NAME

sz − send a file using the zmodem protocol

SYNOPSIS

sz [**−LNbdefnopqruvy+**] [**−ci** *command*] [**−Ll** n] [**−t** *timeout*]

OPTIONS

−L Use *n*-byte packets

−N Overwrite if source is newer/longer

−b Binary file

−c Send command for execution

−d Convert dot to slash in names

−e Escape for all control characters

−f Send full path name

−i Send command and return immediately

−l Flow control every *n* packets

−n Overwrite destination if source is newer

−o Use old (16-bit) checksum

−p Protect file if it already exists

−q Quiet; opposite of verbose

−r Resume interrupt file transfer

−t Set *timeout* in tenths of a second

−u Unlink file after successful transmission

−v Verbose; opposite of quiet

−y Yes, clobber existing files

−+ Append to an existing file

EXAMPLES

sz file </dev/tty01 >/dev/tty01
Send *file*

DESCRIPTION

XMODEM, YMODEM, and ZMODEM are a family of protocols that are widely used is the MS-DOS world for transferring information reliably from one computer to another. In all of these protocols, a series of bytes are sent from one computer to the other, and then an acknowledgement is sent back to confirm correct reception. Checksums are used to detect errors so that transmission is reliable even in the face of noisy telephone lines. *Sz* is a program that sends a file sent from another computer using the zmodem protocol. The file can be received using *rz*.

Sz uses the ZMODEM error correcting protocol to send one or more files over a dial-in serial port to a variety of programs running under MINIX, UNIX, MS-DOS, CP/M, VMS, and other operating systems. It is the successor to XMODEM and YMODEM.

ZMODEM greatly simplifies file transfers compared to XMODEM. In addition to a friendly user interface, ZMODEM provides Personal Computer and other users an efficient, accurate, and robust file transfer method.

ZMODEM provides complete end-to-end data integrity between application programs. ZMODEM's 32 bit CRC catches errors that sneak into even the most advanced networks.

Output from another program may be piped to *sz* for transmission by denoting standard input with −:

"ls −l | sz −

The program output is transmitted with the filename *sPID.sz* where PID is the process ID of the *sz* program. If the environment variable *ONAME* is set, that is used instead. In this case, the command:

"ls −l | ONAME=con sz −ay −

will send a 'file' to the PC-DOS console display. The **−y** option instructs the receiver to open the file for writing unconditionally. The **−a** option causes the receiver to convert UNIX newlines to PC-DOS carriage returns and linefeeds. On UNIX systems, additional information about the file is transmitted. If the receiving program uses this information, the transmitted file length controls the exact number of bytes written to the output dataset, and the modify time and file mode are set accordingly.

If *sz* is invoked with $SHELL set and if that variable contains the string *rsh* or *rksh* (restricted shell), *sz* operates in restricted mode. Restricted mode restricts pathnames to the current directory and *PUBDIR* (usually */usr/spool/uucppublic*) and/or subdirectories thereof.

The options and flags available are:

−+

Instruct the receiver to append transmitted data to an existing file.

−a

Convert NL characters in the transmitted file to CR/LF. This is done by the sender for XMODEM and YMODEM, by the receiver for ZMODEM.

−b

Binary override: transfer file without any translation.

−c

Send COMMAND (follows *c*) to the receiver for execution, return with COMMAND's exit status.

−d

Change all instances of '.' to '/' in the transmitted pathname. Thus, C.omenB0000 (which is unacceptable to MS-DOS or CP/M) is transmitted as C/omenB0000. If the resultant filename has more than 8 characters in the stem, a '.' is inserted to allow a total of eleven.

−e

Escape all control characters; normally XON, XOFF, DLE, CR-@-CR, and Ctrl-X are escaped.

−f

Send Full pathname. Normally directory prefixes are stripped from the transmitted filename.

−i

Send COMMAND (follows *i*) to the receiver for execution, return Immediately upon the receiving program's successful reception of the command.

−L

Use ZMODEM sub-packets of length *n* (follows *L*). A larger *n* (32 <= *n* <= 1024) gives slightly higher throughput, a smaller one speeds error recovery. The default is 128 below 300 baud, 256 above 300 baud, or 1024 above 2400 baud.

−l

Wait for the receiver to acknowledge correct data every *n* (32 <= *n* <= 1024) characters. This may be used to avoid network overrun when XOFF flow control is lacking.

−n

Send each file if destination file does not exist. Overwrite destination file if source file is newer than the destination file.

−N

Send each file if destination file does not exist. Overwrite destination file if source file is newer or longer than the destination file.

−o

Disable automatic selection of 32 bit CRC.

−p

Protect existing destination files by skipping transfer if the destination file exists.

−q

Quiet suppresses verbosity.

−r

Resume interrupted file transfer. If the source file is longer than the destination file, the transfer commences at the offset in the source file that equals the length of the destination file.

−t
Change timeout. The timeout, in tenths of seconds, follows, the **−t** flag.

−u
Unlink the file after successful transmission.

−w
Limit the transmit window size to *n* bytes (*n follows* **(enw**).

−v
Verbose causes a list of file names to be appended to */tmp/szlog*.

−y
Instruct a ZMODEM receiving program to overwrite any existing file with the same name.

−Y
Instruct a ZMODEM receiving program to overwrite any existing file with the same name, and to skip any source files that do have a file with the same pathname on the destination system.

Examples

Below are some examples of the use of *sz*.

"sz −a *.c

This single command transfers all .c files in the current directory with conversion (**−a**) to end-of-line conventions appropriate to the receiving environment.

"sz −Yan *.c *.h

Send only the *.c* and *.h* files that exist on both systems, and are newer on the sending system than the corresponding version on the receiving system, converting MINIX to MS-DOS text format.

SEE ALSO

rz(1), **term**(1).