ZMD File Transfer Program for the Z80 RCP/M

External Reference Specification

ZMD - Version 1.50

CONTENTS

APPENDICES

1.0 Introduction to ZMD

ZMD is a Remote CP/M File Transfer Program written in MICROSOFT MACRO-80 language using ZILOG Z80 assembler. It will run on any Z80 microprocessor using CP/M 2 or 3 and provides full support for systems running ZCMD or ZCPR CCP replacement. ZMD can be configured to run under most popular remote console programs, but is primarily targeted for BYE5+, MBYE and BYE3, or as an entirely stand alone program using modem/port overlays for modem I/O. Full support is provided for most of the popular BBS software such as PBBS, MBBS, RBBS, Metal, Oxgate, ZBBS and others. Communications programs such as MODEM7, MEX, MEX+, PROCOMM, ProYam, QMODEM, IMP and others are all also fully compatible.

This manual is not intended to be specific in detail to the internal characteristics of ZMD and utilities. It was written as a guide to help the implementer decide which of the options and features will be used on his system.

An internal reference manual covering the ZMDSUBS.REL file and user's manual has been in the making. Additional modem/port overlays will be added to the ZMDOVLn.LBR file as they are created. A ZMDCLOCK.LBR is being considered for organizing approved RTC inserts. There are many planned features. If you have any ideas for changes or additions, feel free to contact me.

```
Robert W. Kramer III       RI RCPM-RDOS  (128mb)
1569 40th St.              300/1200/2400 - 24hrs
Rock Island, Il  61201     Data:  (309) 786-6227
                           Voice: (309) 786-6711
```

Features:

- o Menu driven install program with intelligence
- o Automatic installation of new releases
- o User definable modem/port I/O routines
- o 5 seconds to gather 255 filenames in batch mode!
- o Multiple drive/users fully supported in Batch send
- o Overlayable .COM file installation
- o Upload routing to multiple drive/user areas
- o Time and Datestamping in upload description header
- o 100% MBYE/BYE3/BYE5 compatible
- o Truly capable of standalone operation
- o Automatic batch enable (eliminates the 'SB' command)
- o Automatic 'k' block detection of host
- o Fully CP/M 2.n and CP/M 3.n compatible
- o ARC/ARK/LBR member extraction support
- o Automatic BYEBDOS - I/O overlay detection
- o Faster error checking resulting in quicker transfers
- o No delays between files in batch transfers

2.0 Applicable Files

Filename	Kb	Purpose
ZMDHDR .Z80		This file contains all the program switches and values looked at by all the ZMD programs. It is not needed if using the installation program ZINSTL.
ZMDSUBS.REL		Contains the common subroutines for all ZMD and support utilities. This is used when linking a newly assembled version of one of the programs. Source code is not released for this file.
ZINSTL .COM		This is the install program used to configure the ZMD and utility programs. When first ran, ZINSTL reads the first 10 records of ZMD.COM on the current drive/user. If not found, prompts for a new drive/user to log into and tries again. A fast and easy to use menu driven display showing all current program settings. After editing any of the switches/values, a selection of 'J' in the main menu will write the current configuration (first 11 records of ZINSTL) back to ZMD and ALL of it's utilities. Use it on the fly to change switches and values quickly without having to reassemble anything. This takes only seconds and allows you the versatility of changing addresses, file descriptors/categories, upload/download drive/user areas, time restrictions, etc. Anything found in the ZMDHDR.Z80 configuration header can be edited from within ZINSTL. Source code is not provided for this program. If you make any modifications to the ZMDHDR.Z80 configuration file, ZINSTL will NO longer be able to install your .COM files properly.
ZMD .COM **ZMD .Z80**		Main file transfer program. Leave this one online for public use (Usually A0:). It provides full system security while presenting a user freindly interface to uploads and downloads. YMODEM 1k and YMODEM 1k Batch, XMODEM 128 byte CRC, and XMODEM 128 byte checksum are supported with automatic protocol detect.

Filename	Kb	CRC	Purpose
ZMAP .COM **ZMAP .Z80**			Public upload/download matrix display program. Also completely installable with the ZINSTL program. Private areas are shown only if WHEEL byte is on.
ZFORS .COM **ZFORS .Z80**			Sysop description file utility. This one allows the Sysop to add descriptions to the FOR text file (or BBS message base). This program will not run unless descriptions are enabled.
ZNEWS .COM **ZNEWS .Z80**			Sysop file transfer log utility. This one allows the Sysop to append the ZMD.LOG file with a new upload entry. This procedure currently supports 1 entry per program run.
ZMDEL .COM **ZMDEL .Z80**			Sysop file transfer log purge utility. Cleans the ZMD.LOG file of all entries except for "R" uploaded entries. This keeps the log file at a reasonable size.
ZNEWP .COM **ZNEWP .Z80**			Public file transfer log utility allows callers to quickly view statistics of recent uploads such as the drive/user, file size in kilobytes, date of upload (if clock/date options are enabled), etc. This program is usually placed on A0:.
ZFORP .COM **ZFORP .Z80**			Public upload description search and display utility. Users can view the descriptions online of the files recently uploaded. String search capabilities are supported to allow 'picking' the descriptions desired. A caller can quickly view all descriptions pertaining to 'IBM', 'Z80' or any string that will fit on the host system's command line. Multiple string search is also supported, as is date searches (if enabled) and files located in a specific drive/user (if that feature is enabled). A nice companion for ZMD.

NOTE: To prevent possible discrepencies within the ZMDSUBS subroutines file, all programs have been designed to abort with error if the program's main feature is disabled. For example, ZFORS will not run unless it has been installed with descriptions enabled. ZMD can not be disabled, however will abort with error if the modem I/O routines are not found. Please do not try to disable this program function.

3.0 Installation

3.1 Local CRT Clear Screen Sequences

CLRSCRN set to YES enables ZMD and all utility files to clear your screen locally during batch file transfers and menu displays. If your terminal requires a ^Z (1AH), leave **CLRSTR** alone. Otherwise you will have to include your clear screen sequence at label CLRSTR. Up to 6 bytes are allowed. This CLS sequence is terminated with a '$'.

STOP: If running ZMD and utilities as a stand alone package and your local console output vector address cannot be calculated using standard CP/M conventions (i.e. your BIOS JP table has been altered), you will need to provide the address yourself at label CONOUT in your port/modem overlay. This is crucial to ZMD even when not using the clear screen feature as the record count is displayed locally during the file transfer and any output of this nature to the modem at this time would cause serious problems.

3.2 Input/Output

3.2.1 Modem

ZMD provides full support for the Extended BDOS calls set up by some Remote Console Programs such as BYE5 or MBYE44+. In order to use this method, simply skip the overlay process outlined in APPENDIX A. If no overlay is included, ZMD will use BYE's modem I/O routines already established. If an attempt to set the current user area to 241 returns a value of 77 in register A, the following calls to BDOS must be handled by your BYE program:

BDOS Call	**Function**
61	**Check receive ready**
62	**Check send ready**
63	**Send a character**
64	**Receive a character**
65	**Check carrier**
66	**Console status**
67	**Console input**
68	**Local console output**

If your BYE program does not set up it's modem I/O routines for public access by outside programs, you will have to follow the overlay procedure outlined in APPENDIX A on **Modem Interface** to install your modem routines.

NOTE: This feature is fully automatic. No matter which method you use, ZMD will know.

3.2.2 Microprocessor Speed

MHZ is defined to represent your system's microprocessor speed. This allows ZMD to accurately time delays and sleepy caller timeouts. Must be integer value in the range of 1-9.

3.2.3 Modem Speed Indicator Byte

MSPEED should contain the address of your modem speed indicator byte. This is needed so that ZMD can calculate the amount of time needed for transfers and for transfer speed insertion in the log file entry. Your BYE program should stuff this integer value somewhere in memory. An integer value of 1-9 must correspond as illustrated:

Value	Bps	Value	Bps
1	300	7	4800
5	1200	8	9600
6	2400	9	19200

3.2.4 Disk Write Lock Flag

WRTLOC is the high memory byte toggle used for disabling BYE's ability to hang up when the caller's time is up or carrier is lost. This allows current disk write operations to be completed first. Check your BBS documentation - many modern systems don't need this. If unsure, set this to NO. Code to set and reset this toggle assumes it to be located **LOCOFF** bytes from the JP COLDBOOT vector at the beginning of the BYE BIOS jump table. (YES for MBBS and PBBS).

3.2.5 Sleepy Caller Timeout

DESWAIT is the number of minutes of inactivity during an upload description or any prompt before aborting the input routine. (0-255, 0 disables this feature). If a timeout occurs during description entry, the current disk buffer is written to disk and the program exits to CP/M giving control back to your BYE program (if WRTLOC is YES).

3.2.6 Minimum Speed for 1k Blocks

MINKSPD is the minimum modem speed allowed to transfer files using 1k blocks. If on a network such as PC Pursuit, and are able to receive incoming calls, set this byte to 1. The delays these networks use to send data back and forth make 1k packets advantageous to even 300 bps users. If not on a network such as PC Pursuit, it's a matter of preference, See illustration in figure 3.2.3 for list of modem speed values.

3.2.7 File Transfer Buffer Queue

BUFSIZ allows you to change the number of 1k blocks before writing buffer to disk. Normal disk systems can transfer 16k from computer to disk in 2-3-4 seconds or less. Some very slow 5-1/4" floppy systems (such as North Star) may take up to 20-30 seconds to transfer 16k. This would cause several timeouts at 10 seconds each. If you experience any timeouts, try changing this to something smaller, perhaps 8k or even 4k.

3.2.8 Page Pausing

PAGLEN is set to the number of lines to display in between [more] pauses and menu displays. (Set to 0 to disable page pauses.

3.2.9 Automatic Calculation of Disk Buffer Size

STDBUF enabled tells ZMD programs to calculate the amount of memory available for routines using DBUF disk buffer. OUTSIZ will adjust according to the contents of location 6 and 7. If BYE is running, this value is used as is, else if BYE is **NOT** running, 0806H is subtracted from this address location 6 and 7. In either case, the end of program value determined at time of assembly is subtracted from the result of the above to situations. Leftover bytes in L are then discarded to leave an even page boundary.

NOTE: MOST ALL systems will benefit with STDBUF enabled, since maximum number of uploads allowed in batch receive mode is directly related to the amount of memory available. However, If your system uses memory located below the CCP or BYE program, you will need to disable this, in which case OUTSIZ will be set to 16k no matter what. If your system hangs or behaves strangely during description file read/write operations, disable this - try it enabled first.

3.3 Timekeeping

3.3.1 BYE Clock Support

CLOCK enabled will allow ZMD and utilities to look at clock and date information retrieved by BYE (BYE5, MBYEnn and MBYE44+ with extended BDOS enabled) from your system and stored in high memory. The address of this high memory buffer is calculated as the address at (JP COLBOOT+25). If you set this to YES, the following information is retrieved:

Location	**Contents**	**Range**
(JP COLDBOOT+24)	**Maximum time allowed**	**0-255**
(JP COLDBOOT+25)	**LSB of RTCBUF address**	-----
(JP COLDBOOT+26)	**MSB of RTCBUF address**	-----

(RTCBUF+0)	**Current hour**	**0-23**
(RTCBUF+1)	**Current minute**	**0-59**
(RTCBUF+2)	Current seconds	0-59
(RTCBUF+3)	Current century	19
(RTCBUF+4)	**Current year**	**0-99**
(RTCBUF+5)	**Current month**	**1-12**
(RTCBUF+6)	**Current day**	**1-31**
(RTCBUF+7)	**LSB of TOS**	**0-255**
(RTCBUF+8)	MSB of TOS	-----

Clock/date reader code inserts are not supported under this configuration. ZMD programs support all clock/date features with this switch enabled and clock/date reader code is in BYE.

3.3.2 ZMD Internal Clock/Date Reader Code

RTC should be enabled if you cannot take advantage of BYE's RTCBUF explained above and you wish to include your system specific clock and date reader code at label RTCTIM in the ZMDHDR.Z80 configuration table. Be sure to initialize all bytes with binary values (see **Range** column in table 3.3.1 for minimum and maximum values). BCDBIN may be requested from ZMDSUBS.REL to convert binary code decimal value in A register to binary value in A register. Delete all ';<=' lines after your code is installed. This concept is not valid if CLOCK switch is enabled and BYE is running with clock and date reader code installed.

NOTE: Make sure you do not overrun the modem I/O patch area when you insert your clock code.

3.3.3 Restricting Downloads to Time Left

TIMEON switch enabled tells ZMD to restrict downloads to the callers time left on system. If CLOCK is enabled and MBYE/BYE5 is running, BYE's maximum time allowed byte (JP COLDBOOT+24) is used for comparison. If CLOCK is NO (or MBYE/BYE5) and RTC is yes, MAXTOS should be poked by your RTC insert. If CLOCK and RTC are both NO, MAXMIN will be used as a default and incremented or decremented as file transfers occur.

3.3.4 Maximum Minutes Allowed

MAXMIN should be set to your liking if CLOCK and RTC are both NO, and TIMEON is YES. This value will be the default. This is decremented/incremented each time a file transfer is made and the caller is logged off when it reaches 0.

3.3.5 Display Time on System Messages

DSPTIM is set YES to have time on system messages displayed at the start and exit of ZMD. If CLOCK is YES, and MBYE/BYE is running, TOS is gotten from BYE's TON word (in RTCBUF). If CLOCK is NO, or MBYE/BYE5 is not running, the current minutes allowed is subtracted from the original value of MAXMIN at program startup. The result is displayed as time on. This value is actually a tally:

Original MAXMIN + upload time - download time = time on

3.3.6 Logon Hour/Minute Addresses

LHOUR is the address of the caller's logon hour byte set by your BYE or BBS program when the caller logs on. LHOUR+2 is the logon minute. Both values contained in these addresses should be in binary. This should only be set if RTC and either TIMEON or DSPTOS are set YES. (Not used with CLOCK).

3.4 Access Restrictions

3.4.1 ZCMD/ZCPR

ZCPR (bit 0 in ACCMAP) should be set to YES if you intend on monitoring WHEEL byte status or need to restrict receiving SYS, RCP, NDR file types. If this bit is set NO, WHEEL is always 0.

3.4.2 WHEEL Byte

WHEEL byte toggle is for very special users. If ZCPR is set to YES, ZMD will monitor the byte located at this address. If it's 0, time, drive/user/filename and access restrictions remain in force. If this byte is non-zero, all restrictions are bypassed.

3.4.3 Maximum Drive/User

USEMAX should be set to YES if you want to use ZCPR's low memory bytes to keep track of maximum drive and user. ZMD will use the values at locations **DRIVMAX** and **USRMAX** for maximum drive/user. If USEMAX is NO, hardcode **MAXDRV** and **MAXUSR** to your own requirements.

3.4.4 BYE/BBS Program Access Flag Byte

ACCESS if enabled, tells ZMD that your BYE/BBS software support an access flags register. This flag register (AFBYTE) is 1 data byte in length and contains 8 flag bits correspdonding to common BBS restrictions. ZMD can check this register before allowing the 'RM' option to upload preformatted message files to your BBS's message base, or to use the 'RW' option for 'privileged user' upload without being required to give upload descriptions. If enabled, set **ACCOFF** to reflect the number of bytes from JP COLDBOOT to AFBYTE flags byte. ZMD inspects AFBYTE for the following flag data:

```
           Bit:    7 6 5 4 3 2 1 0
                   | | | | | | | |
Privileged user ---* | | | | | | |
         Upload -----* | | | | | |   * Of these bits, only 3,
       Download -------* | | | | |    5, 6 and 7 are used by
           CP/M ---------+ | | | |     ZMD.  Bit numbers are
          Write -----------* | | |     powers of 2, with bit
           Read -------------+ | |       0  being  the  least
            BBS ---------------+ |    significant bit of the
         System -----------------+    byte.
```

3.5 Upload Configurations

3.5.1 Preformatted BBS Message Uploads

MSGFIL switch enables ZMD to accept preformatted message file uploads. Files uploaded with the 'RM' option will be forwarded to the drive/user defined at PRDRV and PRUSR. This upload is then appended to your BBS message base. Your BBS software and BYE program must support this feature. MBBS, QBBS and PBBS all support this feature. PMSG/HMSG is now available on RI RCPM-RDOS.

3.5.2 Hiding New Uploads Until Approved by Sysop

HIDEIT allows Sysops to keep all new regular uploads hidden from public viewing until reviewed and cleared. This way, new uploads will not appear in a DIRectory listing and cannot be viewed or even downloaded by ZMD. This feature is disabled when the WHEEL byte is ON or Private upload mode is enabled. Files that have been hidden will show up in a DIRectory listing when the WHEEL byte is set and a $S option is used to show SYSTEM files. Reference will be made to these files in the log and FOR text file listings if those features are enabled. You can use POWER or NSWEEP to set hidden files to $DIR.

3.5.3 Upload Routing to Multiple Drive/Users

ASKAREA switch set YES enables upload routing to multiple drive/user areas. With this enabled a caller is displayed a list of upload categories to choose from. When he enters his selection, ZMD will calculate the offset to the upload drive/user in TYPTBL and set the upload area based on his selection. This is done at the same time for both Regular and Private uploads). Upload routing is disabled when the WHEEL byte is set, in which case, normal uploads will go to the current drive/user area and private uploads will go to the drive/user equated at PRDRV and PRUSR.

3.5.4 Uploading to Specified Drive/User

SETAREA enabled forces all new uploads to the drive/user defined at **DRV** and **USR**. If the WHEEL byte is set, regular uploads will go to the current or specified drive/user. All private files uploaded with the 'RP' option will be sent to PRDRV and PRUSR regardless of WHEEL status.

3.5.5 Receiving Private Uploads

PRDRV and **PRUSR** are the drive/user area where ALL files sent to the Sysop with the ’RP’ option will go (unless ASKAREA is YES). This permits experimental files, replacement and/or proprietary programs to be sent to an area only accessible by the Sysop. This is also the drive and user area where message files are uploaded, if MSGFIL is set YES. If ASKAREA is YES, ’RP’ uploads will go here only if the WHEEL is set. If MSGDESC is YES, this is the drive and user area the FOR text file will be placed before appending it to the BBS system’s message base.

3.5.6 Crediting Upload Time to Time Left

CREDIT enabled causes ZMD to credit callers for the time they spend uploading non-private files each session. For example, a caller who spends 30 minutes sending an upload gets 30 minutes added to his TLOS. (You must set either CLOCK, RTC or TIMEON to YES to use this feature).

3.5.7 Receiving .COM Files Securely

NOCOMR tells ZMD to rename .COM files to .OBJ and .PRL to .OBP on receive. This feature is also disabled when the WHEEL byte is set.

3.6 Upload Descriptions

This section has to do with upload descriptions. If you do not intend on implementing upload descriptions, set DESCRIB and MSGDESC to NO. The rest of these values are then ignored. If using descriptions, set ONLY one of these to YES, not both.

3.6.1 Forwarding Descriptions to BBS Message Base

MSGDESC should be set YES if your your BBS system supports message uploads, and you prefer upload descriptions to be placed in your BBS message base (set DESCRIB NO). MBBS users need to install MFMSG.COM with the MBBSINIT program. Then set your BYE program to know about message file uploads by setting the MSGFIL option in BYE/MBYE to YES. If set YES, ZMD will produce a FOR text file when writing upload descriptions. This FOR file will go to the drive and user area equated at PRDRV and PRUSR just before being appended to your BBS system's message base.

3.6.2 Forwarding Description to FOR Text File

DESCRIB switch should be YES if you want descriptions to be appended to the current FOR file where they can be viewed by callers with the ZFORP utility. Sysop can add new descriptions with the ZFORS utility. Uploads sent to the Sysop private upload area will not require descriptions, nor will files uploaded with the 'RW' option - user must be a privileged user (bit 7 in ACCESS byte set) or have WHEEL access and PUPOPT must be set YES to use the 'RW' option.

3.6.3 Description Filename

FORNAM/DRIVE/USER is the drive/user and filename of the FOR description text file. This filename must be 11 bytes padded with spaces. If using with DESCRIB set YES, you must indicate what drive/user you want the 'FOR' file to be placed. Drive/user area is automatically changed to PRDRV and PRUSR if descriptions are to be forwarded to the BBS message base

3.6.4 Upload Description Header Information

If your configuration includes DESCRIB set to YES, you’ll have to tell ZMD what information you want included in the first line of each description. Code is included in all ZMD programs to place all (any) information in the upload description header. The following diagram illustrates a full implementation of DESCRIB:

```
-----
ZMD150.LBR - Communications         (C3:)     Rcvd: 09/29/88
                 /                    /              /
         _______/             _______/         _____/
  ASKIND                INCLDU           DSTAMP
```

3.6.5 Including File Descriptors

ASKIND switch enabled causes ZMD programs to ask for the category of the upload(s) and write it into the upload description header. If you set this to YES, make sure you set MAXTYP to the highest letter choice you wish to support and edit the text at KNDTBL up to and including your MAXTYP setting. (Used only with DESCRIB).

3.6.6 Including Upload Drive/User Area

INCLDU enabled will include the drive/user area of the uploaded file into the upload description header. (Used only with DESCRIB).

3.6.7 Datestamping the Description Header

DSTAMP enabled will include the date the upload was received into the upload description header. (NO if no clock) (Used only with DESCRIB).

3.6.8 Overriding Description requests

PUPOPT allows descriptions to be disabled when "RW" is used on the ZMD command line (i.e. **ZMD RW FILE.EXT**). This command may only be used by those considered "priviledged" users on your system or WHEEL users. Uploads of this type will be tagged in the ZMD.LOG file as private, so as not to display with the NEW command. (See ACCESS equate description above for information on detecting ’priviledged’ users).

3.6.9 Automatic Word Wrap

WRAP is set to the column position where word wrap will occur. If using MSGDESC and have problems with an 'Invalid format' error from MFMSG.COM, try setting WRAP to something smaller, like 62 or 63. (Word wrap can be disabled by the user with ^W during description entry. Enter 72 here to disable WRAP completely).

3.7 Download Configurations

ACCMAP is a bit mapped flag register 1 byte in length. It contains 8 flag bits which enable/disable the filename restrictions outlined below. The restrictions always pertain to the file being considered for transfer. Enabling any of these options causes ZMD to look at the high bit of the byte position indicated below (F1=filename byte 1, T2=file type byte 2, etc). These restrictions are always bypassed when using ZCPR and the WHEEL is set.

3.7.1 Restricting .LBR and Single F1 Tagged Files

TAGFIL switch is enabled if you want to restrict callers from downloading certain files, such as very large overlay libraries, game libraries, etc. In most cases, remaining time left on system would be sufficient for restricting downloads. However, with bit 7 of ACCMAP set to 1, ZMD will check the high bit of filename byte 1 and if this is set, the file may not be downloaded. If the file is a ARK/ARC/LBR file, individual members may be downloaded however. This restriction is bypassed if the WHEEL byte is set.

ACCMAP Switch: **1**0000000
Filename Byte: **F**ILENAME.EXT

3.7.2 Disallowing .COM Downloads

NOCOMS should be enabled if you do not want callers to be able to download *.COM files. Most secure systems will enable this restriction. This feature is bypassed when WHEEL byte is set.

ACCMAP Switch: 0000**1**000
Filename Byte: FILENAME.**EXT**

3.7.3 Disallowing .??# Downloads

NOLBS is enabled for those systems which use 'labels' in the third file extent byte of system files to restrict public access to them. ZMD will check T3 byte for a '#' characters. Upon a match, the download is denied. This feature is bypassed when the WHEEL byte is set.

ACCMAP Switch: 000**1**0000
Filename Byte: FILENAME.EX**T**

3.7.4 Disallowing F2 Tagged $SYStem File Downloads

NOSYS enabled tells ZMD to ignore all files with the high bit set in filename byte T2. These files are considered as hidden $SYStem files by CP/M and can be treated the same way by ZMD. This feature is bypassed when the WHEEL byte is set.

ACCMAP Switch: 00**1**00000
Filename Byte: FILENAME.E**X**T

3.7.5 Sending F3 Tagged Files Regardless of Access

DWNTAG if enabled allows any file with the high bit set in filename byte 3 to be sent regardless of the callers access. This comes in very handy for closed systems requiring users to download applications, system information files, BBS lists, etc.

ACCMAP Switch: 0**1**000000
Filename Byte: FI**L**ENAME.EXT

3.7.6 Special Sysop Downloads to Selected Caller

SPDRV and **SPUSR** contain the drive/user area for downloading private 'SP' files from Sysop. This permits you to put a special 'non-public' file in this area, then leave a private note to the person it is intended for mentioning the name of the file and how to download it. Although anybody 'could' download that program, they don't know what (if any) files are there. A high degree of security exists, while the Sysop still has the ability to make special files available. Thus any person can be a temporary 'privileged user'.

NOTE: A breach of security exists if SPUSR is not defined as a higher user area than the maximum allowable public user area.

3.8 Logkeeping

A clock is not necessary for this logkeeping features.

3.8.1 Making ZMD.LOG File

LOGCAL enables the log keeping routines in ZMD. New uploads will be added to the current ZMD.LOG file. If no log file exists, one will be created. All file transfers are logged. You can then use ZNEWP.COM to show listings of recent uploads or ZNEWS to add log entries.

3.8.2 European Date Format (DD/MM/YY)

EDATE causes ZMD and utilities to show date in dd/mm/yy format instead of mm/dd/yy format.

3.8.3 Transfer Log Drive/User/Filename

LOGNAM/LOGDRV/LOGUSR is the drive/user and filename of the ZMD.LOG file transfer log. This filename must be 11 bytes padded with spaces. If using with LOGCAL set YES, you must indicate what drive/user you want the ’ZMD.LOG’ file to be placed.

3.8.4 LASTCALR Drive/User

LASTDRV/LASTUSR/LCNAME is the drive/user of your BBS or BYE program’s LASTCALR.??? file. This filename must be 11 bytes padded with spaces. If using with LOGCAL set YES, you must indicate what drive/user ZMD can find the LASTCALR.??? file. LCNAME should be set to the column position of the caller’s name in the LASTCALR.??? file. (0 for PBBS, 11 for MBBS).

3.8.5 Counting Files Transfers for BBS Software

LGLDS set YES enables session upload and download counting. ZMD will count the number of up/downloads for each logon. Your BBS program then can check **UPLDS** and **DNLDS** counter bytes when a user logs out and update either the user’s file or a file for this purpose. You can either modify your BBS entry program to check the LASTCALR file before updating and then update (risky), or make a separate program that BYE calls when logging off a user (preferred). (YES for PBBS). Don’t forget to initialize UPLDS and DNLDS counter bytes to 0 from your BBS program when somebody logs in.

NOTE: Clear the UPLD and DNLD bytes ONLY when a user logs in, not when he re-enters the BBS program from CP/M.

APPENDICES

A.1 Modem Interface

A.1.1 Creating ZMD Modem Input/Output Overlays

All port/modem overlays are allowed 128 bytes between 580H and 5FFH. This area is always contained in the ZMDHDR.Z80 configuration table and included during the assembly process. The first 27 bytes of this overlay must contain 8 JP instructions in the following order:

Routine	Purpose	Entry	Exit
CONOUT	**Local console output (BIOS)**	**A=char**	-----
INIT	**Initialization**	-----	-----
UNINIT	**Uninitialization**	-----	-----
SNDCHR	**Send character POP AF gets->**	**A=char**	-----
CARCHK	**Carrier check**	-----	**Z=Carrier**
GETCHR	**Receive a character**	-----	**Char in A**
RCVRDY	**Check receive ready**	-----	**Z=char**
SNDRDY	**Check send ready**	-----	**Z=ready**

A.1.2 Installing Your Modem I/O Overlays

ZMD must have access to your modem for obvious reasons. It needs to send data, receive data, perform error checking, monitor carrier and give control back to your BYE program when carrier is lost. It has to know when the modem is ready to send another character and when one has been received. If your system uses extended BDOS calls to access your BYE programs port/modem routines, you can install ZMD and utilities without regard to this section. Otherwise, follow these steps:

1) Find an overlay from ZMDOVLn.LBR that best fits your modem/port requirements. You may have to create one for your system if one doesn’t exist already. Some standard formats are included (see -OVERLAY.LST in the ZMDOVLn.LBR).

2) Edit it with your favorite wordprocessor as needed. Make sure you local console output address (BIOS) can be calculated using standard CP/M methods (i.e. (JP COLDBOOT+9) If your BIOS JP TABLE has been altered, you will have to provide this address for ZMD at label CONOUT. In most cases, ZMD will be able to calculate this address for you. If you will need to initialize anything at program start up, include your custom routine at label INIT and your uninitialize routine at UNINIT. INIT is called at program startup, and of course UNINIT is called at program exit.

3) Assemble with M80, or SLRMAC or other Z80 compatible assembler to produce ZMxx-n.HEX

4) Use MLOAD (included) to load ZMxx-n.HEX over ZMD.COM like this:

```
A0>MLOAD ZMD=ZMD.COM,ZMxx-n
```

ZINSTL will now recognize your modem overlay. You can also use DDT to patch your overlay in. Make sure it starts at 580H and ends by 5FFH (128 bytes).

NOTE: If ZMD attempts to set the current user area to 241 and 77 is returned in register A, BYE's extended BDOS calls will be used for modem I/O.

B.1 RTC Clock/Date Reader Code

A few of ZMD's nice features are dependent upon access to a Real Time Clock. MBYE and BYE5 users who have their clock and date reader code installed need only set CLOCK to YES and leave RTC set NO. If on the other hand your BYE does not read your system clock, you will have to insert your clock and date reader code at label RTCTIM in ZMDHDR.Z80. With RTC set YES, all ZMD routines have access to the following binary values (poked by your clock insert):

Address	Length	Range
MONTH	**1 byte**	**1-12**
DAY	**1 byte**	**1-31**
YEAR	**1 byte**	**0-99**
HOUR	**1 byte**	**0-23**
MINUTE	**1 byte**	**0-59**

Your insert must start at 4FEH and end by 57FH (130 bytes).

C.1 File Descriptors/Categories

This table defines the text to be included in upload description headers (DESCRIB and ASKIND) and/or defines categories for uploading to multiple drive/user areas (If ASKAREA). Change as desired, if this list is not suitable. Do NOT remove any of the text at KNDTBL. Simply edit the category text below up to/including your MAXTYP setting. **MAXTYP** below must be set to whatever letter your maximum choice will be.

```
MAXTYP:    DB      ’W’     ; Highest category you will support.

KNDTBL:

DB   ’  A) - CP/M Utility          ’,CR,LF
DB   ’  B) - CP/M Application      ’,CR,LF
DB   ’  C) - CP/M Game             ’,CR,LF
DB   ’  D) - Wordprocessing        ’,CR,LF
DB   ’  E) - Text & Information    ’,CR,LF
DB   ’  F) - Printer Related       ’,CR,LF
DB   ’  G) - Communications - IMP  ’,CR,LF
DB   ’  H) - Communications - MEX  ’,CR,LF
DB   ’  I) - Communications - Other’,CR,LF
DB   ’  J) - RCP/M Software        ’,CR,LF
DB   ’  K) - BBS List              ’,CR,LF
DB   ’  L) - ZCPR1/2/3             ’,CR,LF
DB   ’  M) - Pascal Utility/Source ’,CR,LF
DB   ’  N) - dBase Utility/Source  ’,CR,LF
DB   ’  O) - Basic Utility/Source  ’,CR,LF
DB   ’  P) - ’’Other’’ Language      ’,CR,LF
DB   ’  Q) - EPSON Specific        ’,CR,LF
DB   ’  R) - ZMD Support           ’,CR,LF
DB   ’  S) - IBM Utility           ’,CR,LF
DB   ’  T) - IBM Application       ’,CR,LF
DB   ’  U) - IBM Game              ’,CR,LF
DB   ’  V) - IBM Communications    ’,CR,LF
DB   ’  W) - Mixed Batch/Misc      ’,CR,LF
DB   ’  X) - <<<< Not Defined! >>>>’,CR,LF
DB   ’  Y) - <<<< Not Defined! >>>>’,CR,LF
DB   ’  Z) - <<<< Not Defined! >>>>’,CR,LF
DB   0             ; leave the table terminator alone.
```

NOTE: Make sure you leave all the categories above EXACTLY 31 bytes long (29 bytes of text plus the CR,LF equals 31) or you will have problems with the double column formatting routines.

D.1 Upload Routing Table

If you decided to enable upload routing to multiple drive users, you will have to set the following table for your own requirements. Edit all areas at TYPTBL up to and including your MAXTYP seting to match the message text in KNDTBL above. Note that PRIVATE uploads may be sent to a different drive as well as a different user area. Each entry is expressed as ’drive letter’,user area. Simply set MAXTYP to the highest letter choice supported. (Do NOT comment out any of these storage bytes).

```
                _________
  NOTE:        /    A    \  <--- Corresponds to category ’A’
              ’A’,1,’B’,15,
                \ /   \ /
 Normal upload --+     |
 Private upload -------+
```

```
TYPTBL:
       _________     _________     _________     _________
      /    A    \   /    B    \   /    C    \   /    D    \
DB   ’A’,1,’B’,15, ’A’,2,’B’,15, ’B’,2,’B’,15, ’B’,4,’B’,15
       _________                   _________
      /    E    \                 /    Z    \
DB   ’B’,3,’B’,15, ... thru ...  ’B’,7,’C’,12
```