a program item. You may also wish to add a line to the [Extensions] section of your WIN.INI file. The line would look something like the following:

```
[Extensions]
krm=c:\kermit\kermit.exe ^.krm
```

This will allow you to start Kermit with a specific session file already loaded by double-clicking on the desired .KRM session file in the Windows File Manager. Kermit will automatically be invoked with the selected Session File loaded.

Program Startup

You must start Windows before running Kermit. Once Windows is running, you can start Kermit by double-clicking on its icon in the Program Manager (if you installed one) or by double-clicking on KERMIT.EXE in the Windows File Manager.

Kermit follows all Windows standards. Those familiar with other MS Windows applications should be able to use Kermit without trouble.

See also...

[Using Kermit](#)

Using Kermit

In order to keep track of different sets of configuration options, Kermit uses the concept of a communications session. When Kermit is first started, an untitled session is automatically initialized. All configuration settings are given their default values.

In most cases, you will need to change some of the configuration options prior to establishing a communications session. See the "Configuration Options" topics for more information.

See also...

[Starting a New Kermit Session](#)

[Saving a Kermit Session](#)

[Opening a Saved Kermit Session](#)

[Configuration Options / Session Settings](#)

[Configuration Options / Terminal Emulation Settings](#)

[Configuration Options / Communications Settings](#)

[Configuration Options / Device Settings](#)

[Configuration Options / Protocol Settings](#)

[Configuration Options / Font Settings](#)

Starting a New Kermit Session

Starting a new communications session is essentially the same thing as returning Kermit to the state it was in when it was first started. All configuration settings are restored to their default values.

To start a new communications session, choose New... from the File Menu. If you have not already saved the settings from the existing session, you will be reminded and given the option of saving them now.

In order to start a new session, any existing connection to a remote system must be terminated. If a connection is active at the time a new session is requested, you will be asked to confirm that you want to disconnect.

See also...

[Saving a Kermit Session](#)

[Opening a Saved Kermit Session](#)

As a convenience, Kermit will bring up the Session Settings dialog. This dialog allows you to enter a brief description of the session, choose the appropriate terminal emulation, and indicate if you want to automatically connect to the remote system any time this session is subsequently opened. See the "Configuration Options / Session Settings" topic for more information on this dialog.

See also...

[Configuration Options / Session Settings](#)

[Configuration Options / Terminal Emulation Settings](#)

[Configuration Options / Communications Settings](#)

[Configuration Options / Device Settings](#)

[Configuration Options / Protocol Settings](#)

[Configuration Options / Font Settings](#)

Saving a Kermit Session

Changes made to a communications session are not kept permanently until they are saved. When you save a session, Kermit will record all of the current configuration settings in a file. You must supply a standard DOS filename to save the settings in. By default the extension of the session save files is ".KRM".

To save the current configuration settings, choose Save or Save As... from the File Menu. Save As... allows you to specify a new file to save your settings in, whereas Save will save your settings to the currently active session save file (as identified on the application title bar). If there is no current session save file (title bar reads "(Untitled)"), then Save will function just like Save As... (it will prompt you for a file name to save the session in).

It is not necessary to disconnect an active connection in order to save your settings.

If a session file name is required, Kermit will display the standard Windows Save As dialog. This dialog functions identically to the Save As dialogs in most Windows applications. You must type in a filename or pick one from the list. If no filename extension is specified, Kermit will automatically supply an extension of ".KRM". It is recommended that you do not use a different extension, because the Open function identifies session files by the ".KRM" extension by default. Session save files may be stored in any directory, but the default directory is the save directory that the Kermit program is installed in.

See also...

[Getting Started](#)

[Starting a New Kermit Session](#)

[Opening a Saved Kermit Session](#)

Opening a Saved Kermit Session

To recall a session you saved previously, choose Open... from the File Menu. The standard Windows Open dialog will be displayed to allow you to choose the session save file to be opened.

By default, the Open dialog will display all files with an extension of ".KRM" in the directory that the Kermit program is stored in.

After choosing a session file to load, Kermit will check to see if the previous session had a connection active. If so, you will be asked to confirm that you want to disconnect. If you choose not to disconnect, you will be returned to the previous session with the connection still active.

If the session you have chosen to load has the Auto Connect on Open option set in the Session Settings dialog, Kermit will attempt to establish a connection with the remote system immediately.

It is also possible to Open a saved session at the time you invoke the Kermit program. You may specify the name of a session save file on the command line that is used to invoke Kermit. The specified session file will be automatically loaded instead of the default settings that would normally be loaded.

If you have set up Kermit as an Icon in the Program Manager, the command line is one of the entries in the Properties dialog for the Icon. This feature makes it possible to create several Icons in the Program Manager for Kermit and each one could start Kermit with a specific session file.

See also...

[Getting Started](#)

[Starting a New Kermit Session](#)

[Saving a Kermit Session](#)

Configuration Options / Session Settings

In order to keep track of different sets of configuration options, Kermit uses the concept of a communications session. When Kermit is first started, an untitled session is automatically initialized. All configuration settings are given their default values.

Setting configuration options is done by simply using a series of dialog boxes to tailor the configuration to your needs. All of the configuration settings are saved as part of the Session Save File and are automatically restored the next time the Session is opened.

To access any of the following configuration dialogs, simply choose the corresponding item from the Configure Menu.

Note that the Session Settings are the most important settings. This dialog configures the high-level configuration (communications type, emulation, and protocol). Additionally, the entries in this dialog (in some cases) determine the options that are available in the other configuration dialogs. For example, if you choose connection type async in the Session Settings, you will get options like baud rate, stop bits, parity, etc. in the Communications Settings Dialog. If you choose connection type tcpip, you will get a different dialog that has options like host, port, etc.

Also note that certain options in the configuration dialogs may be grayed out (and therefore not modifiable) at certain times. For example, the connection option in the Session Settings Dialog is grayed out when a connection is currently open. This is because it is not possible to switch to a different connection type without destroying the currently open connection. To change options like these, you must first disconnect, and then access the appropriate dialog.

Name

You may enter a brief (one line) description of the session. This is primarily useful to help identify what a given session is for.

Connection

A drop down list box presents all available connection types. Choose async if you wish to use modem style communications. Choose tcpip to use telnet type communications.

Emulation

A drop down list box presents all available terminal emulation modules. You should select the terminal emulation which is best suited for the remote system you will be connecting to. See the section on Terminal Emulation in the Program documentation if you are not sure which one to choose.

Protocol

A drop down list box presents all available file transfer protocols. At this time only kermit is supported.

Auto Connect on Open

You may check this box if you would like Kermit to automatically connect to the remote system when this session is subsequently reopened (see below).

See also...

[Configuration Options / Terminal Emulation Settings](#)

[Configuration Options / Communications Settings](#)

[Configuration Options / Device Settings](#)

[Configuration Options / Protocol Settings](#)

[Configuration Options / Font Settings](#)

Configuration Options / Terminal Emulation Settings

The following configuration options control Kermit's terminal emulation settings:

Auto New Line

Normally, a carriage return control character will only move the cursor to the first position on the current line. If this option is selected, a carriage return character will also advance the cursor to the next line.

Local Echo

Normally, characters you type at the keyboard are sent only to the remote system and the remote system "echoes" them back. If the remote system does not echo typed characters, select this option to cause typed characters to be displayed on the terminal screen and sent to the remote system.

Auto Line Wrap

If this option is selected and too many characters are received to fit on the current line, the cursor will automatically advance to the first position of the next line when the last position of a line is used.

See also...

[Configuration Options / Session Settings](#)

[Configuration Options / Communications Settings](#)

[Configuration Options / Device Settings](#)

[Configuration Options / Protocol Settings](#)

[Configuration Options / Font Settings](#)

Configuration Options / Communications Settings

The Communications Settings Dialog will present different options depending on the type of connection that was chosen in the Session Settings.

Async

Baud Rate

Enter the baud rate that you wish to use to communicate with the remote system or the speed your modem requires. Although you are allowed to enter any value, only 300, 1200, 2400, 4800, 9600, 19200, 38400, and 57600 should be used. Additionally, Windows will not support the higher baud rates with some types of serial port hardware.

Word Length

Also known as data bits. Usually 8, but occasionally 7. The administrator of the system you are trying to connect with should be able to give you this information.

Parity

Normally none, but again the administrator of the remote system should be able to supply this information. Note that N=None, E=Even, O=Odd, M=Mark, S=Space.

Stop Bits

Number of stops bits required by the remote system. Almost always 1.

Handshake

Kermit for Windows supports two types of "flow control". Xon/Xoff uses control characters to prevent buffer overruns. RTS/CTS uses two specific lines (wires) on the serial port to achieve the same effect. Which option is appropriate depends on your modem configuration and remote system requirements.

Port

Kermit for Windows support COM ports 1 through 4 (if available). Although all 4 selections are always shown, your computer probably doesn't have all four. Be sure to choose a port that exists and is available and properly configured under Windows.

TCP/IP

Host

Enter the internet host name you wish to connect to (like microsoft.com) or the internet address to connect to (like 129.193.208.56).

Port

Enter the name or number of the TCP/IP service port to use. This entry is almost always telnet and should only be changed if you are sure you understand the use of non-standard ports.

NOTE: although it is called a "port", this is nothing like a "serial port". A TCP/IP port has no connector on the back of your PC, it is a logical port.

Terminal

Most Telnet host servers have the ability to read the terminal type of the users machine at connection time. To enable this feature, you must specify the name of the terminal type you wish to provide to the host. Kermit for Windows will provide this name to the host if it requests it.

NOTE: this entry does NOT determine the type of emulation that Kermit for Windows will perform (that is determined by the Emulation setting in the Session Configuration Dialog). This entry should correspond to a name that your host understands and will associate with the type of emulation you have already

chosen in the Session Configuration Dialog.

Send CR LF for End of Line

If you are attempting to talk to a non-standard telnet service, the host may require both a CR and LF control code at the end of every input line. If you don't understand this option, you probably don't want to use it.

Suppress Negotiation

The Telnet protocol requires that the client and host "negotiate" certain configuration settings at startup. In rare cases you may want to disable this negotiation.

See also...

[Configuration Options / Session Settings](#)

[Configuration Options / Terminal Emulation Settings](#)

[Configuration Options / Device Settings](#)

[Configuration Options / Protocol Settings](#)

[Configuration Options / Font Settings](#)

Configuration Options / Device Settings

The following configuration options control Kermit's device settings:

Connection

Type of connection desired. Dial and answer apply to modems. If no modem is in use, you should choose direct. If dial is chosen, a phone number to dial should be supplied.

Dial Init

The sequence of characters that should be send to the modem to initialize it for dial-out usage.. The default is usually OK.

Dial Cmd

The sequence of characters that should be sent to the modem to cause it to dial a phone number. The default is usually OK.

Answer Init

The sequence of characters that should be send to the modem to initialize it for answer usage. The default is usually OK.

Hangup

The sequence of characters that should be send to the modem to cause it to hang up the phone line. The default is usually OK.

Escape

The sequence of characters that should be send to the modem while online prior to sending command strings.

Connect

The message that the modem will send when a connection has been successfully established.

Redial

Number of times to redial a number that fails to connect followed by the number of seconds to wait between attempts.

Wait

Number of seconds to wait from the time a number is dialed to get a successful connection message.

See also...

[Configuration Options / Session Settings](#)

[Configuration Options / Terminal Emulation Settings](#)

[Configuration Options / Communications Settings](#)

[Configuration Options / Protocol Settings](#)

[Configuration Options / Font Settings](#)

Configuration Options / Protocol Settings

The following configuration options control Kermit's protocol settings:

Packet Size

Size of data packets in bytes desired for send and receive.

Timeout

Number of seconds to wait before assuming a lost packet for send and receive.

Retry Limit

Maximum number of retries before aborting protocol.

Block Check

Kermit provides three different error detection block check types. Block Check Type 1 is the least reliable and Block Check Type 3 is the most reliable.

Debug

Select debugging options as desired. Packet option will display packets sent/received. State option will show state of protocol engine at various points. Other is undefined.

Options

Check the appropriate boxes for advanced Kermit protocol functions. Attrib will allow file information attributes to be used. L. Pkts will allow packet sizes up to 1000 bytes. Windows will allow the sliding windows extension to be used (Super Kermit).

Packets...

This button invokes a nested dialog that allows special customization of various Kermit packet characters. It is not normally needed.

Options...

Not implemented in this version.

See also...

[Configuration Options / Session Settings](#)

[Configuration Options / Terminal Emulation Settings](#)

[Configuration Options / Communications Settings](#)

[Configuration Options / Device Settings](#)

[Configuration Options / Font Settings](#)

Configuration Options / Font Settings

The following configuration options control Kermit's font settings:

Font

Select from the list of available fonts. Only "fixed-pitch" fonts will be displayed. If a font called "Terminal" is available, it is recommended because it is specifically designed to work with communication programs.

Size

Point size for font. A larger point size means a larger font displayed on the window.

Sample

A sample display of the currently selected font.

Note: After changing the selected font, Kermit will attempt to resize the terminal window to provide a full 80 by 25 display. If the window can not be expanded enough to do this, scroll bars will be displayed.

When a new session is created, Kermit requests a default fixed-pitch font from Windows. In some cases (depending on your display driver and resolution), windows does a poor job of selecting a default font.

See also...

[Configuration Options / Session Settings](#)

[Configuration Options / Terminal Emulation Settings](#)

[Configuration Options / Communications Settings](#)

[Configuration Options / Device Settings](#)

[Configuration Options / Protocol Settings](#)

About Kermit

Kermit for Windows was written for use in the Microsoft® Windows™ operating environment.

Kermit is distributed in two separate formats: as a 16-bit application for standard Windows (version 3.0 and higher), and as a 32-bit application for Windows NT.

For further information about this product, please contact:

Wayne Warthen
CompuServe: 73457,2401
Internet: wwarthen@kaiwan.com



Kermit for Microsoft Windows Version 0.80

Copyright © 1990-95 Wayne Warthen

The information contained in this documentation is subject to revision and/or alteration without prior notice.

This documentation represents no obligation, expressed or implied, on the part of the author.

Microsoft® is a registered trademark and Windows is a trademark of Microsoft Corporation.

All other trademarks mentioned herein are trademarks or registered trademarks of their respective corporations, and are hereby acknowledged.

The author wishes to acknowledge and thank the following people for their contributions to this product:

Julie Bielsik: Artwork (Kermit Icon)
Matthew T. Smith: Help File

