
3 File Transfer Information

This section contains information for users of KERMIT32 and VMS C-KERMIT. It is not a tutorial on either of those programs.

Information is also included on using PC-based communication programs to exchange files with a BBS or other information system (other than Internet). PC topics discussed include file compression utilities such as PKZIP.

Within this section, references to "VMS" pertain to OpenVMS-VAX and VAX/VMS only. However, most of this will also be true for OpenVMS-AXP.

3.1 File Transfer Overview

One of the primary attractions of BBSes and other information services is the ability to transfer files from that system to your local system. These files can contain text, data, programs, spreadsheets, graphic images, or just about anything.

There are two basic "modes" of file transfer: individual and batch.

Individual transfer mode is used for file transfer protocols such as XMODEM and KERMIT. It is generally used for situations where the programs performing the file transfer do not exchange information about the file such as its name, size, date, etc.

Batch transfer mode is used for protocols such as YMODEM-B and ZMODEM, and other protocols which do allow for the programs performing the file transfer to exchange information about the file such as its name, size, date, attributes, etc.

Files are usually prepared for transfer between systems by first grouping them into "archives", similar to VMS BACKUP savesets. Usually, the programs which process these archives also provide for compression of the files while they are being "packed" into the archive. On VMS, the LIBRARIAN provides for libraries to be compressed into DCX data-reduced format. On PCs and some UNIX systems, programs such as PKZIP are used.

Compression allows more data to be transferred in a shorter time.

Many file transfer protocols also include a degree of compression. Thus, even when the file being transferred is not already compressed, some performance improvement can still be realized during the transfer.

3.2 File Transfer To/From VMS

In order to transfer files between your system and another without a DECnet connection or other networked protocol, about the only method available is an interaction between two programs.

To transfer "binary" files (programs, spreadsheets, etc.), you need the ability to copy without regard to the data being transferred so long as the target file has the right attributes.

The next two sections describe two ways to transfer files to/from VMS Systems.

3.2.1 File Transfer To/From VMS - ASCII

To transfer ASCII (text) files and data between systems, there are a couple of options.

When you're transferring text only, sometimes it is possible to establish a serial link between two systems and COPY from one while CREATE-ing on the other. This is not very clean, but it does work. You must make sure that TTSYNC and HOSTSYNC are set for both ports.

Another method is to connect a modem to an async. serial port (a Q-bus, BI-bus or XMI-bus mux. such as a port on a DHV-11, DMB-32, etc. or a port on a terminal server) and use the SET HOST/DTE command with the /LOG qualifier. /LOG allows you to "capture" the session to a log file. You can then use your favorite text editor to extract the data from the log file. This method can be used with the SYS\$COMMAND-proprietary "HEX/ASCII" download protocol to receive VMS C-Kermit, if you don't already have it, and your system doesn't have any other means of transferring VMS executables from a BBS or information service. Here again, TTSYNC and HOSTSYNC are going to be critical.

See the appendices for a complete description of this method of downloading. It will DEFINITELY be worth your time!

3.2.2 File Transfer To/From VMS - Kermit

When transferring files between your VMS system and a BBS system such as SYS\$COMMAND, or any other BBS, sometimes KERMIT is about the only choice available.

We won't get into the whole history of KERMIT here. It is enough to say that it is available, both here on SYS\$COMMAND, on other BBSes and information services (check out the VAX+ Forum on Compuserve) as well as from Columbia University. KERMIT is Public Domain software, which means that it is prohibited to sell KERMIT for profit. You could call it a "freebie", except for media costs, postage, etc.

KERMIT32, the "old" KERMIT for VAX/VMS, is still around. It does the job, but leaves something to be desired. The data packets it exchanges with its partner are small, so it can take quite a while to transfer a large file using KERMIT32.

C-Kermit, the "latest-and-greatest" uses a much larger packet size and data compression. Many technical folks (this author included) hold that C-Kermit achieves data transfer rates comparable to those possible using ZMODEM. C-KERMIT for VMS is also smart enough to understand VMS file attributes. RMS files can be transferred between VMS systems using C-Kermit while preserving their attributes.

3.3 Kermit - Thumbnail Sketch

KERMIT is what you might call an "original" client/server system.

File transfer between two systems requires two KERMITs: one running on each system. One KERMIT is set into "server" mode and receives and processes commands from the other KERMIT. The KERMIT that's in server mode does the work while you command it locally from your keyboard or command procedure. The "client" KERMIT is in control of the file transfer operation.

To exchange files with The SYS\$COMMAND BBS, or any other, you can use KERMIT locally in much the same way as you would use SET HOST/DTE. You invoke KERMIT, tell it which port the modem is connected to, then tell it to connect you with that port. KERMIT will then accept input from your keyboard and transmit those keystrokes, character by character, via the modem port to the modem and/or remote system.

When using SET HOST/DTE, you type CTRL+\ to exit back to DCL. CTRL+\ is DTEPAD's "escape character".

KERMIT also uses what's called an "escape character" (not always CTRL+[, ASCII 27) to exit the "connect" mode and again accept commands from you locally. When you type that character, KERMIT leaves the connection alive, but returns to command mode. When you're ready to resume the connection, you will return to your remote session exactly where you left off. You can usually leave the escape character defined as it is, or, while in command mode, specify a different "escape" character such as CTRL+\ for consistency with SET HOST/DTE (DTEPAD) and SET HOST/LAT (LTPAD).

Unlike PC-based communication programs, KERMIT/32 does not provide automatic dialing, dialing directories, choices of terminal emulation, etc. KERMIT/32 simply provides a means of transferring data between systems. C-Kermit, however, does provide more of the "user friendly" functions including a dialing directory or "phone book" and initialization files.

3.3.1 How to Get VMS Kermit From The SYS\$COMMAND BBS

Right now, you're probably saying, "...but I don't have Kermit or C-Kermit! How can I get it?"

The SYS\$COMMAND BBS has anticipated your question and has an answer. Maybe not what you're hoping for, but it is an answer.

For the SOLE purpose of initially acquiring C-Kermit, Zmodem, etc., The staff of The SYS\$COMMAND BBS has devised a "protocol" called "HEX/ASCII". It's a method of converting a VMS .EXE, .ZIP or librarian (.%LB) file to an ASCII (printable) form of hexadecimal data. This can be "captured" by using the /LOG qualifier of the SET HOST/DTE command. You can use this same method method to get the DEHEX.COM procedure (see the "[R]ead a text file" command on the File menu). Then, you can log off of the BBS, type CTRL+\ to return to DCL, and use your favorite editor to "cut and paste" the log file (SETHOST.LOG) into DEHEX.COM and the hexadecimal form of the Kermit program.

DEHEX.COM will convert the hexadecimal form of the file back to a .EXE. However, DEHEX.COM IS a DCL procedure. On small VAXes, it can take some hours to complete. Still, this is better than the alternative (nothing!).

For a complete description of the HEX/ASCII download and DEHEX.COM processes, refer to the Appendices of this manual.

Also, because HEX/ASCII is not 100% trouble-free, we also carry C-Kermit in "hexified" forms:

For OpenVMS-VAX, CKVKER.DHX (use DEHEX.COM to restore) and CKVKER.HEX (use VMSDEH.MAR to restore). For PC users, these are available as a PKZIP archive, CKVK-PC.ZIP. You can PKUNZIP them, then transfer them to your OpenVMS system by any means you prefer. The archive includes both DEHEX.COM and VMSDEH.MAR, with instructions on how to assemble and link VMSDEH.

For OpenVMS-Alpha (previously known as OpenVMS-AXP), CKAKER.DHX (use DEHEX.COM to restore) and CKAKER.HEX (use VMSDEH.MAR to restore). For PC users, these are available as a PKZIP archive, CKAK-PC.ZIP. You can PKUNZIP them, then transfer them to your OpenVMS system by any means you prefer. The archive includes both DEHEX.COM and VMSDEH.MAR, with instructions on how to compile and link VMSDEH. (On OpenVMS-Alpha, MACRO32 is actually a 3GL. It converts complex VAX instructions (CISC) to Alpha instructions (RISC).)

3.4 Using VMS Kermit To Access The SYS\$COMMAND BBS

This discussion assumes that you have KERMIT on your system. See the Appendices and the previous section for information on how to get Kermit for VMS.

To access The SYS\$COMMAND BBS, or any other, using KERMIT for VMS, the first step is, of course, to select a port on your system to which a modem is connected. This can be a hardwired port such as on a DMB32, DHV11 or other async. serial mux., or it can be a port on a terminal server. Be sure to set the required characteristics for that modem on that port using SET TERMINAL/PERMANENT.

The next step is to ALLOCATE the modem port to your process and run KERMIT. If you intend to perform any file transfers, especially binary files such as .TLBs or .ZIPs, tell KERMIT:

```
SET FILE TYPE BINARY
```

for C-Kermit or

```
SET FILE TYPE FIXED 512
```

for KERMIT32.

Then, you can tell KERMIT to CONNECT you to the modem port. Once it does, it will tell you what the current "escape" character is. Typing that character will cause KERMIT to return to command mode while keeping your remote session alive.

Once KERMIT connects you to the modem port, what you type will be sent to the modem. From here, you can interact with the modem to accomplish dialing up the

remote system. When the modem connects, what you type on your keyboard will be sent via the modem to the remote system.

While connected to the BBS, then, you can set up for a file download. When setting up the download, be sure to specify that you will be using the KERMIT protocol.

When the BBS tells you to begin your download procedure, type KERMIT's "escape" character to get back to KERMIT's command prompt. Then, simply tell it to RECEIVE. Your local KERMIT will tell the remote KERMIT (the "server") that it is ready to receive the file. The two KERMITs will then begin to interact and transfer the file's contents to a new file by the same name on your local system. When the file transfer is complete, KERMIT lets you know. You can then CONNECT to the modem port once again to resume your session, if KERMIT doesn't do that for you (some KERMITs might).

3.5 Packing And Unpacking VMS Files In Text Libraries (.TLB)

As a de-facto standard on BBSes and CompuServe, groups of VMS files are typically packed into a text library (.TLB) file. This process also compresses the library to reduce the file transfer time as much as possible. See the on-line HELP on your VMS system or the LIBRARIAN Reference manual for more information on libraries and library file compression.

Utilities for making and unpacking text libraries can be found on The SYS\$COMMAND BBS in three forms:

MAKE_TLB.COM	(download as text)
MAKE_TLB.TLB	(download as binary)
MAKE_TLB.ZIP	(download as binary)

All three forms contain MAKE_TLB.COM. MAKE_TLB will process all of the files in your current default directory (including itself) and insert them into a text library. In the process of doing so, it reads itself as data and produces from that another procedure called EXTRACT_TLB.COM, which unpacks a text library into separate files in your current default directory. Both MAKE_TLB.TLB and MAKE_TLB.ZIP contain both MAKE_TLB.COM and EXTRACT_TLB.COM as separate files.

Any .TLB file that you download from The SYS\$COMMAND BBS should contain EXTRACT_TLB.COM. If you don't already have a copy of this on your system, you can extract it from the library using a command in this form:

```
$ LIBR/TEXT/EXTR="EXTRACT_TLB^COM"/OUT=EXTRACT_TLB.COM filespec
```

...where "filespec" is the name of the text library. This will produce EXTRACT_TLB.COM in your current default directory. You can then unpack the rest of the files in the library with the following command form:

```
$ COPY ddcu:[dir]EXTRACT_TLB.COM []  
$ @EXTRACT_TLB filespec
```

All of the files in the library will be extracted into your current default directory. Executable image (.EXE) files will be restored to the correct format for executables. It's a good idea to keep a copy of EXTRACT_TLB.COM around so you don't have to extract it every time:

```
$ LIBR/TEXT/EXTR="EXTRACT_TLB^COM"/OUT=EXTRACT_TLB.COM filespec
$ COPY EXTRACT_TLB.COM ddcu:[dir]
```

3.6 PKZIP and VMS ZIP Archive (.ZIP) Files

Files on PCs are usually archived using the PKZIP archiving and compression utility from PKware, Inc. The files in the archive are usually intended for use on a PC, but may also be DCL command procedures, documentation files, or just about anything that can be transferred back to a VMS system fairly easily.

There is even a version of this archiving protocol for VMS called ZIP and UNZIP. These are public domain programs and can be found on The SYS\$COMMAND BBS as ZIPUNZIP.TLB. They can also be found on other VMS-oriented BBSes such as SYS\$OUTPUT and in the VAX+ Forum on CompuServe.

3.6.1 Unpacking Files From PKZIP Archive (.ZIP) Files (MS-DOS)

To unpack a .ZIP file, you need the PKUNZIP program. To get PKUNZIP, download the PKZ204G.EXE file from The SYS\$COMMAND BBS, then refer to section 3.6.3 which deals with self-extracting archives for information on how to extract the PKZIP and PKUNZIP programs.

Before unpacking a .ZIP archive, you should first examine its contents to see if it contains an entire directory tree or just a group of files. To do this, use a command of the form:

```
C:\> PKUNZIP -v filespec
```

...where "filespec" is the name of the archive. The .ZIP extension is optional for PKUNZIP V2.x. PKUNZIP will display a list of the files contained in the archive showing the file's actual size, how (or if) it was compressed, its compressed size, the resulting compression ratio, the date of the file, a checksum, the file's attributes and its name.ext. You can then unpack the archive like so:

```
C:\> PKUNZIP filespec destination
```

...where "filespec" is the name of the archive and "destination" is the path specification for where you want the files to be placed. The destination can include a drive or a directory name. If the destination is omitted, the files in the archive will be placed in your current directory on your current disk.

If the names of the files in the archive include a path specification, then you should use the "-d" option when you unpack the archive to restore the original directory structure. In this case, the command to unpack the archive would look like this:

```
C:\> PKUNZIP -d filespec destination
```

The "-d" option MUST be entered in lower case. If the destination is omitted, the directory structure will be rebuilt starting from your current directory on your current disk.

See the PKUNZIP documentation supplied with the program for more information on PKUNZIP, or use this command to get help on-line:

```
C:\> PKUNZIP/?
```

3.6.2 Unpacking Files From PKZIP Archive (.ZIP) Files Using VMS UNZIP

PKZIP archives can be transferred to VMS. You can then use the UNZIP program described in section 3.6 to extract files for use on your VMS system.

UNZIP will extract all files into VMS files with the STREAM_LF file format. As a result, text files, when viewed with an editor, will have a carriage return at the end of each line. Other files, such as programs and object files, will be of very little use on VMS. Graphics files, PostScript files and spreadsheet files may be usable, with some further massaging of their contents. Experiment with this to see how useful this is for your application.

To extract text files from PKZIP archives using VMS UNZIP, use the "-A" option of VMS UNZIP. This will produce a more typical sequential file with CARRIAGE_RETURN record attributes and a variable record length:

```
$ UNZIP -A archive_name
```

Most "binary" MS-DOS and Windows files will get royally scrambled if you try to unpack them on VMS using the UNZIP program.

3.6.3 Unpacking Files From A Self-Extracting PKZIP Archive (.EXE) File (MS-DOS)

When you see a .EXE file on The SYS\$COMMAND BBS, it will almost never be a VMS executable image. It will almost always be a self-extracting archive file. Exceptions will be noted in the file descriptions.

A self-extracting archive file is an archive file that has been appended to a smaller version of the PKUNZIP program. While there are other compression utilities out there, PKZIP and PKUNZIP are the most common. We will only discuss self-extracting PKZIP archives here, so be aware that the command options and syntax for other self-extracting archives may be different.

Self-extracting PKZIP archives behave in much the same way as PKUNZIP, they just have fewer options. The "-v" ("view") and "-d" ("restore directory") options are available, however, as are some of the others. See the previous discussion in section 3.6.1 for more information on how to use self-extracting PKZIP archives. Simply exclude "PKUNZIP" from your commands and use the archive name and destination. For example, to unpack PKZ204G.EXE onto a diskette in drive A, the command might look like this:

```
C:\> PKZ204G A:
```

This will cause the files in the archive to be unpacked onto the diskette in drive A.

3.6.4 Packing PC Files Into PKZIP Archive (.ZIP) Files (MS-DOS)

The PKZIP program will compress files into an "archive" file. It can include the relative path of each file so that an entire directory tree, or portion of one, can be restored using PKUNZIP.

To pack all the files in your current directory into an archive, the following command will do it for you:

```
C:\> PKZIP archive_name filespec
```

...where "archive_name" is the name you give to the archive, include the .ZIP extension, and "filespec" is the name of the file(s) to be compressed into the archive. "filespec" can be the name of a single file or it can contain any valid DOS wildcard expression.

Some of PKZIP's options include:

- eX Use maximum compression
- rP Recurse subdirectories and include path information.

For example, to archive an entire directory structure and all the files in it, you could use the following command:

```
C:\> PKZIP -eX -rP archive_name *.*
```

This would cause PKZIP to compress into the archive every file in the current directory and every directory below it and include the path for each file. The structure could then be restored using PKUNZIP with the "-d" option.

For more complete information, see the documentation that comes in the PKZ204G.EXE self-extracting archive.

3.6.5 Making PKZIP Archive (.ZIP) Files Self-Extracting (.EXE) (MS-DOS)

Included with PKZIP and PKUNZIP is a utility called ZIP2EXE which makes self-extracting archives from .ZIP archives. The command is very simple:

```
C:\> ZIP2EXE archive_name new_archive_name
```

If "new_archive_name" is omitted, it defaults to "archive_name.EXE". This creates a new file containing the self-extraction code. The archive contents are then appended to it.

Sometimes it is more convenient to upload an archive which is self-extracting, due to disk space constraints or other considerations.

Large archives, once made self-extracting, may no longer fit on a single diskette, however. It may be more convenient in some cases to keep the archive and PKUNZIP separate. Some third-party archive utilities that can read .ZIP files do not handle self-extracting archives. VMS ZIP sometimes gets confused by self-extracting archives.

Making an archive self-extracting also requires more transfer time since the file is made larger by adding the self-extraction code. Self-extracting archives result in more disk

space consumption also since the self-extraction code is duplicated in every self-extracting archive.

When ever possible, we recommend using .ZIP archives instead of self-extracting archives since the .ZIP files are smaller (which saves disk space) and take less time to transfer.

3.6.6 Packing VMS Files Into VMS ZIP Archive (.ZIP) Files

VMS ZIP has the ability to preserve the ENTIRE VMS file specification, including the version number when VMS files are packed into .ZIP archives using VMS ZIP. VMS ZIP will also preserve the RMS attributes of the files it packs into an archive file. These features are controlled using command line options.

To use VMS ZIP, establish a "foreign" command, like so:

```
$ ZIP ::= $ddcu:[dir]ZIP.EXE
```

...where "ddcu:" is the disk where the ZIP.EXE file resides and "[dir]" is the directory (path) on that disk where the ZIP.EXE file can be found.

Here are some key command line options for use with VMS ZIP:

l thru 9

This controls how much compression is applied where "1" is the least and "9" is the most.

V

The "V" option causes VMS ZIP to store the RMS attributes of the file in the .ZIP archive so that when the file is extracted from the archive, it's original RMS attributes can be restored. Since "V" MUST be capitalized, the option string containing this option must be enclosed in double-quotes. This also prevents the filename and filetype extension from being truncated to 8 characters and three characters, respectively.

w

The "w" (lowercase "w") option causes VMS ZIP to store the version number along with the filename and filetype extension.

For example:

```
$ ZIP "-9Vw" MYZIP MYDISK:[MYDIR]MYFILE*.*
```

Note that the use of uppercase letters in an option string requires that the options be enclosed in quotes.

Issuing the following command will cause VMS ZIP to display its command line options:

```
$ ZIP -?
```

3.6.7 Unpacking Files From VMS ZIP Archive (.ZIP) Files Using VMS UNZIP

This process is essentially the same as for PKUNZIP, except that some of the command line options differ from PKUNZIP. Also, VMS UNZIP always restore files and directory

structures starting from your current default device and directory. VMS UNZIP does not support a destination specification.

To use VMS UNZIP, establish a "foreign" command, like so:

```
$ UNZIP ::= $ddcu:[dir]UNZIP.EXE
```

If VMS ZIP was told to store the RMS attributes of the file into the archive, UNZIP will restore the original attributes when it restores the file. If VMS ZIP was told to store the original version number into the archive, UNZIP will restore the original version number as well.

Note that UNZIP will issue advisory messages for every file it restores if the file was archived on a different version of VMS than the one on which you're running VMS UNZIP.

You should have at least V5.0 of VMS UNZIP in order to achieve the behavior described here.

If VMS ZIP was told to include path information for the files stored in the archive, VMS UNZIP will attempt to restore the original directory structure, just like PKUNZIP does when using the "-d" option. For VMS UNZIP, however, this is the DEFAULT behavior, not the optional behavior. To have VMS UNZIP restore all files to your current default directory, include "-j" in the UNZIP options.

For example, the following command will cause VMS UNZIP to restore all files in the archive to the current default disk/directory and restore the original version number:

```
$ UNZIP -JW archive_name
```

3.6.8 Self-Extracting (.EXE) Archives On VMS

Self-extracting archives on OpenVMS are produced by first creating the archive, then appending it to a copy of the "SFX" (Self-Extract) Program. Optionally, you can then use the "-A" option of VMS ZIP to adjust the offsets within the self-extracting archive to accommodate the executable code which precedes the archive itself.

The "SFX" (Self-Extract) programs for VAX and Alpha can be found on The SYS\$COMMAND BBS in the appropriate file areas: file area 3 for VAX (SFX_VAX.EXE), file area 5 for Alpha (SFX_AXP.EXE). Download them as BINARY.

Example of how to make a self-extracting archive:

```
$ ZIP "-9Vw" MYFILES *.*;*
$ COPY SFX_VAX.EXE,MYFILES.ZIP MYFILES.EXE
$ ZIP "-A" MYFILES.EXE
```

Note:

The -A option of ZIP may only be available in more recent versions (V2.1 and later). Use the "-h" option to view the available UNIX-style options, or "\$ ZIP/HELP" to display the DCL-style options in ZIP V2.1 and later.

To "unpack" the self extracting archive, simply RUN it.

The self-extract program supports only a limited set of command line options. So, it is usually best to make the archive using the appropriate options so that no options will be needed when the archive is unpacked. To use command-line options when you unpack a self-extracting archive, either create a "foreign" command for it, or place it in your DCL\$PATH (OpenVMS V6.2 and later); then, invoke it as needed.