

# CrossePAC User's Guide

Version 1.35  
October, 1994 - August, 1995

Digital Strategies, Inc.  
382 Springfield Avenue, Suite 308  
Summit, New Jersey 07901

B & C Solution  
Ole Piisvej 4, Box 54  
DK-3100 Hornbæk, Denmark



OVERVIEW.....	1
Technical support .....	1
Sales and marketing.....	1
QUICK TOUR .....	2
FUNCTION REFERENCE.....	3
COMMAND VERBS .....	4
OPTIONS.....	5
FILE LISTS AND WILDCARDS .....	9
SCRIPT FILES .....	9
CrossePAC READER .....	9
ARCHIVE FORMAT CONSIDERATIONS.....	10
File Transfer Considerations.....	10
DATA "PACKAGING" .....	11
INTERNET MAIL .....	11
EXAMPLE COMMANDS .....	12
ERROR MESSAGES.....	12
GCOS IMPLEMENTATION SPECIFICS.....	18
GCOS INSTALLATION .....	19
Installation from Diskette.....	19
Installation from Tape.....	19
TSS operation.....	19
Batch operation .....	20
DOS & OS/2 INSTALLATION.....	20



## OVERVIEW

CrossePAC is a multi-platform data compression utility that creates archive files (PAC files) consisting of one or more source files. Archives created by CrossePAC are specifically designed for portability across a wide range of hardware and software platforms. In order to improve portability and avoid possible communications conflicts during file transfer, CrossePAC is able to build archives in any of three formats: Text, ASCII, or Binary. In addition, each format can be written in either blocked (line oriented) or unblocked (stream) mode. File and path names are normalized when stored so that they can be reconstructed on a target system that differs from the originator.

Included in this User's Guide are instructions for using CrossePAC, some considerations for formatting and transferring files, and some examples of command lines. Installation instructions are included in the material sent with the software media. At a minimum, you should read the General Functions section of the User's Guide. This will give you a quick start in using CrossePAC.

Technical support for CrossePAC is available via electronic mail, telephone, and FAX.

### EUROPE

CompuServe Mail:	100115,1127
Internet Mail:	100115.1127@COMPUSERVE.COM
Telephone:	+45 49.70.47.90
FAX:	+45 49.70.47.80

### North America

CompuServe Mail:	>MHS:CPAC@DSINJ
Internet Mail:	CPAC@DSINJ.MHS.COMPUSERVE.COM
Telephone:	(405) 364-3701
FAX:	(405) 321-8577

Sales and marketing support for CrossePAC is available at the email addresses above and by telephone or FAX at the following numbers:

### EUROPE

Telephone:	+45 49.70.47.90
FAX:	+45 49.70.47.80

### North America

Telephone:	(908) 273-8983
FAX:	(908) 273-8860

## QUICK TOUR

CrossePAC is command line driven. Each CrossePAC command starts with the name of the program, CPAC, followed by a series of parameters. If you type CPAC without any parameters, you see the following short help screen:

```
Usage Syntax: CPAC <command> [<options>] <PACfile> <file>* -<file>*
```

```
<command>   = Add | eXtract | DELETE | List | ListMore | Replace |
              View | reFormat | Print | Help | ?
<options>   = Format={Text | Ascii | Binary}  Blksize=nn
              eXCludechars=<character list>  +/-Subdirectories
              +/-UsePath  +/-UseDevice  +/-Overwrite
              MessageLevel={All|Warnings|Errors|None}
              OutPath=<pathname>
              StorePath={Full|Recursed|None}
<PACfile>   = Name of PAC file to use. Wildcards not permitted
              Extension of .PAC will be assumed
<file>      = [<filespec> | IndeXfile=<filename>]
-<file>     = filename or wildcard to exclude
<character list> = List of characters or decimal equivalents
              specified as "\\nnn\" where nnn is the decimal value
              of the character
<filespec>  = Explicit file name or wildcard specification
<filename>  = Explicit file name
```

### CrossePAC Help Screen

The first parameter, <command>, is required and instructs CrossePAC as to what is to be done with the rest of the parameters. Commands include "Add" (adds files to a PAC file), "eXtract" (extracts files from a PAC file), etc. A full description of each command is provided in the Command Reference portion of the Detail Functions section in this document.

The <options> parameters are not required. In general, options control file processing (i.e. walking subdirectory trees). During compression, operations options also control the compression file format. A full description of each option is provided in the Options Reference of the Detail Functions section of this document.

When specifying <command> or <option> parameters, only the upper case letters are required. In the case of "DELETE", all letters are required, with "reFormat", only the "F" is required. Toggle <options> use a + or - in front of the option to turn it on or off respectively.

The <PACfile> name is required. This is the name of the compressed file which CrossePAC is to operate on. The <PACfile> is created or added to during compression commands and read during decompression operations. The extension name will always be ".PAC" when an extension is not specified. No other extension names can be used.

The <file> portion of the command line is required and determines what files are to be processed by CrossePAC. During compression this is the list of file names to include in the PAC file; during decompression this is the list of file names to extract. Multiple wildcard file specifications are fully supported by CrossePAC up to the limit of the command line buffer for the operating system (under DOS this is 127 characters) on which the utility is run.

## FUNCTION REFERENCE

The CrossePAC command line is composed of a command verb, an archive file name (PAC file), and various options and file lists. At a minimum, all CrossePAC functions require a command verb and an archive file name. Some operations also require a file list. The command verb is always the first parameter on the line. The first file name on the command line will be used as the archive file name. Options may appear anywhere on the command line after the command verb. File lists must follow the archive file name.

The components of the CrossePAC command line fall into four categories: Commands, Keyword Options, Toggle Options, and file specifications.

Commands are special words that tell CrossePAC what to do. The word below, for example, is a command.

ADD

Keyword options are words immediately followed by an equal sign and a value. In the example below 'BLOCKSIZE' is a keyword and '80' is its value.

BLOCKSIZE=80

Toggle options are words with a prefix of '+' or '-' to indicate that the option is ON (+) or OFF (-). In the examples below the OVERWRITE option is turned off and the SUBDIRECTORIES option is turned on.

-OVERWRITE  
+SUBDIRECTORIES

File specifications are path and file names or wildcards that are valid for the system on which CrossePAC is running. The examples in this manual will use the DOS file name syntax, however, in actual use you should use the file and path name syntax that is normal for your system. The examples below are valid file specifications.

C:\TOOLS\MAGIC.COM  
3RDQRTR.DOC  
..\ANNUAL.WKS  
\*.TXT  
\*.\*

In the reference below, all CrossePAC command verbs and options are shown in a mixture of upper and lower case characters. Commands and options are not case sensitive. The mixed case is used to indicate the abbreviated form of each option. You may abbreviate any command or option to the uppercase letters shown or you may include as many or as few of the lowercase characters as you wish. The following examples list both valid and invalid abbreviations for the command verb FREshen.

VALID

fre  
freshen  
fres  
freh  
frehn

INVALID

f  
fsh  
frsh  
fred

## COMMAND VERBS

- Add** Compresses and adds files specified on the end of the command line to a PAC file. For new PAC files, all files matching the specification are added to the PAC file. If the PAC file already exists, then the new files are appended onto the existing files. Any files in the existing PAC file which match the file specifications are replaced. This command requires a file list.
- Replace** Compresses and replaces files in an existing PAC file. Only files specified on the command line which already exist in the PAC file are affected. This allows you to update large PAC files with newer versions of the same files. Note that Replace does not perform a date compare or any other file compare logic! It is possible to replace a newer file already in a PAC file with an older file of the same name. This command requires a file list.
- FREshen** Compresses and replaces files in an existing PAC file when the file on disk is newer than the corresponding file in the archive. A file list is optional for this command. If a list is supplied, only those files matching the supplied list will be freshened. If no list is supplied then all files in the archive will be freshened.
- Update** The Update command is similar to the FREshen command in that it will replace existing files in the archive based on the date and time stamp, but it will also add new files that match the file specification given on the command line. A file list is required for this command.
- reFormat** Changes the file structure of the PAC file based on the options specified in the <options> parameters. If you create a PAC file which needs to be transmitted to a variety of locations, you may need to format the same PAC file in several different ways. For example, using this command you can take a Binary PAC file and change it to a Blocked Text file with no "\$" characters. When reFormatting a PAC file the files are not re-compressed and the file list at the end of the command line is ignored. All files remain in the PAC file, only the format of the PAC file changes. This command requires the use of the 'Format=' and/or 'Blksize=' options.
- List** Displays device, path, filename and compression information for files in the specified PAC file. A file list is optional for this command. If a file list is included only files which match the file list are displayed, otherwise all files are displayed.

<b>ListMore</b>	Works just like List except the display pauses at each full screen.
<b>View</b>	Displays the contents of files in a PAC file. This option is most useful for text-type data. The files are decompressed before they are displayed. If a file list is included then only matching files are displayed.
<b>Print</b>	Works just like View except the contents are sent to the standard printer device.
<b>eXtract</b>	Decompresses files from a PAC file which match the file list at the end of the command line. Files are placed in the current directory unless options are used to specify directory and/or device placement. The PAC file is not modified in any way during an eXtract command. A file list is optional for this command. If no file list is supplied, all files in the root directory of the archive are extracted. NOTE: For archives created with an embedded directory structure, it is necessary to either specify the directory structure in the file list or to use the +Subdirectories option to access files that are not located in the 'root' directory of the archive.
<b>DELETE</b>	Removes files from a PAC file which match the file list at the end of the command line. If files are DELETE'd from an archive file they can not be recovered other than by adding them back from disk via the Add command. This command requires a file list. NOTE: For archives created with an embedded directory structure, it is necessary to either specify the directory structure in the file list or to use the +Subdirectories option to access files that are not located in the 'root' directory of the archive.

When processing an existing archive it is important to note whether the archive contains an embedded directory structure. CrossePAC treats the embedded structure the same way a hierarchical structure is handled on disk. Unless a directory or path is specified or unless subdirectory processing is specifically requested, file operations are performed on the files in the "current directory". For CrossePAC the "current directory" of the archive is always the "root" or top-level directory. This means that a request to extract or otherwise process specific files in the archive will operate only on the files in the "root" directory unless a directory name is specified or the "+Subdirectories" option is used. The "List" command verb is an exception in that it will always list the entire contents of an archive including all subdirectories unless a specific file list is given on the command line. Thus, it is possible to determine the embedded directory structure of an archive in advance using "List".

By default, CrossePAC creates archives that include the directory structure that was specified with the file list and any structure that occurs as a result of "walking" the disk structure. Using the 'StorePath=' option, described below, can change the way directories are stored.

## OPTIONS

Options control how CrossePAC performs the command you have selected. Options are used to control the archive format, how files are processed, and other aspect of CrossePAC's operation. Appendix A describes the default settings for each option.

NOTE: The PAC file format options listed below are used to control the encoded structure and character set of an archive. None of these options limit in any way the data that is

contained in the files that make up the archive. You may choose any combination of format options without compromising the data in your files or the ability to extract an EXACT match of the original files.

**Format=** Specifies the format for the resulting PAC file during compression commands. Valid Format options are Text, ASCII, and Binary. The Format option controls the format of the archive file only, it has no effect on the type of data that may be contained in the archive.

**Text** This is the most portable PAC file format and should probably be used when creating archives that will be sent to mainframes, across Internet mail, or to systems of unknown type. Only printable ASCII characters are included in the PAC file. The Text format imposes an overhead of 23% on the size of the archive when compared to the Binary format.

**Ascii** The ASCII format uses the full 7-bit ASCII character set. This format is very efficient for archives that will be transferred over 7-bit data communications networks but where there are no other protocol or file content restrictions. The ASCII format imposes an overhead of 14% on the size of the archive when compared to the Binary format.

**Binary** The Binary format uses all 8 bits of each byte and results in the smallest possible archive size. This format is best suited for archives that will be transferred between similar system types, such as DOS, where file content and structure are not restricted. There is no size overhead for Binary format archives.

**Blksize=** Specifies the block size or line length for PAC files. Valid values range from 40 to the maximum text line for your system. Each block is ended with a CR and LF character. A block size of 0 results in an unblocked, or stream mode archive. NOTE: Specifying a block size for a binary or ASCII format archive will result in the CR and LF characters automatically being excluded since they will be used to delimit lines and can no longer represent data.

**eXCludechars=** Specifies characters which can not be used in the creation of PAC files. If a specific character or series of characters may cause problems with data communications hardware or software, these characters can be excluded from the PAC file.

To exclude a character which can not be typed use \### decimal notation where '###' is the decimal value of the character. For example, to exclude the Escape character, type XC=\027.

The maximum number of characters that can be excluded from each PAC file is as follows:

Text 45 characters  
ASCII 63 characters  
Binary 127 characters.

In actual practice, the eXcludechars= option should be used sparingly, since it will increase the size of the PAC file. In general, if a large number of characters need to be excluded you should consider changing the format of the PAC file. For example, if you are using the Binary format and find that a large number of characters must be excluded, switch to the ASCII format. Similarly, switch to Text from ASCII if there is a large number of conflicting characters.

#### **+/-Subdirectories**

Toggles searching of subdirectories during both compression and decompression. The default for this option is off. When off, all file specifications will be processed against the current directory or as specified in the file list at the end of the command line. When on (+S), all subdirectories under the current or specified directory will be searched for matching files.

The Subdirectories option is used for both compression and decompression commands. During compression, it is used to indicate if subdirectories on the disk should be scanned while looking for files to compress. During decompression, it is used to indicate if subdirectories in the PAC file should be scanned looking for files to decompress.

#### **+/-UseDevice**

Toggles the use of physical device names in the PAC file during expansion. This option has no effect during compression. The default for this option is off. When off, no device specifier is used during decompression. When on (+UD), the physical device name included in the PAC file is used during decompression to restore files to more than one physical device.

#### **+/-UsePath**

Toggles the use of path names during decompression. This option has no effect on compression commands. The default for UsePath is off (-UP) meaning that files are decompressed to the current directory regardless of what directory they were in when they were compressed. If UsePath is on (+UP), then the original directory structure used in the PAC file is recreated during decompression.

#### **+/-Overwrite**

Toggles the automatic overwriting of existing files during decompression. This option has no effect on compression commands. The default for Overwrite is off (-O) meaning CrossePAC will warn you before it overwrites a file already on the disk with a file from the PAC file. If Overwrite is on (+O) then existing files will be overwritten automatically.

#### **IndeXfile=**

Specifies the name of a file containing the file names to be processed. This file is structured as a standard text file with one file name listed per line. The IndeXfile= parameter can be used in place of or in addition to a list of file names specified on the command line. This option is particularly suited to situations where repetitive compression tasks must be performed.

#### **OutPath=**

Specifies a target directory into which files will be placed during extraction.

#### **StorePath=**

Specifies how file path names will be stored in the archive. The options available with StorePath are:

- |                 |  |
|-----------------|--|
| <b>None</b>     | Strip all path information from the file name before storing it in the archive. When multiple directories are processed use of this option may result in different files with duplicate names being included in the archive.   |
| <b>Full</b>     | Retain all path information in the archive for each file.  |
| <b>Recursed</b> | Include only that part of the path information that is a result of "walking" the directory hierarchy. This option is useful for creating an archive that retains all of the relative file structure information so that the structure can be easily re-created in a different location. This option is meaningful only when wildcard file specifications are used and the Subdirectories option is ON. |

#### **MessageLevel=**

This option can be used to control CrossePAC's output to the screen and may be useful when running CrossePAC as a component of another application. The options available with MessageLevel are:

- |                 |  |
|-----------------|--|
| <b>All</b>      | Display all messages. This is the default. |
| <b>Warnings</b> | Display only warning and error messages.   |
| <b>Errors</b>   | Display only error messages.               |
| <b>None</b>     | Do not display anything on the screen.     |

**-<file>** An explicit file name or wildcard preceded by a minus '-' will be interpreted as a file exclusion. File exclusions are applied collectively and act as a filter for all files processed. The location of file exclusions relative to other file names has no bearing on their effect.

**@<file>** An explicit file name preceded by a commercial at sign '@' will be interpreted as a script file. Script files can be used when the command sequence is too long for the standard command buffer of the host system or when tasks must be performed repetitively. Refer to the SCRIPT FILES section for more information on preparation and use of script files.

## **FILE LISTS AND WILDCARDS**

The file list parameters follow the name of the PAC file to be operated on or created and are used to tell CrossePAC which files are to be operated upon. File lists can be used to refer to external files on disk that are to be compressed and stored in the archive or they may refer to the files contained in the archive. The distinction of whether a file list refers to internal or external files is made by the command verb in use.

File names stored in the archive are always displayed using the syntax and structure that is valid for file names on the system where CrossePAC is running. This means that an archive that is built on one system

type, such as a mainframe, and then processed on another system type, such as DOS, will appear to contain files with names that conform to local system naming conventions.

A file list can be an explicit file name, a wildcard specification, an 'IndeXfile=' option, or a combination of these. File names and wildcards are system-dependant and will conform to local system conventions. Explicit file names and wildcard specifications can also be used as file exclusions.

## SCRIPT FILES

In order to facilitate operations that require very long command sequences and/or repetitive use, CrossePAC supports the use of script files. A script file is a standard text file containing one CrossePAC command, option, or file specification per line. The script file can be prepared with any editor that can create standard text files without formatting codes. Programmer's editors commonly produce this type of file and most word processors have an option to save a file in pure text form.

Script file processing is invoked by placing the name of the script file on the command line with a commercial at sign '@' prefix. CrossePAC processes elements on the command line until the script file parameter is reached and then "redirects" argument retrieval to the file. Because of this, you may specify a portion of the options on the command line and the remainder in a script file. Command line options that appear after the script file are not processed. You may "chain" from one script to another by placing a script file option at the end of the script. Note that this is truly a "chain" operation and that control is NOT returned to any prior script once a chain occurs.

## CrossePAC READER

The CrossePAC Reader is a specialized version of CrossePAC that is limited to extracting files from archives, viewing files in archives, and listing the directories of archives. The CrossePAC reader cannot create or modify an archive file.

## ARCHIVE FORMAT CONSIDERATIONS

CrossePAC's three archive formats are Text, ASCII and Binary. Text format archives consist solely of characters from the printable (non-control) ASCII character set. ASCII format archives are built using the entire 7-bit ASCII character set including all control characters. Binary format archives employ the full range of each 8-bit byte. In addition to the three character set options, archives can be created in a blocked or unblocked structure. Blocked archives are written as line oriented files and are the most portable between dissimilar systems. Blocking can range from 40 bytes to the maximum line length for the systems being used. Unblocked archives are written as a "stream" of bytes.

The choice of an archive format depends, to the greatest extent, on whether the archive will be transmitted to another system, the characteristics of the target system and the file transfer mechanism to be used. Each format provides a compromise between portability and compression efficiency. Unblocked Binary format creates the smallest possible archive files. However, this is also generally the least portable since non-ASCII characters are used and "stream" mode files are not supported on all systems. Blocked Text format archives are the most portable but are approximately 24% larger than their unblocked Binary equivalents.

### File Transfer Considerations

In a mainframe environment, communications are usually based on 7-bit synchronous or asynchronous lines with varying degrees of line protocol imposed on the data flow. Using the appropriate software, it is possible to transfer even binary data across communications lines of this type. In this case, file transfer software must "quote" any characters greater than 127 decimal and any control characters that conflict with the line protocol. Even the best file transfer software, however, imposes a 33% overhead during binary file transmission and some older packages suffer overhead of 100%. In terms of file transfer, conventional text files are the least costly to move over this type of line.

Note that even when communicating with the host via a LAN, terminal emulation and file transfer software may assume that only a 7-bit protocol-restricted data path is available and thus reformat the data in-flight to meet those restrictions. Where communications lines are full 8-bit and protocol free, AND the file transfer software is aware of the line capabilities, binary file transfer incurs no penalty.

Protocol collisions can be avoided through the use of the "eXCludechar=" option. This option allows specific characters to be eliminated from the character set used to build the archive. It has absolutely no effect on the content of source files within the archive. While this capability can be very valuable in working around a particularly restrictive communications environment, it should be used sparingly. If you find that it is necessary to exclude large numbers of characters, then you will probably be better off to "step down" to the next format (i.e. from Binary to ASCII, or from ASCII to Text).

As a general rule, use Binary format archives ONLY with 8-bit, protocol-free communications. For normal 7-bit lines with protocol, use Text format. ASCII format can be used for 7-bit protocol free transmission.

Obviously, the goal here is to achieve the fastest effective transfer time. In the final analysis, a little testing with your particular communications setup is the best way to determine the optimum format. When testing, use archives that take at least a few minutes to transfer so that your timing comparisons will be meaningful.

## DATA "PACKAGING"

CrossePAC can also be used as a data "packaging" tool. In heterogenous computing environments CrossePAC can be the key to making data portable between environments. This can involve moving data directly from one platform to another such as transferring a data file from DOS to Unix but can also include "warehousing" data on a foreign system for distribution purposes. A typical warehousing task would be to compress all of the executable files required for a DOS application and store the archive on a mainframe where it can then be downloaded to other PC's.

One other unique aspect of CrossePAC's archive format is the fact that a PAC file can be "embedded" in another file. Each CrossePAC archive begins with a unique header signature. When reading an archive CrossePAC bypasses any data that precedes the signature and ignores any data that follows the logical end of the archive. This feature allows a CrossePAC archive file to be embedded in another file.

## INTERNET MAIL

The popularity of the Internet has exploded in recent years with thousands of new users signing on each month. In particular, the Internet has become heavily used to exchange electronic mail between corporate networks, interactive services such as CompuServe, commercial email services, and Internet users. For long time users of corporate electronic mail systems this new world-wide connectivity has multiplied the value of their systems many times over. Unfortunately, the historical origins of the Internet prevent it from providing some of the higher-level services to which present day email users have become accustomed.

While the homogenous environment provided by LAN-based email commonly permits the "attachment" of arbitrary files to email messages, the Internet does not. Because Internet mail must be portable to all of the computing platforms that occur on the Net, messages are limited to text only. This means that binary files such as spreadsheets, graphic images, and executables cannot be mailed without first "encoding" them as text. In addition, some Internet gateways (such as CompuServe) impose a limit on the size of messages that can pass through.

CrossePAC's ability to both compress data and produce an archive in an encoded text format makes it ideal for attaching files to Internet mail. In our tests we have found that using a text format and a block size of 70 works well. Since Internet mail does not support attachments as a separate part of the message you can simply include the archive as a part of the message text. Many email programs support file insertion as an editor function. Windows based applications can make use of Windows' Clipboard cut and paste facilities to copy the archive into the message text.

When extracting the files in an embedded archive of this type it is not necessary separate the archive from the message text. Simply save the entire message to a file and run CrossePAC using the saved file name as the archive name. CrossePAC will automatically bypass the message text and begin processing when it locates the archive header signature. NOTE: Since CrossePAC assumes a file extension of .PAC it will be necessary to supply the entire file name on the command line if the .PAC extension was not used.

## EXAMPLE COMMANDS

### Create

In the following commands, if the archive already exists it will be updated by adding or replacing the listed files as necessary.

The following command creates an archive called DOCS.PAC and adds the files 1STQRTR.DOC, 2NDQRTR.DOC, and 3RDQRTR.DOC. The archive will be created with the default format and block size.

```
CPAC A DOCS 1STQRTR.DOC 2NDQRTR.DOC 3RDQRTR.DOC
```

The following command creates an archive called PROGS.PAC that contains all of the source files in the directory \PROJECT1.

```
CPAC ADD PROGS \PROJECT1\*.C
```

The following command creates an archive called PROGS.PAC that contains all of the source files in the directory \PROJECT1 and any subdirectories it contains.

```
CPAC A +S PROGS \PROJECT1\*.C
```

The following command produces the same archive as above but does not include any files that begin with DEMO.

```
CPAC ADD +SUB PROGS \PROJECT1\*.C -DEMO*.*
```

The following command creates an archive that contains all of the source files in the directory \PROJECT2 and any subdirectories it contains but for each file stores only the directory structure that is subordinate to \PROJECT2.

```
CPAC A +S PROGS \PROJECT2\*.C SP=RECURSED
```

### Replace

The following command updates the existing archive called DOCS.PAC by unconditionally replacing the file 2NDQRTR.DOC.

```
CPAC R DOCS 2NDQRTR.DOC
```

### Freshen

The following command updates the existing archive called SHEETS.PAC by replacing all files with a .XLS extension which are older than their counterparts on disk.

```
CPAC FRE SHEETS *.XLS
```

The following command updates the existing archive SHEETS.PAC by replacing all files in all directories which are older than their counterparts on disk.

CPAC FRE SHEETS +S \*.\*

### **Delete**

The following command deletes all files from the existing archive BACKUP.PAC that have an extension of .BAK.

CPAC DELETE BACKUP +S \*.BAK

NOTE: If the updated archive BACKUP.PAC that results from this delete operation does not contain any files, then the archive file itself will also be deleted.

### **List Archive Directory**

The following command will list the directory of the archive SAVESTUF.4ME.

CPAC L SAVESTUF.4ME

### **View a File in the Archive**

The following command will display the file READ.ME in the archive SWREL30.PAC.

CPAC V SWREL30.PAC READ.ME

## ERROR MESSAGES

CrossePAC error messages are divided into two categories. The first, argument errors, occur while CrossePAC is processing the command line and/or script files(s) and before any archive processing begins. These errors are generally self-explanatory. The second type, processing errors, occur while CrossePAC is processing an archive file.

Processing error messages are composed of a hexadecimal message number and an error text string. Common errors that require user intervention are listed below by message number. Any processing errors that occur which are not listed below represent an internal problem and should be reported to DSI.

0003 file not found

A request to process an existing archive failed because the named archive file could not be found.

0004 insufficient memory

There is not enough memory available in the system to create internal buffers and tables.

0005 file create error

The system returned an error while attempting to create a new archive file.

0006 bad header

While opening an existing archive file, CrossePAC detected invalid header information. This indicates that the archive file has been corrupted.

0007 can't create temp update file

While attempting to update an existing archive file, CrossePAC was unable to create a temporary work file.

0008 header write error

An I/O error occurred while writing the header of a new or updated archive file. This generally indicates a disk drive or media problem but can also be caused by a lack of disk space.

0009 too many exclude chars

The limit of excluded characters has been exceeded for the archive format in use. The 'eXcludechars=' option should be used very sparingly. If more than a few characters must be excluded it is more efficient to switch to another archive format. (i.e. from binary to ASCII, from ASCII to text).

000a incompatible PAC file version

An attempt to open an existing archive file failed because the archive version is incompatible with the copy of CrossePAC that is in use. This generally indicates an archive that was built with a newer version of CrossePAC.

000b PAC signature not found

An attempt to open an existing archive file failed because the header signature could not be found. This is generally caused by specifying a non-archive file on the command line or by a severely corrupted archive.

000c invalid PAC format

An attempt to open an existing archive file failed because the archive is in an unknown format. This can occur when processing an archive written by a newer version of CrossePAC or by a corrupted archive.

000d no subfile on input

An attempt to open an existing archive file failed because the archive contains no data. Normally, CrossePAC will not create an empty archive. This error is an indication of a damaged archive file or a processing interruption when the archive was created.

000e translation overlap

An attempt to open an existing archive file failed because of an overlap in the embedded translation table. CrossePAC includes a complete copy of its character translation table as a part of the archive header so that any character translation that occurs during file transfer will automatically be accounted for when reading the archive. However, any character translation that does not follow a one-to-one mapping (i.e. two characters translated to the same value) is not reversible. This results in translation table overlap.

0101 no subfiles present

When closing an new or updated archive, CrossePAC found no files in the archive. In this case the archive file is considered logically 'empty' and is deleted. This can result when creating a new archive in which none of the listed files or wildcards resulted in a match or when a DELETE operation removes all files from an archive.

0103 I/O error closing file

An I/O error occurred while closing a new or updated archive. This is generally caused by a lack of disk space but can also result from a disk drive or media failure.

0303 insufficient memory

There is not enough memory available in the system to create internal buffers and tables.

0304 external file non-existent

While preparing to add to or update an archive, CrossePAC could not open the external source file that was to be added. Since CrossePAC checks the existence of files before attempting to add them, this error is generally the result of a file access permission problem or a sharing conflict.

0305 error creating work file

While preparing to add to or update an archive, CrossePAC was unable to create a temporary work file. This generally indicates a lack of disk space but can also result from a disk drive or media problem.

0306 I/O error

An error occurred while positioning the archive file in preparation for adding or updating a file. This generally indicates a disk drive or media problem.

0403 insufficient memory

There is not enough memory available in the system to create internal buffers and tables.

0405 error creating work file

During an extract operation CrossePAC received an error while attempting to create the new file to receive the extract. This can occur due to insufficient disk space, insufficient file access permissions, file sharing conflicts, or a disk drive or media problem.

0406 invalid subfile header

When attempting to read the header of the next file in an archive, CrossePAC detected invalid information. This is generally caused by a damaged archive.

0407 I/O error

An I/O error occurred while creating or updating an archive. This is generally a result of insufficient disk space but can also be caused by a disk drive or media problem.

0409 error opening printer

When processing a 'Print' request CrossePAC was unable to open the standard print device.

040b error rebuilding file name

During an extract operation CrossePAC was unable to reconstruct the file name of a file in the archive. This is generally caused by insufficient memory.

040c user cancelled expansion

This message is displayed as a result of the user selecting the 'cancel' option in response to an overwrite warning.

040d file could not be accessed

During an extract operation CrossePAC could not open the external disk file into which a file was to be extracted. This can be due to insufficient disk space, a file access permissions conflict, or disk drive or media problems.

0501 I/O error

When creating or updating an archive an I/O error occurred while writing a file into the archive. This is generally caused by insufficient disk space but can also result from a disk drive or media problem.

0701 seek error

An I/O error has occurred while positioning the archive file. This can be caused by a damaged archive or a disk drive or media problem.

0702 I/O error

An I/O error has occurred while positioning the archive file. This can be caused by a damaged archive or a disk drive or media problem.

0801 command processing failed

A fatal error occurred while processing the command line. This is typically caused by insufficient memory.

0905 insufficient memory

While updating an archive file there was not enough memory available for internal buffers and tables.

0906 bad file name

While updating an archive file CrossePAC was unable to reconstruct a file name. This is typically caused by insufficient memory.

0907 I/O error

An I/O error occurred while updating an archive file. This is generally due to insufficient disk space but can also be caused by a disk drive or media problem.

## GCOS IMPLEMENTATION SPECIFICS

Version 1.34 of CrossePAC for GCOS has the following restrictions:

1) Random or non-linked files are not supported as PAC files. Default for PAC files created under GCOS is Format=Text and Blksize=140. For PAC files created elsewhere, only Format=Text or Ascii with Blksize=40 to 1272, are supported under GCOS.

2) Random or non-linked files are not supported as destination files.

Note: Random files (i.e. UFAS) may be compressed under GCOS and decompressed under DOS or OS/2 (i.e. for processing by GCOS-8 COBOL Workbench).

3) Wildcard file names are not supported.

4) The +Subdirectories option applies ONLY to eXtraction commands. Walking of FMS catalog structure is not supported.

5) The +UsePath option which implies directory (catalog) creation is not supported.

6) CrossePAC will create a new PAC file during each Append/Replace/Delete/Reformat operation. Sufficient FMS space, for both old and new PAC files, must be available during such operation. The old PAC file is released and the new PAC file renamed, upon completion of the operation.

## GCOS INSTALLATION

CrossePAC for GCOS is shipped either on diskette or tape (FMS save format).

### Installation from Diskette

Installation from software diskette require a Binary-File-Transfer facility (i.e. Kermit, Microfit) for the H\* executable files. Transfer files from diskette to GCOS as follows:

```
a:\CPAC          ->  CMDLIB/CPAC (binary)
a:\CPACB         ->  CMDLIB/CPACB (binary)
a:\CPACJCL       ->  CMDLIB/CPACJCL (text)
```

### Installation from Tape

The software release tape is in FMS save format and need to be restored as follows:

```
$  IDENT      ...
$  FILSYS
$  PRIVITY
RESTORE BCSTB8/CPAC,NEWNAM/umc/
$  TAPE9      PR,X1DD,,99999,,CPAC-GCOS,,DEN16
```

where umc = your choice of UMC for installation.

Once installed it is recommended that you make CPAC available through CMDLIB as follows:

```
COPY umc/CPAC/CPAC          ->  CMDLIB/CPAC
COPY umc/CPAC/CPACB        ->  CMDLIB/CPACB
COPY umc/CPAC/CPACJCL      ->  CMDLIB/CPACJCL
```

Note: If installed directly under CMDLIB (i.e. umc = CMDLIB), you need to rename the CPAC catalog (i.e. CPAC134).

### TSS

When installed under CMDLIB, CrossePAC is available through the TSS command CPAC.

```
CPAC A DOCS 1STQRTR.DOC 2NDQRTR.DOC 3RDQRTR.DOC
```

CrossePAC includes on-line documentation in the standard EXPL format. You may either create a specific EXPL catalog (i.e. CMDLIB/CPACEXPL) or integrate the documentation into the standard EXPL structure (i.e. CMDLIB/EXPL). Following documentation files are provided:

```
INDEX
EXPL
TSS
BATCH
```

either from the diskette (i.e. a:\EXPL\file) or the FMS umc (i.e. umc/CPAC/EXPL/file).

## Batch

Execution from batch is very similar to TSS, except that CrossePAC is launched via JCL instead of a command.

```
$ IDENT      ...  
$ SELECT    CMDLIB/CPACJCL  
CPAC A umc/DOCS umc/1STQRTR.DOC umc/2NDQRTR.DOC umc/3RDQRTR.DOC
```

Since there is no implicit UMC under batch execution, the files specifications of command line must include the full catalog-file string.

## DOS & OS/2 INSTALLATION

CrossePAC for DOS and/or OS/2 is shipped on diskette or bundled with GCOS-8 COBOL Workbench.

### Installation from G8WB

The bundled CrossePAC license(s) - both DOS and OS/2 - are installed under \$G8WB\DIR\EXEDLL, where either version is retrieved as required during embedded use.

If you want to use CrossePAC as a general command, you need to include \$G8WB\DIR\EXEDLL (i.e. G8WB\EXEDLL) in your path setting, or copy the command files to an existing directory.

```
CPAC.EXE   - for use under DOS or DOS-Box under OS/2  
CPAC2.EXE  - for use under OS/2
```

### Installation from Diskette

Create a new directory and copy the CPAC.EXE file (DOS or OS/2) into the directory, and include the directory in your path setting.

```
C:\MD CPAC  
COPY A:\CPAC.EXE
```

or copy CPAC.EXE into an existing directory, which is already included in your path setting.