PDP-11 KERMIT USER GUIDE

For RSX-11M, RSX-11M+, Micro-RSX, RSTS/E, P/OS, RT-11, and TSX+

Version 3.58

Brian Nelson

Computing Services
University of Toledo,
Toledo, Ohio

September 1987

1. PDP-11 Kermit

Author:	Brian Nelson, University of Toledo, Ohio
Documentation:	Brian Nelson
Language:	Macro-11
Version:	3.58
Date:	September, 1987
Systems Supported:	RSTS/E, RSX-11M/M+, P/OS, Micro-RSX, RT-11 and TSX+

Kermit-11 Capabilities At A Glance:

Local operation:	Yes
Remote operation:	Yes
Transfer text files:	Yes
Transfer binary files:	Yes
Wildcard send:	Yes
File transfer interruption:	Yes
Filename collision avoidance:	Yes
Can time out:	Yes
8th-bit prefixing:	Yes
Repeat count prefixing:	Yes
Alternate block checks:	Yes
LONG Packet protocol support:	Yes
Sliding Windows protocol support:	No
Terminal emulation:	Yes
Communication settings:	Yes
Transmit BREAK:	Yes (depends on system)
IBM mainframe communication:	Yes
Transaction logging:	Yes
Session logging:	Yes
Debug logging:	Yes
Packet logging:	Yes
Act as server:	Yes
Talk to server:	Yes
Advanced server functions:	Yes
Local file management:	Yes
Command/Init files:	Yes
File attributes packets:	Yes
Command macros:	No
Raw file transmit:	Yes

1.1. File Systems on the PDP-11

1.1.1. File Specifications

The general format of a file name is:

```
NODE::DEVICE:[DIRECTORY]NAME.TYPE;VERSION
```

'Node' refers to the DECNET node name, for example, `FUBAR::`, if applicable. 'Device', if present, refers to the physical device or logical name where the file resides.

For RSTS/E, 'device' can be a physical device, such as `DB0:` or `DU1:`, or it can be a user or system logical name which may include both a physical device name and a directory name. If the device name is a logical name, is it composed of 1 to 9 alphanumeric characters, including '`$`', as in `DISK$ONE:`, `LB:` and so on. For instance, the DCL system command

```
$ ASS/SYS DB1:[200,210] SRC$DIR
```

would associate both the device `DB1:` and directory `[200,210]` with `SRC$DIR:`. Explicitly given directories override directory names imbedded in a logical name. Names longer than nine characters are truncated by the executive.

In the case of RSX-11M/M+ and RT-11, the device name can be either a physical name, such as `DU0:`, or a logical name which will translate to a physical device name, such as `LB:`.

On RSTS/E and RSX-11M/M+, the [directory] is a UIC (user identification code) or PPN (project,programmer) number of the format [NNN,MMM]. All users are assigned a UIC (or PPN) when accounts are created, this is the number you give to LOGIN to log into the system. It is also your default UIC (or PPN). Micro-Rsx and P/OS may have directories in either UIC format or named directory format, such as `[1,2]` or `[KERMIT]`. For P/OS, the default directory is `[USERFILES]`. Directories are not used in RT-11.

The NAME field is the primary identifier for the file. The name can be one to nine characters for RSX-11M/M+ and P/OS, and one to six characters for RSTS/E, RT-11 and TSX+. The TYPE field is usually used to group files according to some convention. For example, `XXX.FTN` refers to a Fortran-77 source file, `FOO.C` to a 'C' source file, and `K11POS.TSK` refers to a task image.

The version field is applicable ONLY to RSX type systems. The default version is always the highest version number.

All systems mentioned support some sort of filename wildcarding, the flexibility of which varies by executive. All support the use of '`*`' to represent either a fully wildcarded NAME or TYPE. RSTS/E supports the use of '`?`' to match any single character, whereas the others use a '`%`' to match any single character. The RSTS/E Kermit server will translate '`%`' to '`?`' internally for the GET and REMOTE DIR commands (see the section on Kermit-11 server operation).

Examples of wildcarded filenames:

`*.B2S`	Match any file with a TYPE of B2S.
`K11%%%.MAC`	match any file starting with K11, followed by one to three characters, with a TYPE of MAC.
`K11???.MAC`	Same as above, but for RSTS/E only.
`XYZ.*;*`	All versions of files with a NAME of XYZ with any TYPE (RSX-11M/M+ and P/OS only).

1.1.2. File Formats (Binary and Text)

1.1.2.1. RT-11 and TSX+

RT-11 treats all files as a contiguous stream of characters. There is no information stored in the directory to tell the system (or program) that a file is readable text (source program, runoff document,...) or consists of binary data (executable program, object file, `.SYS` file,...). An application program like Kermit-11 needs to know what type of file to expect, thus the presence of the SET FILE TYPE command (discussed later). The only real convention is that text files are streams of seven bit data with each record terminated by a carriage return/line feed character sequence and that binary files normally follow a filename TYPE convention. The TYPE (`.SAV`, `.SYS`, ...) is what Kermit-11 will look at to decide if a file should be sent as a text or binary file.

1.1.2.2. RSTS/E, P/OS and RSX-11M/M+

These systems can provide for a large number of file attributes for each file by using either FCS11 (RSX-11M/M+) or RMS11 (all). Text files are normally considered to be either STREAM format (FB$STM) or VARIABLE with implied carriage control (FB$VAR and FB$CR). RSTS/E has historically defaulted to STREAM, whereas the RSX based systems use VARIABLE. Kermit-11 follows those defaults when creating files unless told to do so otherwise by the presence of attribute data. The conversion of the internal data representation to one that can be transmitted to another Kermit is transparent for these types of files. Both the file attributes and the filename TYPE are examined by Kermit-11 to determine if a file needs to be sent as a text file (default) or a binary file. Additionally, on RSTS/E Kermit checks the file protection code, as one of the bits in it is used to flag an executable file (bit 6).

In all cases, unless (at this time) Kermit-11 is talking to another Kermit-11, or if Kermit-11 can't tell if a file is consists of binary data, the command SET FILE TYPE FIXED must be used to force Kermit to either send or get a non-text file correctly. When Kermit-11 is running in binary mode, all data is read from (or written to) the file without any translation or internal record control information. Any attribute information in the file's directory entry is ignored and the data read (or written) in 512 byte unformatted blocks. Thus it is indeed possible to transfer files like task images and object libraries. Since Kermit-11 supports a subset of a protocol feature called 'attributes', two Kermit-11's connected together can also correctly transfer files other than simple text and unformatted binary files, such as RMS indexed or relative files.

1.1.3. Saving Files on the PDP-11 From Your Microcomputer

You can send textual files to Kermit-11 without any special considerations as Kermit-11 defaults to creating normal text files. However, if you are sending a binary file (perhaps an `.EXE`) from say, your Rainbow under MS-DOS, you would need to tell Kermit-11 to expect binary data. This is done with the Kermit-11 command SET FILE TYPE FIXED. This will force Kermit-11 to write the data out exactly as it comes, in 512 byte unformatted records. Sending the same file back to the Rainbow would not require any special action since the file, as it sits on the PDP-11, has the proper information in the directory entry to tell Kermit-11 that the file is binary. As a note, for RT-11 you would need to use a filetype that is normally considered 'binary' like `.SAV` or `.OBJ` (see above notes for RT-11).

Never try to do a wildcarded send with mixed binary and text files with the file type set to FIXED. The result could be unusable as not all systems store text data in the same internal format. For example, if Kermit-11 is forced into binary mode (via SET FIL TYP FIX) and is requested to send a file with implied carriage control (normal for RSX text files), it will actually send, for each line, two bytes representing the record length, followed by the data and then followed by a ASCII NUL to pad the record to an even length. That is not incorrect, rather, it is EXACTLY how the data was stored on disk.

In general, avoid sending anything other than unformatted binary files and text file to unlike systems. For example, requesting a RMS indexed file from the PDP-11 to be sent to a PC would case Kermit-11 to send it as a binary file, but the file attributes would be lost. Sending such a file back to the PDP-11 would result in an unusable file unless you could reconstruct the attribute information.

1.1.4. Program Operation

Kermit-11's prompt is normally "Kermit-11>". This can be changed if need be via the SET PROMPT command. Invoking Kermit-11 is very site dependent.

1.1.4.1. RSTS/E

If Kermit-11 has a ccl definition, it would likely be invoked as "KER" or "KERMIT". If not, try "RUN $KERMIT", as this is a likely place where Kermit-11 may have been put. Otherwise consult your local support staff.

1.1.4.2. RSX-11M/M+

If Kermit-11 has been installed, it most likely will have a task name of `...KER` which means that typing "KER" should get things running. If not, consult your local support staff.

1.1.4.3. RT-11/TSX+

On version 5 of RT-11, programs can be run simply by typing the filename. Thus, if there is a file `SY:KERMIT.SAV`, simply type "KERMIT". If this fails, contact your local support staff for assistance.

1.1.4.4. P/OS

Kermit-11 is generally run from DCL on P/OS. The program is invoked via the DCL RUN command, as in RUN K11POS or RUN KERMIT, depending on what the task image name is.

Note that for the case where Kermit is installed (for RSTS/E and RSX-11M/M+) that Kermit-11 can get command line arguments, as in:

```
$ KER SERV                    Kermit starts as a server.
> KER send fubar.txt          Kermit sends the file.
```

Otherwise, the program is run interactively from the Kermit-11> prompt:

```
$ KERMIT
Kermit-11 V3.54
Kermit-11>SET BLO 3           Changes checksum type.
Kermit-11>SER                 Enter Kermit server.
```

Note that whenever Kermit-11 starts up, it will always try to find a file called `KERMIT.INI` in your current directory. This file can contain any valid Kermit command, though the usual use of this is to place various Kermit-11 SET commands in it. If this file does NOT exist, it will try to find it in `LB:[1,2]KERMIT.INI` (excluding RT-11). In addition to the `.INI` file, commands may be placed in a file and then executed via the Kermit-11 TAKE (or @) command.

1.2. Local and Remote Operation

Kermit-11 by default assumes that all file transfers will occur over the terminal line that you are currently logged in on (`TI:`, `TT:`, `KB:`). This is known as REMOTE mode (the PDP-11 is the remote system). This would be the desired case if you are running Kermit on a microcomputer such as a Rainbow and are currently logged into the PDP-11 through the micro. However, if you wanted to dial out, say by an autodial modem, from the PDP-11 to another system, you need to tell Kermit-11 to use some other terminal line. This would be called LOCAL mode (the PDP-11 is the local system). The line can be altered with the SET LINE command (see section on SET and

CONNECT). A SET LINE command is done implicitly if Kermit-11 finds itself running on a PRO/350, under either P/OS, RT-11 or TSX+.

Since support of parity varies by both interface type (DL11 vs DZ11) and by operating system, Kermit-11 makes NO attempt to find out what the current parity of it's line is. Kermit-11 generates it's own parity which is set with the SET PARITY command.

There are a couple of things to point out regarding Kermit-11 and LOCAL mode (you did a SET LINE command):

- The system manager may have lines other than your own protected (or owned by the system). On RSTS/E lines are often made unaccessible unless your account possesses the needed privilege(s). On RSX-11M/M+, privilege is required to alter settings on any other terminal line. You may have to talk to your system manager to get access to an outgoing terminal line.

- Once connected to a modem through another line, a means must exist for the connection to be broken (if the host you are calling won't do it). Given that your line has full or partial modem control (DZV11, DZ11, DH11, DHU/V11) the RSX, RT-11/TSX+ and RSTS/E Kermits have a HANGUP (or DISCONNECT) command, which instructs the system to disconnect the modem. Unless this is done, you never get disconnected and could run up a tidy phone bill.

Kermit-11 has, as of v3.53, a rudimentary command line editor. You can recall previous commands with the UP-Arrow key, and exit the command with the LEFT and RIGHT arrow keys. The RUBOUT key, of course, deletes characters, while the Control-R key retypes the line. Control-E moves to the end of the line and Control-H moves to the start of the line.

1.3. Kermit-11 Commands

Kermit-11 has the following commands available:

@	Synonym for TAKE
BYE	Logout a remote server
CONNECT	Connect to a remote system
COPY	Local copy of a file(s)
CWD	Set new working directory
DELETE	Local delete of a file(s)
DIAL	Have a connected modem dial a number
DIRECT	Local directory display
DISCONNECT	Hangup a remote line
DISPLAY	Internal debugging
ERASE	Local delete of a file(s)
EXIT	Exit to system
FINISH	Stop a remote server without logging out
GET	Get a file(s) from a remote server
HANGUP	Hangup a remote line
HOST	Execute system command locally (where applicable)
LOCAL	Force interpretation of command to the local system
LOGFILE	Create a log file
QUIT	Same as EXIT
PRINT	Print a file locally (where applicable)
RECEIVE	Receive a file(s) from a remote kermit
REMOTE	Prefix for file management commands to a server
RENAME	Local rename of filename(s)
SEND	Send a file(s) to a remote Kermit
SERVER	start a Kermit server
SET	Change Kermit parameters
SHOW	Display Kermit parameters
TAKE	Execute indirect command file

TYPE	Local display of file on terminal
WHO	Local display of logged in users (RSTS/E only)

1.4. Commands for File Transfer

Kermit-11 includes the standard repertoire of Kermit file transfer commands, including SEND, RECEIVE, and GET.

The SEND Command

Syntax: `SEND` *filespec*

The SEND command causes a file or file group to be sent from the PDP-11 to the other system. If filespec contains wildcard characters then all matching files will be sent, in alphabetical order (according to the ASCII collating sequence) by name. If filespec does not contain any wildcard characters, then the single file specified by filespec will be sent.

SEND Command General Operation

:

Files will be sent with their PDP-11 file name and type (for instance FOO.BAR). Each file will be sent according to the record type and attributes recorded in its file descriptor. Kermit-11 attempts to translate all formats of text file to a format usable on any system. Note that there is no need to set the FILE TYPE parameter for sending files, since Kermit-11 always uses the information from the file directory entry and the filetype (extension) to determine how to send the file.

If communication line parity is being used (see SET PARITY), Kermit-11 will request that the other Kermit use a special kind of prefix notation for binary files. This is an advanced feature, and not all Kermits have it; if the other Kermit does not agree to use this feature, binary files cannot be sent correctly. This includes executable programs (like `.EXE` files, CP/M `.COM` files), relocatable object modules (`.OBJ` files), as well as any text file containing characters with the eighth bit on.

Kermit-11 will also ask the other Kermit whether it can handle a special prefix encoding for repeated characters. If it can, then files with long strings of repeated characters will be transmitted very efficiently. Columnar data, highly indented text, and binary files are the major beneficiaries of this technique.

If you’re running Kermit-11 locally, for instance dialing out from a PDP-11 to another system using an autodialer, you should have already run Kermit on the remote system and issued either a RECEIVE or a SERVER command. Once you give Kermit-11 the SEND command, the name of each file will be displayed on your screen as the transfer begins. As the transfer continues, you will get a small display of the packet count along with the number of packets rejected. See the SET TERMINAL and SET UPDATE commands for more information. You may also type Control-X or Control-Z to interrupt the current file or file group. Control-E will also abort the transfer by sending an ’error’ packet to the other Kermit.

The RECEIVE command

Syntax: `RECEIVE [`*filespec*`]`

The RECEIVE command tells Kermit-11 to receive a file or file group from the other system. The name is taken from the incoming file header. If an incoming file has the same name as an existing file, Kermit-11 will by default create a new file. On RT-11 and RSTS/E, the old file will be deleted by the executive. On RSX-11M/M+ and P/OS, a new file with a higher version number will be created. To avoid files being superceded, see the SET FILE [NO]SUPERCEDE command.

Incoming files will all be stored with the prevailing file type, ASCII by default, which is appropriate for text files. If you are asking Kermit-11 to receive binary files from a microcomputer or other 8-bit system, you must first type SET FILE TYPE FIXED. Otherwise, an error may occur when receiving the file. Please note that this does NOT apply to two Kermit-11 programs connected to each other. In that case the sending Kermit-11 will tell the receiving Kermit-11 to switch to binary mode if need be.

If parity is being used on the communications line, then 8th-bit prefixing will be requested. If the other side cannot do this, binary files cannot be transferred correctly.

If you are running Kermit-11 locally, you should already have issued a SEND command to the remote Kermit, and then escaped back to Kermit-11. As files arrive, their names will be displayed on your screen.

If a file arrives that you don't really want, you can attempt to cancel it by typing Control-X; this sends a cancellation request to the remote Kermit. If the remote Kermit understands this request (not all implementations of Kermit support this feature), it will comply; otherwise it will continue to send. If a file group is being sent, you can request the entire group be cancelled by typing Control-Z.

Normally, one runs the remote Kermit as a SERVER, thus the RECEIVE command is never used, rather, the GET command, described next, is used.

The GET Command

Syntax: `GET [`*remote-filespec*`]`

The GET command requests a remote Kermit server to send the file or file group specified by *remote-filespec*. This command can be used only when Kermit-11 is local, with a Kermit server on the other end of the line specified by SET LINE. This means that you must have CONNECTed to the other system, logged in, run Kermit there, issued the SERVER command, and escaped back to the PDP-11.

The remote filespec is any string that can be a legal file specification for the remote system; it is not parsed or validated locally. Any leading spaces before the remote filespec are stripped, and lower case characters are raised to upper case.

As files arrive, their names will be displayed on your screen. As in the RECEIVE command, Control-X (`^X`) to request that the current incoming file be ancelled, `^Z` to request that the entire incoming batch be cancelled.

If the remote Kermit is not capable of server functions, then you will probably get an error message back from it like "Illegal packet type". In this case, you must connect to the other Kermit, give a SEND command, escape back, and give a RECEIVE command.

1.4.1. Server Operation

The SERVER command puts a remote Kermit-11 in "server mode", so that it receives all further commands in packets from the local Kermit. The Kermit-11 server is capable (as of this writing) of executing the following remote server commands: SEND, GET, FINISH, BYE, REMOTE DIRECTORY, REMOTE CWD, REMOTE SPACE, REMOTE DELETE, REMOTE TYPE, REMOTE HELP, REMOTE COPY, REMOTE RENAME, REMOTE WHO, REMOTE LOGIN and REMOTE HOST.

Any nonstandard parameters should be selected with SET commands before putting Kermit-11 into server mode, in particular the file type. The Kermit-11 server can send all files in the correct manner automatically. As noted before, if a Kermit-11 is talking to another Kermit-11, they will negotiate any 'binary' parameters automatically. However, if this is NOT the case and you need to ask Kermit-11 to receive binary files you must issue the SET FILE TYPE FIX command before putting it into server mode, and then you must only send binary files. You cannot send a mixture of text files and 8-bit binary files to a Kermit-11 server unless the files are not for use on the PDP-11.

1.4.2. Commands for Servers

When running in local mode, Kermit-11 allows you to give a wide range of commands to a remote Kermit server, with no guarantee the that the remote server can process them, since they are all optional features of the protocol. Commands for servers include the standard SEND, GET, BYE, FINISH commands, as well as the REMOTE command.

The BYE Command

The BYE command tells a remote server to log out of the remote system. In addition, some remote systems will also disconnect the line for you. If this is not the case, the DISCONNECT command will (depending on your interface) cause the line to be dropped. See DISCONNECT.

The FINISH Command

The FINISH command tells the remote Kermit server to exit without logging out of the remote system. You can then CONNECT back to the Server operation system.

The REMOTE Command

Send the specified command to the remote server. If the server does not understand the command (all of these commands are optional features of the Kermit protocol), it will reply with a message like "Unknown Kermit server command". If does understand, it will send the results back, and they will be displayed on the screen. The REMOTE commands are:

REMOTE COPY *filespec newfilespec*
: Copy file. The server is asked to make a copy of the specified file. Both filespecs must be in the correct format for the remote system. Kermit-11 does not parse or validate the file specifications. Any leading spaces will be stripped and lower case characters converted to upper case. Note that this command simply provides for copying a file within the server's system - it does not cause a file to be transferred.

REMOTE CWD *directory*
: Change Working Directory. If no directory name is provided, the server will change to the default or home directory. Kermit-11 currently does not ask for a password.

REMOTE DELETE *filespec*
: Delete the specified file or files. The names of the files that are deleted will appear on your screen.

REMOTE DIRECTORY [*filespec*]
: The names of the files that match the given file specification will be displayed on your screen, perhaps along with size and date information for each file. If no file specification is given, all files from the current directory will be listed.

REMOTE HELP
: The remote server will send back a list of server commands that it can execute.

REMOTE HOST *command*
: Pass the given command to the server's host command processor, and display the resulting output on your screen. Not all Kermit servers can do this function. In the case of Kermit-11, only the RSTS/E Kermit-11 server can execute the REMOTE HOST command.

REMOTE LOGIN *user password*
: Ask a remote server to log into a different account or username. The support for this command is rarely implemented as many systems layer login/logout support over the executive. A Kermit-11 server can only support this on RSTS/E, and at that only for version 9.0 or later. Of the various DEC PDP-11 operating systems, only RSTS/E has the support for logging in and out built into the executive and accessible with directives.

REMOTE RENAME *oldfile newfile*
: Change the name on the specified file (or files). Both file specifications must be valid for the server's system.

REMOTE SPACE
: Display information about disk usage in the current directory.

REMOTE TYPE *filespec*
: Display the contents of the specified file on your screen.

REMOTE WHO
: Display current status of user's logged in.

1.5. Commands for Local File Management

These commands provide some local file management capability without having to leave the Kermit-11 program. These commands are very similar to the REMOTE commands in function and syntax. They are all executed locally, and are available when Kermit-11 is either local or remote. The arguments to these commands are the same as the arguments expected from the user Kermit when Kermit-11 is processing a command in server mode. Additionally, these commands can be prefixed by the LOCAL keyword.

COPY *filespec newfilespec*
CWD *directory*
DELETE *filespec*
DIRECTORY [*filespec*]
HELP
HOST *command*
RENAME *oldfile newfile*
SPACE
TYPE *filespec*
WHO

1.5.1. The CONNECT Command

The CONNECT command will allow you to connect in as a terminal over the line that was specified by the SET LINE command. (Using the CONNECT command before using the SET LINE command will result in an error message.) The terminal line must be one which is accessible to the user.

The distributed RSX-11M/M+ task has been built with the `/PR:0` switch to enable the task to change other terminal settings. Additionally, for RSX-11M/M+, the MCR command `SET /SLAVE=TT`*nn*`:` should be done before entering Kermit-11.

If you are running `K11POS.TSK` on a PRO/350, Kermit will set the line to `XK0:` and the speed to 9600 by default.

Please note that Kermit-11 CAN NOT change the speed of a DL11 type interface, nor can it change the speed of a PDT-150 modem port (use SPEED.SAV).

The following is an example of using a Racal-Vadic VA212 autodialing modem to log into a remote TOPS-20 system. There is one point at which there is no echoing of the user input, this is following the typing of the local 'escape sequence', which by default is Control-\ followed by a 'c'. The control-backslash informs the terminal emulator that the next character is a command. In this case, the command was 'C', which means to return to the local PDP-11 system. Control-\ followed by `?` would print a help message. All the commands prior to the DIAL command were contained in the INI file, `KERMIT.INI`.

```
$ kermit
Kermit-11 V3.46 Last edit: 21-Feb-1986
Kermit-11>set modem vadic
Kermit-11>set pho num cu 9K12121234567
Kermit-11>set logfile 20.log
Kermit-11>set deb console
Kermit-11>set lin tt58:
Link: TT58: Speed: 9600, DTR not present
Kermit-11>set dtr
Kermit-11>set spe 1200
Kermit-11>dial cu
Using: 9K12121234567
Connection established, type CONNECT to access remote
Kermit-11>connect

enter class 4
class 004 start

CU20B
@log xx.abcdef
 CU20B, TOPS-20 Monitor 5.1(5101)-2
 Job 28, TTY32, 2-Apr-84 4:15:24PM
 Previous login was 2-Apr-84 4:10:16PM
        .
        .
@logout
[Confirm]
Logged out Job 28, User XX.ABCDEF , TTY 32,
  at  2-Apr-84 16:19:34,  Used 0:00:11 in 0:04:10

Kermit-11>disc
KERMIT link TT58: disconnected
Kermit-11>exit

$ logout
```

1.6. The SET Command

Syntax: `SET` *parameter keyword*

The SET command is used to set various parameters in Kermit. The format of the SET command is:

SET ATTRIBUTES

Syntax: `SET ATTRIBUTES {ON, OFF}`

Part of the Kermit protocol is the support of file attributes. Connected Kermits that support this can send information to each other about file size, time/date of creation, RMS file headers and other useful things. Due to potential problems with incompatible implementations this feature can be disabled. In this case, the sending Kermit-11 will never try to send file attributes, even though the receiver may have indicated that it supports this.

SET BAUD

This is the same as SET SPEED. See HELP SET SPEED

SET BINARY-TYPE

Kermit-11 has a default list of filetypes that are scanned to decide if a file should be sent in binary mode in addition to checking file attributes for RSX, P/OS and RSTS/E. The user can, however, override this list with the this command. The default list is fairly inclusive, with types such as `.SAV` and `.TSK` forcing Kermit-11 into binary transmission. See HELP SET FIL for the default list. Examples:

```
Kermit-11> set binary-type .sav
Kermit-11> set bin .exe
```

SET BLOCK-CHECK

Syntax: `SET BLOCK_CHECK {1, 2, 3}`

The SET BLOCKCHECK command is used to determine the block check sequence which will be used during transmission. The block check sequence is used to detect transmission errors. There are three types of block check available. These are the single character checksum (default), the two character checksum, and the three character CRC (cyclic redundancy check). This command does not ensure that the desired type of block check will be used, since both Kermit's involved in the transfer must agree on the block check type. Kermit-11 will request that the type of block check set by this command be used for a transfer. If the other Kermit has also had the same block check type requested, then the desired block check type will be used. Otherwise, the single character checksum will be used. The command should be given to BOTH Kermits since Kermit-11, when in server mode, has no say about what kind of checksum it wants to use. (See Kermit protocol manual for more information.)

SET CONSOLE

Syntax: `SET CONSOLE {7, 8}`

The SET CONSOLE command is used under P/OS to control the passing of 8 bit data to the terminal during the connect command. If you are getting multinational characters being printed, this is a very useful thing to set. The default is SET CON 7.

SET DEBUG

Syntax: `SET DEBUG {ALL, CONSOLE, CONNECT, FILE, PACKET, STATE}`

The SET DEBUG command is used to specify the type and level of debugging to a disk file . This disk file must have been created by the SET LOGFILE command.

SET DEBUG ALL

SET DEBUG ALL will turn on logging for CONSOLE,CONNECT,FILE,PACKET and STATE to the disk file specified by SET LOGFILE. This command is the same as SET DEBUG ON. The command format is:

SET DEBUG CONSOLE

SET DEBUG CONSOLE will turn on logging for all i/o during a remote connect to the disk file specified by SET LOGFILE. This command is the same as SET DEBUG CONNECT.

SET DEBUG CONNECT

SET DEBUG CONNECT will turn on logging for all i/o during a remote connect to the disk file specified by SET LOGFILE. This command is the same as SET DEBUG CONSOLE.

SET DEBUG FILE

SET DEBUG FILE will log all file 'opens' and 'creates' to the file specified by SET LOGFILE.

SET DEBUG HELP

SET DEBUG HELP gives the user a list of all qualifiers which can be used with SET DEBUG.

SET DEBUG NONE

SET DEBUG NONE 'turns off' all debugging. This is the same as the SET DEBUG OFF command.

SET DEBUG OFF

SET DEBUG OFF 'turns off' all debugging. This is the same as the SET DEBUG NONE command.

SET DEBUG ON

SET DEBUG ON will 'turn on' logging for CONSOLE,CONNECT,FILE,PACKET and STATE to the disk file specified by SET LOGFILE. This command is the same as SET DEBUG ALL.

SET DEBUG PACKET

SET DEBUG PACKET will 'turn on' logging of all receive and transmit packets to the disk file specified by SET LOGFILE.

SET DEBUG STATE

SET DEBUG STATE will turn on logging of all internal Kermit-11 state transitions.

SET DELAY

Syntax: `SET DELAY` *seconds*

The DELAY parameter is the number of seconds to wait before sending data after a SEND command is given. This is used when Kermit-11 is running in remote mode to allow the user time to escape back to the other Kermit and give a RECEIVE command.

SET DEFAULT

Syntax: `SET DEFAULT` *device*

The DEFAULT parameter allows you to specify a device and UIC (or PPN) for all subsequent file opens (for SENDING) and file creates (for RECEIVING). It is disabled by typing SET HOME. Example:

```
Kermit-11>set default db2:[200,201]
```

This is quite useful for Kermit-11 running on a DECNET link, as you can set the default for file operations to include node names and passwords as in:

```
Kermit-11>set def orion::sys$system:[fubar]
```

SET DIAL

Kermit-11 has knowledge built in to it of a number of the more common ’smart’ autodial modems. To find out if your modem is directly supported try the command SET MODEM ?. If your modem is not in this list then you need the SET DIAL command to generate the data base used by Kermit to control the modem. Kermit uses this information to implement the DIAL command. A command such as DIAL can only be done when Kermit knows both how to format commands to the modem, and what kind of text the modem will send back to it in response. As an example, the VADIC VA212PA modem is awakened from an idle state by the character sequence 05 015 (in octal), which is a Control-E followed by a carriage return. In response to this two-character string, the modem responds with:

```
HELLO: I’M READY
*
```

Thus Kermit has to know that when it sends the wakeup sequence it needs to wait for the asterisk to be sent back by the modem. At this point Kermit will know that the modem is in a state awaiting further commands, such as that to dial a phone number.

It is not possible for Kermit to have knowledge of all makes of modems. Instead Kermit supports a command, SET MODEM USER_DEFINED, which then allows you to use the SET DIAL command to inform Kermit how the modem works. Once Kermit knows how to control the modem, you can use the DIAL command to initiate a call from Kermit.

The SET DIAL commands are:

SET DIAL WAKEUP	Define the wakeup string
SET DIAL PROMPT	Define the prompt the modem uses
SET DIAL INITIATE	Define a string to start dialing
SET DIAL CONFIRM	Define the string to confirm number
SET DIAL FORMAT	Define the number formatting string
SET DIAL SUCCESS	Define string(s) for call complete
SET DIAL INFO	Define string(s) for informative text
SET DIAL FAILURE	Define string(s) for call failure
SET DIAL CONFIRM	Define string for number confirmation
SET DIAL WAKE_RATE	Set pause time between wakeup characters
SET DIAL DIAL_RATE	Set pause time between number digits
SET DIAL DIAL_PAUSE	Define string for dial tone pause

Suppose we had to tell Kermit about the Racal Vadic VA212PA modem (though in reality Kermit already knows about that kind). In checking the owners manual for it, we find that:

- To wake the modem up, we type a control E followed by a carriage return.
- To dial a number, we type the letter D followed by a carriage return. At this point, the modem prints a NUMBER? prompt, we then type the desired number in. It reprints the number and then waits for a carriage return from us to confirm that its really the correct phone number.
- When it completes dialing, it will print ’ON LINE’ or ’ONLINE’ for a successful call, otherwise it may display on the terminal ’BUSY’, ’FAILED CALL’, ’NO DIAL’, ’VOICE’ or ’TIME OUT’. While it is waiting for its call to be answered, it may print the line ’RINGING’ several times in order to tell you that it is working on it.

The Kermit commands required would be:

```
Kermit-11>SET MODEM USER_DEFINED
Kermit-11>SET DIAL WAKEUP \05\015
Kermit-11>SET DIAL PROMPT *
Kermit-11>SET DIAL INITIATE D\015
Kermit-11>SET DIAL FORMAT %P%S\015
```

```
Kermit-11>SET DIAL CONFIRM \015
Kermit-11>SET DIAL SUCCESS ONLINE
Kermit-11>SET DIAL SUCCESS ON LINE
Kermit-11>SET DIAL INFO RINGING
Kermit-11>SET DIAL FAILURE BUSY
Kermit-11>SET DIAL FAILURE FAILED CALL
Kermit-11>SET DIAL FAILURE NO DIAL
Kermit-11>SET DIAL FAILURE VOICE
Kermit-11>SET DIAL FAILURE TIME OUT
Kermit-11>SET DIAL DIAL_PAUSE 9K
Kermit-11>DIAL 14195551212
```

The notation "`\05\015`" indicates the Control E followed by a carriage return; 05 is octal for control E, 015 is octal for carriage return. An alternate notation for octal numbers can be used by placing the value inside of inequality characters, as in SET DIAL WAKE <05><015> though the former is preferred.

The notation "`%P%S\015`" indicates to Kermit that the phone number from the dial command is to be followed by a carriage return; the `%S` is simply a placeholder for the phone number. The presence of the `%P` is to indicate where to insert the dial pause string, in this case we need to dial 9 and wait for a second dial tone. The "K" is the Racal Vadic code to get the modem to pause. If you are dialing on a direct line, the DIAL_PAUSE command is unneeded. If for any reason you need to pass a "\" or "<" to your modem, simply prefix the character with another "\", as in "\\".

Many modems require only the WAKEUP, PROMPT, FORMAT and result strings. The Digital DF112 is an example of this; its definition would look like:

```
Kermit-11>SET MODEM USER_DEFINED
Kermit-11>SET DIAL WAKEUP \02
Kermit-11>SET DIAL PROMPT READY
Kermit-11>SET DIAL FORMAT %S#
Kermit-11>SET DIAL SUCCESS ATTACHED
Kermit-11>SET DIAL FAILURE BUSY
Kermit-11>SET DIAL FAILURE DISCONNECTED
Kermit-11>SET DIAL FAILURE ERROR
Kermit-11>SET DIAL FAILURE NO ANSWER
```

Some modems may be unable to accept data at the line speed; in this case we would need to use the SET DIAL WAKE_RATE and SET DIAL DIAL_RATE. These two commands accept a delay time in milliseconds; the actual delay will not be precise as the PDP-11 line clock interrupts sixty times per second. Furthermore, on RSTS/E the finest granularity for timing is one second; thus setting delays would result in delays of one second increments.

In general, not all of the result fields need be specified except for the call completed strings; Kermit will time out after a while if it can't match a response with any definitions.

Further information can be found in the sections on SET MODEM, DIAL, REDIAL and SET PHONE.

SET DTR

The SET DTR command is very similar to the DISCONNECT (or HANGUP) command. SET DTR, where supported, raises DTR for a predetermined amount of time, whereas the DISCONNECT (or HANGUP) command drops DTR. The SET DTR is only functional on RSTS/E, which by default keeps DTR low until either RING INDICATOR or CARRIER DETECT goes high. This is opposite of the behavior on RT11 and RSX11M/M+, both of which normally assert DTR. The SET DTR command raises DTR for at least 30 seconds (depending on the version of RSTS/E) and is useful for making connections to front end switches (such as MICOM and GANDALF). On RT11, SET DTR is identical to the HANGUP command; it simply drops DTR for two seconds. In this case (RT11 and TSX+) this command is only supported on RT11 5.2 and TSX+ 6.0 with the XL/XC and CL drivers, respectively. This command is a no-op on RSX11M/M+ and P/OS. For further information on modem support, see the later section regarding such.

SET DUPLEX

Syntax: `SET DUPLEX {FULL, HALF}`

The DUPLEX parameter controls whether an outgoing link (set via the SET LINE command) is a full duplex link (the default) or a half duplex link. All it does for half duplex is to cause all characters typed after the CONNECT command to be echoed locally.

SET END-OF-LINE

Syntax: `SET END-OF-LINE <octal ASCII value>`

The END-OF-LINE parameter sets the ASCII character which will be used as a line terminator for all packets SENT to the other KERMIT. This is normally not needed for most versions of KERMIT.

SET ESCAPE

Syntax `SET ESCAPE` <octal ASCII value>)

This command will set the escape character for the CONNECT processing. The command will take the octal value of the character to use as the escape character. This is the character which is used to "escape" back to Kermit-11 after using the CONNECT command. It defaults to control (octal 34). It is usually a good idea to set this character to something which is not used (or at least not used very much) on the system being to which Kermit-11 is CONNECTing.

SET FILE

Syntax: `SET FILE {NOSUPERCEDE, SUPERCEDE, TYPE` *file-type*`}`

The SET FILE command allows you to set various file related parameters.

SET FILE TYPE ASCII

File type ASCII is for text files. SET FILE TYPE TEXT is the same.

SET FILE TYPE AUTO

Kermit-11 will normally try to decide if a file must be sent in binary mode based on the file attributes and filetype. If, for instance, the directory entry for `FUBAR.TXT` showed it to be RMS (or FCS) fixed length records, Kermit-11 will switch to binary mode and send it verbatim. If the receiving Kermit is Kermit-11, then the sending Kermit will send attribute data over also. The file types shown in Table 1-1 also will normally be sent as binary files unless you use the SET FILE TYPE NOAUTO command.

SET FILE TYPE BINARY

File type BINARY is for non-text files. Note that binary files which are generated on a PDP-11 system cannot be transferred to another (non PDP-11) system without losing file attributes. This means that (for example), an RSM11 indexed file cannot be transmitted with Kermit-11 at this time. You can not have parity set to anything but NONE

`*.TSK`	RSX, IAS, and RSTS tasks
`*.SAV`	RT11 and RSTS save images
`*.OBJ`	Compiler and macro-11 output
`*.STB`	TKB and LINK symbol tables
`*.CRF`	TKB and LINK cross reference files
`*.TSD`	'Time shared DIBOL' for RT11
`*.BAC`	RSTS Basic-plus 'compiled' files
`*.OLB`	RSX, IAS, and RSTS object libraries
`*.MLB`	RSX, IAS, and RSTS macro libraries
`*.RTS`	RSTS/E run time systems
`*.EXE`	VMS executable

Table 1-1: Kermit-11 File Types

to use binary file transfer (see HELP SET PARITY) unless the other Kermit can process eight bit quoting. Two Kermit-11's connected to each other will use binary transmission automatically via the Kermit attribute packets, preserving file attributes where it makes sense (i.e. RSTS/E and RSX only).

SET FILE TYPE DECMULTINATIONAL

PDP-11 Kermit normally strips the high bit of every character on both transmission and reception of files (unless the SET FILE TYPE FIXED command was given). The SET FIL DEC command will cause Kermit-11 to leave all data intact but still obey the host file system when reading or writing files. In other words, Kermit will write sequential implied carriage control files with eight bit data if this command is used.

SET FILE TYPE FIXED

This is the same as SET FILE TYPE BINARY.

SET FILE TYPE NOAUTO

SET FILE NOAUTO disables Kermit-11 from trying to base binary transmission mode on file attributes or filetype.

SET FILE SUPERCEDE

Syntax: `SET FILE {SUPERCEDE, NOSUPERCEDE}`

SET FILE [NO]SUPERCEDE allows Kermit-11 to accept or reject files received (from either the RECEIVE or GET commands) on a per file basis. The default is SUPERCEDE. By doing SET FILE NOSUPERCEDE Kermit-11 will always check to see if the file to be created is already there (independent of version number) and reject it to the sending server if it exists. This presumes that the Kermit sending the file understands the protocol to reject one file of a (possibly) wildcarded group of files. The main use of this is to resume getting a group of files, as in `GET KER:K11*.*` or `GET KER:MS????.*` having lost the connection after transferring some of the files. If this is set, then any files already transferred will not be transferred again.

SET HOME

SET HOME resets the default device and UIC (or PPN) to nothing, ie, all file opens and creates use your default disk (`SY:`) and your UIC (or PPN).

SET IBM-MODE

Syntax: `SET IBM {ON, OFF}`

The SET IBM ON (or OFF) will instruct Kermit-11 to wait for an XON following each packet sent to an IBM host in linemode. Since the default for IBM mode may not always be appropriate for your IBM compatible system, you can always use the SET HANDSHAKE XON and SET DUPLEX HALF to avoid the parity setting implied by using IBM mode.

SET LINE

Syntax: `SET LINE` *device-designator*

The SET LINE command sets the terminal name up for use with the connect command. To use this you must have access to that device. On many systems terminal lines other than your own are protected from access, and may require special procedures to access them. The form of the device name is TTnnn:, where 'nnn' is a decimal number for RSTS and an octal number for RSX-11M/M+. For RT-11, the device name is simply the MT unit number shown by the SHO TER command, as in '5' for DZ11 unit 0 line 4. If the system is running RT-11 version 5 you can do a SET LIN XL:. At worst case, Kermit-11 can use the console port on RT-11. For more information see the notes later on for RT-11 If you are running `K11POS.TSK` for P/OS on the PRO/350, Kermit-11 will set the line to `XK0:` and the speed to 9600 baud when Kermit starts. To override the line or speed, set HELP SET LINE and HELP SET SPEED. Examples:

```
Kermit-11>SET LINE TT55:        (for RSTS and RSX-11M/M+)
Kermit-11>SET LINE 5            (for RT-11 and MT service)
Kermit-11>SET LINE XK0:         (for P/OS, done implicitly)
Kermit-11>SET LINE XL:          (for RT-11 and XL handler)
```

See HELP CONNECT, HELP SET DUPLEX and HELP SET SPEED for more information. Also, for TSX+, see notes regarding TSX later in these notes. The RT-11 XL handler has notes later on also.

SET LOGFILE

Syntax: `SET LOGFILE` *filespec*

The SET LOGFILE command creates a debug dump file for you. It must be used BEFORE any SET DEBUG commands can be used. See HELP DEBUG for further information about debugging modes.

SET MODEM

The SET MODEM command defines the type of MODEM use for dialing out on the line set with the SET LINE command, or, in the case of the PRO/350, the XC or XK port. There are only a few modems defined at this time, they are:

VADIC	Generic RACAL-VADIC autodial
VA212PA	Stand alone VADIC VA212
VA212PAR	Rack mounted VADIC VA212
VA4224	Rack mounted VADIC VA4224 .v22bis
HAYES	Hayes smartmodem
DF100	DEC DF112
DF200	DEC DF224
DF03	DEC DF03
MICROCOM	

The DIAL command is then used after the SET MODEM command. For example, on a PRO/350 running P/OS:

```
Kermit-11>set prompt PRO>
PRO>set modem va212pa
PRO>dial 5374411
Modem in command mode
Modem dialing
Connection made, type CONNECT to access remote
PRO>con
Enter class ? VX785A
Class start
Username: BRIAN
Password: _____
```

...*and so on*

SET PACKET-LENGTH

Syntax: `SET PACKET-LENGTH` *length*

You can alter the default transmitted packet length with the SET PACKET-LENGTH command. This should not normally be needed unless the line is very noisy, at which time you should probably give up anyway.

SET PARITY

Syntax: `SET PARITY {EVEN, ODD, MARK, NONE, SPACE}`

This is used with the SET LINE and CONNECT commands to specify the type of parity for the remote link. It defaults to NONE and can be any of ODD, EVEN, MARK or SPACE.

All parity generation is done via software, no special hardware is used. The use of software parity generation is restricted to 8 bit links only. The character format, if parity is set to anything but NONE, will be 7 bits of data followed with high bit set or cleared to indicate the parity. If you set parity to anything but NONE (the default), Kermit-11 will be forced to request 8bit prefixing from the other Kermit-11, which is a method by which Kermit can ’prefix’ eight bit characters with a shift code. You MUST use parity (even if MARK or SPACE) when using Kermit-11 with the IBM CMS Series/1 or 7171 3270 emulator, or in linemode through a 3705 front end.

SET PAUSE

Syntax: `SET PAUSE` *seconds*

PAUSE tells Kermit to wait the specified number of seconds between each packet being sent to the other Kermit. This may be useful under situations of heavy system load. This may be automatically computer by Kermit-11 in a future release as a function of line speed.

SET PHONE

Syntax: `SET PHONE {NUMBER, TONE, PULSE, BLIND}`

The SET PHONE NUMBER command allows you to associate a phone number with a symbolic name for later use with the DIAL command. These definitions could be placed in your `KERMIT.INI` file, and then referenced later. Example:

```
Kermit-11>set pho num work 5374411
Kermit-11>set pho num market 16174671234
Kermit-11>dial work
```

The other two SET PHONE options, SET PHONE [TONE][PULSE] and SET PHONE BLIND are not useful unless the appropiate dial formatting string and character sequences for selecting PULSE or TONE, and BLIND dialing are present in the modem definition macros in `K11DIA.MAC`. The format effector for TONE/PULSE is `%M` and the effector for BLIND is `%B`. Currently (in 3.54) only the VA4224 has entries for these options.

SET POS

Syntax: `SET POS {DTE, NODTE}`

The SET POS command allows options SPECIFIC to P/OS to be altered. The most useful option is the SET POS [NO]DTE command. This allows Kermit-11 to use PRO/Communications version 2 for terminal emulation, if this product has been installed on the PRO/350. Of course, if this option is chosen, control is returned to the PRO with the EXIT key (F10) rather than with Control \C.

SET PROMPT

Syntax: `SET PROMPT` *prompt*

The SET PROMPT command is useful if you are using two Kermit-11's to talk to each other. By using the SET PROMPT command, you can change the prompt from 'Kermit-11>' on either (or both) Kermit to something that would indicate which system you are currently connected to. Examples:

```
Kermit-11>set prompt Kermit-11/1170>
Kermit-11>set prompt Fubar>
Kermit-11>set prompt ProKermit-11>
```

SET RECEIVE

Currently the SET RECEIVE and SET SEND basically work the same in that they only alter the END-OF-LINE character and the START-OF-PACKET value, as in:

```
Kermit-11>set rec start 2
Kermit-11>set rec end 12
```

The command SET RECEIVE PACKET-LENGTH command is discussed below.

SET RECEIVE END-OF-LINE

This instructs Kermit-11 to expect something other than the default carriage return (octal 15) at the end of a packet. Kermit-11 will ignore packet terminators. The SET SEND END command is of more use in conditioning outgoing packets.

SET RECEIVE START-OF-PACKET

The normal Kermit packet prefix is Control-A (ASCII 1); this command changes the prefix Kermit-11 expects on incoming packets. The only reasons this should ever be changed would be: Some piece of equipment somewhere between the two Kermit programs will not pass through a Control-A; or, some piece of of equipment similarly placed is echoing its input. In the latter case, the recipient of such an echo can change the packet prefix for outbound packets to be different from that of arriving packets so that the echoed packets will be ignored. The opposite Kermit must also be told to change the prefix for its inbound packets and the prefix it uses on outgoing packets.

SET RECEIVE PACKET-LENGTH

Syntax: `SET RECEIVE PACKET-LENGTH` *length*

This command has two functions. The first, and normal one, is to reduce incoming packet lengths in the event that normal sized Kermit packets can not be passed through the communications circuit. There could be, perhaps, some 'black box' somewhere in the link that has a very small buffer size; this command could be used to reduce the size that the SENDING Kermit will use.

The other use is to enable a protocol extension to Kermit called 'Long Packets'. The actual protocol is documented elsewhere, let's just say that this is a way for two Kermit's to use packet sizes far greater than the normal ('Classic') packet size if 90 characters or so. The main use of this feature is in file transfer over links that introduce considerable delay, it is not uncommon for packets to incur an one to two second delay. The net result is a VERY slow running Kermit with an effective speed of perhaps 300 to 600 baud rather than 1200 or 2400 baud. By making the packets longer, we raise the effective speed of such a circuit. The main restriction on the packet size chosen is the link, a given circuit may not pass 500 character packets. Also, BOTH Kermits must support this extension to the protocol, they will always negotiate it before any file transfer. See the notes at the end of this document for more information.

It is HIGHLY recommended that you use the CRC block check, as the default type one checksum could be inadequate for such long packets, as in:

```
Kermit-11>SET BLO 3
```

SET RECORD-FORMAT

Syntax: `SET RECORD-FORMAT {STREAM, VARIABLE}`

Kermit will, by default, create RMS11 variable length implied carriage control records for text files. You can override this and change it to create stream ascii records with the SET RECORD-FORMAT STREAM command. This is useful for RSTS/E systems if you need file compatability with BASIC Plus. This command would be most useful in a `KERMIT.INI` file, which is executed by KERMIT when Kermit starts.

SET RETRY

Syntax: (SET RETRY)*number*

SET RETRY value tells Kermit to try that many times on a NAK'ed packet before giving up. This should only be needed if the line is extremely noisy or the PDP-11 host is running very slowly due to the system load.

SET RSX

The SET RSX command is intended to deal with the peculiarities often found with RSX systems. There are currently three SET RSX commands, as in:

`Kermit-11>SET RSX FASTIO`	Default for packet reading, waits for <CR>.
`Kermit-11>SET RSX CHARIO`	Read one char at a time for packet reading.
`Kermit-11>SET RSX TC.DLU n`	Alters the TC.DLU setting.
`Kermit-11>SET RSX CONNECT ALT`	Uses a new (v2.33) connect driver which bypasses TTDRV flow control.
`Kermit-11>SET RSX CONNECT DEF`	Use old connect code (2.32)

The SET RSX command is subject to change and the above options may be removed in the future. Note the the SET RSX CHARIO may be needed when transfering files with parity enabled. This command alters the method by which a packet is read; instead of waiting for a carriage return, Kermit reads the typeahead byte count and then issues a read for that many characters. This is the same method Kermit-11 ALWAYS uses under P/OS.

SET RT-11 CREATE-SIZE

Syntax: (SET RT-11 CREATE-SIZE)*number*

The SET RT-11 CREATE value command was added to assist those RT-11 users with very small disks to be able to get files with sizes greater that half of the available contiguous space available. While this is NOT a problem going from one Kermit-11 to another Kermit-11 since the PDP-11 Kermit supports a subset of the protocol known as 'ATTRIBUTES', other Kermits may not support the exchange of file sizes (most do not). Thus if your largest contiguous space is 300 blocks and you want to get a 250 block file, the command:

```
Kermit-11>set rt-11 cre 250
```

would be needed, as RT-11 by default only allocates 50 percent of the available space.

SET RT-11 FLOW-CONTROL

Syntax: `SET RT-11 {FLOW-CONTROL, NOFLOW}`

Note that for the connect command under RT-11 you will most likely need xon/off flow control to be generated by Kermit-11. This is enabled with the SET RT-11 FLOW command. This is by default NOFLOW since the modem the author uses, a Vadic 212PA, can't handle XONs and XOFFs while in command mode. The solution here is to escape back to Kermit command mode after the remote system has been logged into, and then type SET RT-11 FLOW.

The effect of SET RT-11 FLOW is for Kermit-11, when in connect mode, to send an XOFF to the host every eight characters. When the loop in the connect module finds no more data in the input buffer, it sends up to 2 XON characters (in case the first XON got lost) to tell the remote system to start sending again. The reason for doing so is that the RT-11 multiple terminal service is very slow about handling input interrupts and does not do any of it's own flow control when it's internal ring buffer gets full. This has been tested at line speeds up to 4800 baud without losing data. This setting should not be needed for use with the XC/XL handlers.

SET RT-11 FLOW has NO effect on packet transmission, since the Kermit packet size is never mode than 96 characters, and the RT-11 input buffer is 134 characters in size.

The SET RT-11 [NO]FLOW command replaces the older SET RTFLOW [ON][OFF].

SET RT-11 VOLUME-VERIFY

Syntax: `SET RT-11 {VOLUME-VERIFY, NOVOLUME}`

Normally RT-11 Kermit-11 will check the directory header of a disk to verify that it most likely contains a valid RT-11 file structure before trying to read the directory. If for some reason your disk does not contain the standard data at offset 760 in the header, Kermit-11 will reject the disk. The SET RT-11 NOVOL command will instruct Kermit-11 to bypass that check.

SET SEND

The SET SEND command controls what Kermit-11 will be doing for outgoing packets in that you may want to alter the packet terminator and/or the start of packet character (by default, 15 octal and 1 octal respectively. See HELP SET RECEIVE for more information.

The only extra option for SET SEND is SET SEND [NO]XON. If the command SET SEND XON is give, then every packet sent will be prefixed with an XON character. This could be useful in situations where flow control is erratic. The actual intent of this option was to try to circumvent a firmware bug in the DHV11 when used under RSTS/E.

SET SPEED

Syntax: `SET SPEED` *speed*

SET SPEED value sets the line speed for the device specified via the SET LINE command, and used for the CONNECT command. Changing the speed of a terminal line requires privilege for RSTS and RSX-11M/M+. The SET SPEED command will only function with a DH11, DHV11, DZ11 or DZV11 multiline interface. Example:

```
Kermit-11>set speed 1200
```

1200 Baud would be a normal speed to use with a VA212PA or a DF03.

Please note that Kermit-11 CAN NOT change the speed of a DL11 type interface, nor can it change the speed of a PDT-150 modem port. For a PDT-150 modem port, use a command of `/M/S:`*nnnn* to change the speed to *nnnn* for the `SPEED.SAV` program.

SET TIMEOUT

Syntax: `SET TIMEOUT` *seconds*

The timeout value tells Kermit how long to wait to get a packet from the other Kermit. If system loads are high, it may be desirable to increase this beyond the default of 10 seconds.

SET TERMINAL

Syntax: `SET TERMINAL {TTY, VT100}`

The SET TERMINAL command simply controls the way which Kermit-11 prints packet counts while send or receiving a file (or group of files). The simplest way is the default, SET TER TTY. Using SET TER VT100 will cause Kermit to display headers for the numbers printed, at a possible cost in packet speed due to screen control overhead. On the PRO/350, VT100 is assumed. On RSTS/E v9.0 and later, the executive is queried for the terminal type.

SET UPDATE

Syntax: `SET {UPDATE` *number*`, NOUPDATE}`

The SET UPDATE command controls the frequency at which the packet count display is updated. The default is 1, displaying each packet. A SET UPD 0 will disable all packet count logs, whereas a SET UPD N will update the display every N packets. The SET NOUPDATE command is the same as SET UPDATE 0.

1.6.1. The DIAL Command

The DIAL command is new for version 3.29 of Kermit-11. The DIAL command is used to dial a number on an attached modem of known type (see SET MODEM). To find out the current known modems, use the SET MODEM ? command. The following example shows a RACAL-VADIC VA212 modem connect to the `XK:` port on a PRO/350 running P/OS version 2.

```
Kermit-11>set prompt PRO>
PRO>set modem va212pa
PRO>dial 5374401
Modem in command modem
Modem dialing
Connection failed, !BUSY
```

```
PRO>dial 5374411
Modem in command modem
Modem dialing
Connection made, type CONNECT to access remote
PRO>con
Enter class ? VX785A
Class start
Username: BRIAN
Password: ......................
```

See SET MODEM for more information.

1.7. System Manager's Notes

1.7.1. Odds and Ends

There are a few odds and ends that should be made aware to the system manager of any PDP-11 system regarding Kermit-11. They are as follows, grouped by operating system. Please note that installation instructions are in `K11INS.DOC` and that additional information may be in Kermit-11's online help command.

Restrictions

Prior to version 2.21, Kermit-11 did not support 8-bit prefixing. Prior to version 2.23, Kermit-11 did not support repeat character encoding.

The PRO/RT-11 version of Kermit-11 will request 8-bit prefixing due to the fact that the XC handler does not support 8BIT data. For most Kermits this should not be a problem. The XC handler always strips bit 7 from the character being sent, so the PRO/RT-11 version of Kermit will request prefixing of such. It does so internally by setting PARITY to SPACE (always clear the high bit, bit seven).

Note that this implies that a SET PARITY SPACE command will force Kermit-11 to request '8bit' prefixing in order to transfer binary files across a seven bit link.

P/OS

Kermit-11 will run on under P/OS on the Pro/350, the executable file is called `K11POS.TSK`. It does NOT run from a menu, the normal way to run it is via the RUN command in DCL. It will support the Kermit-11 attribute packets, thus a PRO/350 connected to a PDP-11 host can transparently handle binary and other types of files. The P/OS Kermit-11 can be run either as a local Kermit or a Kermit server. This has been tested under P/OS version 2 connected to both a PDP-11/23+ and PDP-11/70 RSTS/E host.

When Kermit-11 is started on the PRO, it will automatically do a `SET LINE XK0:` and a SET SPEED 9600. You can, of course, change the speed to whatever you need with the SET SPEED command. The line should be left as `XK0:`.

The top row function keys are mapped internally. Kermit-11 maps F5 (break) into a true BREAK (a space of 275 ms), F6 (interrupt) to Control-C, F10 to Control-Z, F11 to escape (octal 33) and F12 to backspace (octal 10). The incoming escape sequence DECID is intercepted to allow Kermit-11 to send back a device response of VT100.

RSTS/E

Kermit-11 runs on version 7.2 or later of RSTS/E. Due to options present in version 8, binary file transfers will not be possible under version 7.2 of RSTS/E. This is due to the use of 8 bit mode for the terminal link to allow all characters to be passed. The so called ’8BIT’ terminal setting was new as of version 8.0-06 of RSTS/E.

Any RSTS/E system running Kermit-11 will need the sysgen option for multiple private delimiters in the terminal driver. This special mode is needed since the ’normal’ RSTS/E binary terminal mode has a ’feature’ that disables binary mode whenever the terminal times out on a read. Since timeouts are essential to Kermit error recovery, binary mode can not be used for i/o.

Certain functions of Kermit-11 require that the system manager install Kermit with temporary privileges, these commands are the SYSTEM, WHO and REMOTE HOST commands. Kermit-11 does NOT need these to operate correctly.

Kermit-11 can only be built (from source, not from HEX files) under RSTS/E version 8.0 or later due to the use of RMS11 v2.0 and new assembler directives.

Support for the server remote login is only available under RSTS/E 9.0 or later. Also, a REMOTE LOGIN command to a RSTS/E server will fail unless the user has the WACNT privilege. While the LOGIN program will skip the password lookup if WACNT is present, Kermit will require a password.

RSX-11M/M+

Kermit-11 can not be installed non-checkpointable due to an apparent RMS11 bug. In other words, don’t try to install the task ’/CKP=NO’.

To use the connect command effectively, typeahead support is needed in the terminal driver. For RSX-11M+, set the typeahead buffer size high, as in SET /TYPEAHEAD=TT22:200. Also, if your connect line is TT22: (as above), use the mcr command SET/SLAVE=TT22:

Kermit-11 can only be built under RSX-11M version 4.1 or later, or under RSX-11M Plus version 2.1 or later due to the use of RMS11 v2.0 and new assembler directives.

There is a SET RSX command, see HELP SET RSX for further information.

As a side issue, please note that the file `K11POS.TSK` is quite usable under RSX, the difference being that `K11RSX.TSK` has DECNET support and RMS-11 overlayed in the task image (besides which, due to the lack author’s systems running RSX may not be up to date) linked into it, whereas K11POS has NO Decnet support but IS linked to the RMS11 library RMSRES (v2), thus K11POS saves disk space as well as supporting named directories, ala VMS style.

RT-11

Kermit-11, as of version 2.20, has been tested under RT-11 version 5.0 under the FB and XM monitors using a DZ11 line for the link, and also on a PDT-150 using the modem port for the link. It has additionally been run under Micro-11’s and the PRO/350 using the XL and XC handlers respectively.

Kermit-11 requires .TWAIT support as well as multiple terminal support (unless the XL/XC handler is used). The use of multiple terminal support allows Kermit-11 to use any type of interface sysgened, including the DZ11 and DZV11. It is possible under version 5 of RT-11 to use the XL: handler instead of the multiple terminal support. The use of the XL: driver will result in much faster file transfer at high baud rates. Note that XL: must be set up at system startup or at some time later to set the proper speed, CSR and vector.

For those users who do not have multiple terminal support and do not have the XL handler, Kermit-11 will force the use of the console for data transfers. This will require that Kermit-11 request eight bit prefixing from any other Kermit wishing to send binary data files. Additionally, you can force console mode by doing a SET LINE TT:

Please note that the device name syntax for terminal lines follows the MT unit numbers, thus if a SHO TER gave unit 5 for DZ11 line 0 the the device name would be SET LINE 5. If you use the XL handler, you would say SET LINE XL:. To force the console to be used, you would SET LINE TT:.

Additionally, Kermit-11 for RT-11 looks for its help file, `K11HLP.HLP`, on `DK:` first and then on `SY:` if the first one fails.

Full wildcarding is supported for RT-11, in the form *.type, name.*, *.* and the % character to match any single character.

Kermit-11 can only be built on RT-11 version 5.0 or later due to the use of new assembler directives.

Please note that for the connect command under RT-11 and the use of the MT service, you will most likely need xon/off flow control to be generated by Kermit-11. This is enabled with the SET RTFLOW ON command. This is by default OFF since the modem the author uses, a Vadic 212P, can't handle XONs and XOFFs while in command mode. The solution here is to escape back to Kermit command mode after the remote system has been logged into, and then type SET RTFLOW ON.

Due to overlaying constraints, the RT-11 Kermit-11 will not accept wildcards for the RENAME and DELETE commands and the REMOTE server equivalents.

The executable files are `K11XM.SAV` for the XM system and PRO/350, and K11RT4 for the FB system.

As a final (I hope) RT-11 note, see the RT-11 v5.1 Release Notes page 9-2 and chapter 12. The discussion relevant here regards the use of the XL/XC handlers.

Note that the default XL: handler vector (DL-11, DLV-11) is 300 and the CSR is 176500. For the Micro-11, PDP-11 and LSI-11, when the DL11/DLV11 interface is installed the field service representative will inform you what the CSR and VECTOR are. If they are NOT 176500 and 300, then to use the XL: handler you will need, prior to running Kermit-11, to set them. Suppose the DL vector is 400 and the CSR is 176510. Then the following DCL commands would set the addresses for RT-11:

```
.SET XL CSR=176510
.SET XL VECTOR=400
```

You SHOULD NOT ever alter these settings for XC: on the PRO/3xx. The ONLY settings you can alter for the PRO/3xx is the speed, as in DCL command SET XC SPEED=nnnn. Kermit-11 CAN NOT alter the XC: speed itself. As noted previously in this document, Kermit-11 executes the Kermit-11 command SET LIN XC: implicitly if it finds itself running on a PRO/3xx system.

Note that if your modem requires DTR to be present, you must use either an interface that asserts it (as does the PDT and PRO communications port), force it high internally to the modem, or build a cable to force it high. See HELP MODEM for more information.

TSX+

While most of the above notes for RT-11 apply for TSX+, there are a few differences of note. The first, in that TSX+ is a timesharing system, allows the Kermit user to log in normally from another system running Kermit (as in a Rainbow) and give the TSX+ Kermit the SERVER command and commence file transfer operations from the other system (ie, the Rainbow). If you are dialing INTO a TSX+ system, you need to give the TSX command:

```
.SET TT 8BIT
```

to be able to transfer data to your local (PC, other PDP-11,...) system without incurring the overhead of the Kermit protocol known as eight bit prefixing. If this is not possible, due to your local system requiring parity, or some other intervening device adds parity, then you should give Kermit the command SET PARITY SPACE, to let Kermit know that it can't send binary data as-is.

To use Kermit-11 to dial out from the TSX+ system, the following commands are needed. Note that TSX+ commands will be preceeded by the normal RT-11 prompt, the ever present DOT ('.'), whereas Kermit-11 commands will be prefixed by the default Kermit-11 prompt, 'Kermit-11>':

```
.SET CL LINE=n          Where 'n' is the unit number
.SET CL NOLFOUT
.SET CL SPEED=n         Where 'n' is the speed for that unit
.ASS CL XL
Kermit-11>SET LIN XL:
Kermit-11>CONNECT
```

As of Kermit-11 version 3.44, you may use CL directly in the SET LINE command, as in:

```
.SET CL3 LINE=3
.R K11XM
Kermit-11>SET LIN CL3
Kermit-11>SET SPEED 1200
Kermit-11>CONNECT
```

A sample command file in actual use is:

```
SET CL3 LINE=3
SET CL3 NOLFOUT
SET CL3 TAB
SET CL3 FORM
SET CL3 SPEED=2400
ALLOCATE CL3:
R K11XM
DEALLOC CL3
SET CL3 LFOUT
SET CL3 LINE=0
SH CL
```

If you are running PRO/TSX+, then Kermit will make the assignment of LINE 3 to either CL0 or CL1 if you are running Kermit from the console, ie, LINE 1. The speed will default to the last SET SPEED or the speed set at system boot.

Lastly, TSX+ needs PLAS support to use `K11XM.SAV`, see the installation notes for further data.

RSTS/E version 9.x

RSTS/E does not control modems signals in the manner that RSX or VMS does. VMS always asserts DTR whereas RSTS/E will not assert DTR until the terminal driver can see RCD (also known as DCD) which is pin 8 (eight) for the RS232 connection. To connect directly to a modem (like a VADIC 212, sorry, no DEC modems here) we must do one of two things:

1. Force the modem (via strapping options or whatever) to assert RCD (DCD) pin 8, thus RSTS/E will see carrier and raise DTR (pin 20 for RS232)

2. Set the terminal to LOCAL (RSTS/E V9 syntax 'SET TER TTxx:/NODIAL/PERM') and break pin 20 (DTR) and connect pin 20 to 8 on the modem side. This will cause the modem to be able to dial out and allow RSTS/E to connect to it. You will also need to have the modem assert RCD, pin 8. Keep in mind that the Kermit-11 command DISCONNECT (or HANGUP) will not function if a line is set to NODIAL (INIT SET syntax 'LOCAL'). This has been tested on a Racal Vadic VA212.

3. Break pin 8 (RCD) and loop DTR (pin 20) on the CPU side to RCD (pin 8) on the CPU side. Then use the command SET DTR in Kermit-11 to get RSTS to raise DTR and thus loop it's DTR signal back to RCD. See the next note regarding this.

For those of you who have port switches such as the Gandalf type, there is one additional problem. For Gandalf, suppose you want to connect a DZ11 line to to an AMTB2. You will have a problem, in that the Gandalf AMTB2 wants to see RCD (DCD) asserted to make a connection. What you may need to do is this:

Make a cable for the DZ11 to AMTB2 port as follows:

```
        CPU side                          AMTB2 side
                    20--|
                    8---|-----------8
                    7---------------7
                    3---------------2
                    2---------------3
```

Note that 20 is tied to 8 on the CPU side. Also, 2 is swapped for 3.

Then, the Kermit-11 command SET DTR, which forces RSTS to raise DTR for 30 seconds, will cause the DTR signal to loop back to the RCD (DCD) signal and thus tell RSTS that there is carrier detect which will raise DTR (the chicken or egg question) and get things rolling. The Kermit-11 HANGUP (or DISCONNECT) command will drop DTR and force the modem to break the connection.

RSX and Modems

While the author's experience on RSX is limited, the following notes may be of use. Dialing out on a LOCAL line will often require that the modem assert internally DTR. If a line is set REMOTE on RSX, the driver will assert DTR and RTS. For a modem, like a VA212PAR strapped at the factory defaults, this will cause the modem to assert DSR and RCD. On the VADIC in particular, the modem will drop RCD during a DIAL command unless the modem is configured to assert RCD continuously. For dialing out, ideally the modem should be able to assert RCD via an option or internally settable strap or switch. If this is not possible, an alternative is to break line 8 (RCD) and jumper DTR (20) to RCD (8) on the CPU side. This will force RSX to always see carrier detect and allow a dial sequence to complete. The Kermit-11 command DISCONNECT (or HANGUP) will still disconnect the modem as the modem will drop from the line when it sees DTR go low (assuming the modem is not strapped to assert DTR internally).

1.8. Typical Kermit-11 Transfer Rates

Some sample timings for Kermit-11 and long packet support. The packet size in the RSTS/E to P/OS was 500 bytes, the size from RSTS/E to RSTS/E was 700 bytes. These sizes are somewhat arbitrary, they depend more on the system's buffering capabilities than anything else.

Host buffering capabilities:

P/OS	500 (estimated)
RSTS/E 9.0 or later	up to 7000, given sufficient system pool
RSX-11M+	255 (I/D space CPU only)
RSX-11M	34
RT-11	134 (could be larger with mod to XC/XL)

As it can be seen, large packets make sense only for RSTS/E, P/OS and RSX-11M+ if one wishes to avoid XON/XOFF overhead at high speeds. It should be possible to run larger packets on M+ and RT-11 at lower speeds.

File transferred: `K11POS.TSK`, size 102,400 bytes (200 disk blocks). Actual data packet characters AFTER prefixing was 120,857.

Time seconds	Speed baud	Data rate	Comments
1436	1200	84/sec	11/44 to PRO/350, 'Classic' Kermit local phone call
1237	1200	97/sec	11/44 to PRO/350, 500 Char packets local phone call
2915	1200	41/sen	11/44 to PRO/350, 'Classic' Kermit local call, 1 second ACK delay.
1492	1200	81/sec	11/44 to PRO/350, 500 Char packets local call, 1 second ACK delay.
304	9600	397/sec	11/44 to 11/44, 'Classic' Kermit, connected locally via Gandalf switch.
245	9600	493/sec	11/44 to 11/44, 700 char packets, connected locally via Gandalf switch.

The last two timings are much lower than the line speed due to the fact the the PDP 11/44 is running 100% busy trying to keep up with character interrupts using a normal terminal driver. A special purpose driver, such as the XK driver found on P/OS, would have lower overhead and allow somewhat faster data rates.

Long packets were chosen for Kermit-11 due to the lack of suitable interrupt driven i/o (at this time) under one of the operating systems, RSTS/E. The Sliding Windows would likely function better in those situations where the circuit delay is much higher, or when the circuit can not accommodate large packet sizes.

1.9. Common Problems

Connection Fails

Check modem control signals. RSX needs TC.DLU set to two to talk to a dial out modem, otherwise you will need to strap or jumper signals in the modem to have carrier detect set high. RSTS/E also should have the modem assert carrier detect. If not, see the previous notes about modems. If all else fails, put a breakout box in the line and observe what signals are present.

File Transfer Fails.

If the file transfer aborts on retries immediately, there may be a parity problem. If the problem shows up on binary files, try a SET PAR SPACE command to Kermit; that will force eight bit data to be prefixed into seven bits. If you instead get a retry about once every 10 seconds, the other Kermit is not responding and your Kermit is timing out. Check to see if your connection is still present, and try the SET PARITY command.

If you are sending binary data between unlike Kermits, you will most likely have to give the proper command to each to prepare them for the binary data; this is the SET FILE command; for Kermit-11 it's SET FIL BIN (or SET FIL TYP FIX); for VMS Kermit it's SET FIL TYP FIX.

If your Kermit's packets are being echoed back, try a SET SEND START value command for your Kermit, and a SET REC START samevalue for the other Kermit. This will force Kermit to ignore any echoed packets as they won't have the default start of packet character (a CONTROL A, octal 1).

Index

Table of Contents

List of Tables