## NAME

sx, sb, sz – XMODEM, YMODEM, ZMODEM file send

## SYNOPSIS

sz [–**+abdefkLlNnopqTtuvyYZ**] *file ...*
sb [–**adfkqtuv**] *file ...*
sx [–**akqtuv**] *file*
sz [–**oqtv**] **-c COMMAND**
sz [–**oqtv**] **-i COMMAND**
sz -TT

## DESCRIPTION

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

*Sz* is not intended be called from *cu(1)* or other communications programs. Unix flavors of Omen Technology's Professional-YAM communications software are available for dial-out applications. ®

**Sz** sends one or more files with ZMODEM protocol.

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.

Advanced file management features include AutoDownload (Automatic file Download initiated without user intervention), Display of individual and total file lengths and transmission time estimates, Crash Recovery, selective file transfers, and preservation of exact file date and length.

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.

**Sb** batch sends one or more files with YMODEM or ZMODEM protocol. The initial ZMODEM initialization is not sent. When requested by the receiver, **sb** supports **YMODEM-g** with "cbreak" tty mode, XON/XOFF flow control, and interrupt character set to CAN (^X). **YMODEM-g** (Professional-YAM **g** option) increases throughput over error free channels (direct connection, X.PC, etc.) by not acknowledging each transmitted sector.

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.

**Sx** sends a single *file* with **XMODEM** or **XMODEM-1k** protocol (sometimes incorrectly called "ymodem"). The user must supply the file name to both sending and receiving programs.

Iff **sz** is invoked with $SHELL set and iff 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 fourth form sends a single COMMAND to a ZMODEM receiver for execution. **Sz** exits with the COMMAND return value. If COMMAND includes spaces or characters special to the shell, it must be quoted.

The fifth form sends a single COMMAND to a ZMODEM receiver for execution. **Sz** exits as soon as the receiver has correctly received the command, before it is executed.

The sixth form (sz -TT) attempts to output all 256 code combinations to the terminal. In you are having difficulty sending files, this command lets you see which character codes are being eaten by the operating system.

If **sz** is invoked with stdout and stderr to different datasets, Verbose is set to 2, causing frame by frame progress reports to stderr. This may be disabled with the **q** option.

The meanings of the available options are:

\h@|216u+0u@(backslash) (VMS) Force the next option letter to upper case.

**+** Instruct the receiver to append transmitted data to an existing file (ZMODEM only).

**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** (ZMODEM) Binary override: transfer file without any translation.

**c COMMAND**
Send COMMAND 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 MSDOS 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 COMMAND**
Send COMMAND to the receiver for execution, return Immediately upon the receiving program's successful recption of the command.

**k** (XMODEM/YMODEM) Send files using 1024 byte blocks rather than the default 128 byte blocks. 1024 byte packets speed file transfers at high bit rates. (ZMODEM streams the data for the best possible throughput.)

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

**l N** 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** (ZMODEM) Send each file if destination file does not exist. Overwrite destination file if source file is newer than the destination file.

**N** (ZMODEM) Send each file if destination file does not exist. Overwrite destination file if source file is newer or longer than the destination file.

**o** (ZMODEM) Disable automatic selection of 32 bit CRC.

**p** (ZMODEM) Protect existing destination files by skipping transfer if the destination file exists.

**q** Quiet suppresses verbosity.

**r** (ZMODEM) 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.

**rr** As above, but compares the files (the portion common to sender and reciever) before resuming the transfer.

**t tim** Change timeout to *tim* tenths of seconds.

**u** Unlink the file after successful transmission.

**w N** Limit the transmit window size to N bytes (ZMODEM).

**v** Verbose causes a list of file names to be appended to /tmp/szlog . More v’s generate more output.

**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.

**Z** Use ZMODEM file compression to speed file transfer.

## EXAMPLES

**ZMODEM File Transfer** (Unix to DSZ/ZCOMM/Professional-YAM)

```
% sz –a *.c
```

This single command transfers all .c files in the current Unix directory with conversion (**–a**) to end of line conventions appropriate to the receiving environment. With ZMODEM AutoDownload enabled, Professional-YAM and ZCOMM will automatically recieve the files after performing a security check.

```
% 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 Unix to DOS text format.

```
$ sz -\Yan file1.c file2.c file3.c foo.h baz.h ®(for VMS)
```

**ZMODEM Command Download** (Unix to Professional-YAM)

```
 cpszall:all
    sz –c "c:;cd /yam/dist"
    sz –ya $(YD)/*.me
    sz –yqb y*.exe
    sz –c "cd /yam"
    sz –i "!insms"
```

This Makefile fragment uses **sz** to issue commands to Professional-YAM to change current disk and directory. Next, **sz** transfers the *.me* files from the $YD directory, commanding the receiver to overwrite the old files and to convert from Unix end of line conventions to PC-DOS conventions. The third line transfers some *.exe* files. The fourth and fifth lines command Pro-YAM to change directory and execute a PC-DOS batch file *insms* . Since the batch file takes considerable time, the **–i** form is used to allow **sz** to exit immediately.

**XMODEM File Transfer** (Unix to Crosstalk)

```
% sx –a foo.c
ESC
rx foo.c
```

The above three commands transfer a single file from Unix to a PC and Crosstalk with *sz* translating Unix newlines to DOS CR/LF. This combination is much slower and far less reliable than ZMODEM.

## ERROR MESSAGES

"Caught signal 99" indicates the program was not properly compiled, refer to "bibi(99)" in rbsb.c for details.

## SEE ALSO

rz(omen), ZMODEM.DOC, YMODEM.DOC, Professional-YAM, crc(omen), sq(omen), todos(omen), tocpm(omen), tomac(omen), yam(omen)

Compile time options required for various operating systems are described in the source file.

## VMS VERSION

The VMS version does not support wild cards. Because of VMS DCL, upper case option letters muse be represented by \ proceding the letter.

The current VMS version does not support XMODEM, XMODEM-1k, or YMODEM.

VMS C Standard I/O and RMS may interact to modify the file contents.

## FILES

32 bit CRC code courtesy Gary S. Brown.

sz.c, crctab.c, rbsb.c, zm.c, zmodem.h Unix source files

sz.c, crctab.c, vrzsz.c, zm.c, zmodem.h, vmodem.h, vvmodem.c, VMS source files.

/tmp/szlog stores debugging output (sz -vv) (szlog on VMS).

## TESTING FEATURE

The command "sz -T file" exercises the **Attn** sequence error recovery by commanding errors with unterminated packets. The receiving program should complain five times about binary data packets being too long. Each time **sz** is interrupted, it should send a ZDATA header followed by another defective packet. If the receiver does not detect five long data packets, the **Attn** sequence is not interrupting the sender, and the **Myattn** string in **sz.c** must be modified.

After 5 packets, **sz** stops the "transfer" and prints the total number of characters "sent" (Tcount). The difference between Tcount and 5120 represents the number of characters stored in various buffers when the Attn sequence is generated.

## NOTES

*Sz* is not intended be called from *cu(1)* or other communications programs. Unix flavors of Omen Technology's Professional-YAM communications software are available for dial-out applications.

If a program that does not properly implement the specified file transfer protocol causes *sb* to "hang" the port after a failed transfer, either wait for *sb* to time out or keyboard a dozen Ctrl-X characters. Every reported instance of this problem has been corrected by using ZCOMM, Pro-YAM, DSZ, or other program with a correct implementation of the specified protocol.

Many programs claiming to support YMODEM only support XMODEM with 1k blocks, and they often don't get that quite right. XMODEM transfers add up to 127 garbage bytes per file. XMODEM-1k and YMODEM-1k transfers use 128 byte blocks to avoid extra padding.

YMODEM programs use the file length transmitted at the beginning of the transfer to prune the file to the correct length; this may cause problems with source files that grow during the course of the transfer. This problem does not pertain to ZMODEM transfers, which preserve the exact file length unconditionally.

Most ZMODEM options are merely passed to the receiving program; some programs do not implement all of these options.

Circular buffering and a ZMODEM sliding window should be used when input is from pipes instead of acknowledging frames each 1024 bytes. If no files can be opened, **sz** sends a ZMODEM command to echo a suitable complaint; perhaps it should check for the presence of at least one accessible file before getting hot and bothered.

A few high speed modems have a firmware bug that drops characters when the direction of high speed transmissson is reversed. The environment variable ZNULLS may be used to specify the number of nulls to send before a ZDATA frame. Values of 101 for a 4.77 mHz PC and 124 for an AT are typical.

## BUGS

On at least one BSD system, sz would abend it got within a few kilobytes of the end of file. Using the "-w 8192" flag fixed the problem. The real cause is unknown, perhaps a bug in the kernel TTY output routines.

The test mode leaves a zero length file on the receiving system.