Pascal Tools II
Apple II*
Installation & Operating Instructions

Pascal Tools II
Apple II*
Installation & Operating Instructions

ADVANCED BUSINESS TECHNOLOGY, INC.

12333 Saratoga Sunnyvale Road, Saratoga, California 95070   408/446-2013

TABLE OF CONTENTS

PASCAL TOOLS II
A Tool Kit for the Serious Pascal Programmer

from
Advanced Business Technology, Inc.
and
Silicon Valley Software, Inc.

General Information

Pascal Tools II is a package of various programming aids (tools) for professional programmers using Pascal on the Apple II* which has been made available by Advanced Business Technology, Inc. Any serious Apple Pascal user will find this tool kit indispensible for the handling of large programming and other projects. These tools have been developed over time by Silicon Valley Software, Inc., as an aid to their software development of production systems on the Apple II. The programs included in Pascal Tools II are primarily directed at the handling of text files (and, for some of the utilities, primarily Pascal program source files). A companion package is available, Pascal Tools I, which concentrates primarily on the handling of binary and code files. All of the utilities in Pascal Tools I and Pascal Tools II have been proven useful and rugged in a production environment.

*Apple II is a trademark of Apple Computer, Inc.

CONTENTS OF THE PASCAL TOOLS II DISKETTE

Your Pascal Tools II diskette contains the following executable files:

PATCH.CODE—Disk File Hexadecimal Editing Utility,
DUMPPCODE—P-Code Disassembler,
GREP.CODE—General Purpose Pattern, Grabber and Replacement Utility,
CMP.CODE—Binary File Compare, Disk Verify,
FILEGEN.CODE—Arbitrary File Generator, and
ECHO.CODE—Keyboard Character Code Indicator.

GENERAL OPERATING INSTRUCTIONS

Each tool is e(Xecuted under the Apple Pascal system. The Pascal Tools II diskette can be removed during the execution of each utility. The operation of each tool is prompt driven, with "help" available on every user input by typing ? followed by return (just ? for PATCH). It should be possible to use these tools immediately without further reading in the manual, but in order to make full use of all of their capabilities, the user is referred to the complete description of each system in the remainder of this manual.

PATCH
Disk File Hexadecimal Editing Utility

CAPABILITIES

PATCH allows the interactive inspection, editing, and printing of arbitrary binary files (.CODE, .TEXT, or data files) in hexadecimal. PATCH is a screen oriented editor in which the user positions the cursor on the PATCH display using the same cursor control keys used in the Apple Pascal text file Editor. Once the cursor is positioned, changes to the displayed disk block can be entered in hexadecimal. Under user command, the modified block of data can either be placed back on the diskette or treated as temporary and discarded. The PATCH display shows the data both as hexadecimal and in its ASCII representation in parallel on the screen. PATCH provides a hexadecimal dump facility through a command which causes the PATCH display to be directed to a device (printed) or placed in a file.

Normally disk blocks associated with a file are edited using PATCH, but it is also possible to edit directly on the disk device, including directory areas and ignoring file boundaries. In the event that a directory has been lost on a diskette, this capability can be very useful in locating files which might otherwise be lost.

Help is available in PATCH by typing ? or H when PATCH is prompting. PATCH is terminated by typing Q when PATCH is prompting. Since PATCH is very interactive, the best way to learn to use it effectively is through experimentation.

OPERATION

PATCH is operated by e(Xecuting PATCH. The command menu will be displayed at the top of the screen. At this time the Pascal Tools II diskette may be removed from the system without affecting the operation of PATCH.

PATCH is designed to configure itself to the user's screen characteristics and cursor control characters. It should automatically determine whether the system has a 40 column screen or an 80 column screen. For purposes of the manual, the 80 character display will be used in examples so that if you have a narrow screen, you may see slightly different prompts.

The command menu contains the following prompts:

```
Quit   New.file   Read + - Update
Print  List.file  Help ? = Screen
```

Each action is initiated by typing the single keystroke which is the initial character of the command.

Typing Q terminates PATCH. Typing ? or H causes the help file to be displayed.

Typing S allows the interactive changing of the screen parameters from wide to narrow or from narrow to wide (in the unlikely event that PATCH makes the wrong determination of the user's screen width or if the user prefers to have the wide display created even though the system's screen is narrow).

Normally, a file name and block offset within the file will be indicated to PATCH for editing purposes. Typing N causes PATCH to prompt for the name of the file to be operated on. The entire file name must be typed, including the .TEXT or .CODE suffix. Typing R causes PATCH to prompt for the block number within that file to be operated on. Once the file name is set, repeated R commands may be given to PATCH. If no file name is set, issuing the R command causes PATCH to first prompt for the file name and then the disk block within that file. (Note: the first block in the file is block 0. PATCH indicates the number of blocks in the file, but the highest block number is one less than this number.)

The + and - commands are shorthand for R the next block and R the previous block.

Once a disk block has been read into PATCH, it is automatically displayed on the screen. On a wide screen, only one half of the block fits at a time. On a narrow screen, only one quarter of the block fits at a time. The entire block is available for inspection and modification, even though only a portion of the block is displayed. In order to see portions of the block not displayed, move the cursor to a byte not currently on the screen (see below).

In order to move the cursor around the displayed block, use the same control keys used in the Apple Pascal text file Editor to move right, left, up, and down. Also, the return key moves the cursor to the start of the next display line, the space key moves the cursor to the right, and the tab key (ctrl-I) moves the cursor one byte forward. In addition to these cursor moving commands, the = command prompts for a byte number (in the range 0 to 511) and positions the cursor on the byte number specified.

Once the cursor is positioned, the indicated byte will be altered by typing the characters 0-9 or A-F. As bytes are changed, the cursor advances to succeeding bytes. These changes are immediately displayed both in the hexadecimal portion of the display and in the ASCII portion of the display; however, there is no change made to the copy of the block on the diskette until PATCH is given the U command to update the original block. Once any change is made in a block, the "Modified" indicator will be displayed.

In order to obtain printed output of the PATCH display, or to place the PATCH display into a file, use the L command to set a listing file and the P command to print the current block to that file or device. Issuing just the P command causes the target file or device to be prompted for, with default file name PRINTER: used by PATCH if just return is entered. The entire disk block is printed, not just the portion which happens to be on the screen.

NOTES

In order to edit the diskette as a device, ignoring file boundaries and getting access to the directory portion of the device, specify just the unit name as the file name to be edited (i.e., specify #4: as the file name).

Once the file name is set, it can be altered using the N command. Similarly, once the listing file is set, it can be reset using the L command.

It is not impossible to mess up the PATCH display by specifying CONSOLE: as the listing file. This is not recommended. Similarly, the display can be affected by typing successive return keys when PATCH is prompting for integers. If this should happen, type ? to get help and return to PATCH. This will fully reinitialize the screen image.

DUMPPCODE
P-Code Disassembler

CAPABILITIES

DUMPPCODE displays the information in Apple Pascal .CODE files in a symbolic form, including the P-Code instructions generated by the Pascal or FORTRAN compilers. In addition to the P-Code instructions, DUMPPCODE displays various information about the segment and procedure structure of the .CODE file and information used by the linker in processing compilation units.

DUMPPCODE should be used in conjunction with the Apple Pascal Operating System Reference Manual, particularly the chapter on architecture of the P-Machine which explains the various P-Machine op codes and the chapter on file formats which describes the other kinds of information found in .CODE files.

Help is available to all of DUMPPCODE's prompts by typing ? followed by the return key. DUMPPCODE may be aborted by typing <esc> followed by the return key to any prompt.

OPERATION

DUMPPCODE is operated by e(Xecuting DUMPPCODE. DUMP-PCODE prompts

```
Listing file [CONSOLE:] -
```

for the name of the file on which to place the output. At this time, the Pascal Tools II diskette can be removed from the system without affecting the operation of DUMPPCODE. If the user types just return, the output is placed on the device CONSOLE:

DUMPPCODE then repeatedly prompts for input files to disassemble:

```
Input file [.CODE] -
```

DUMPCODE will automatically add the suffix .CODE to the file name provided if it is not supplied by the user (only .CODE files are valid input files to DUMPPCODE). Typing just return to the prompt terminates the execution of DUMPPCODE normally.

OPTIONS

Options may be specified to DUMPPCODE at any time that user input is requested. Options are set or reset with + or - followed by the appropriate option letter. At any time, the current value of the options is displayed when help (?) is requested. The following options are recognized by DUMPPCODE:

C	controls whether code is disassembled (default +C).
H	controls whether code is dumped in hex (default -H).
M	controls listing of miscellaneous information (default +M).

The C option controls whether the disassembled object code will be listed, or just the other information in the code file. Setting the H option causes the code file to be listed in hexadecimal (with parallel ASCII interpretation) in addition to the other information. The M flag controls whether the linker interface information is listed.

NOTES

Typing <esc-ret> to any prompt aborts DUMPPCODE. If DUMPPCODE is terminated in this manner, rather than by just return to the input file prompt, it is considered an abnormal termination and the listing file is not made permanent.

GREP
General Purpose Pattern Grabber and Replacement Utility

CAPABILITIES

GREP searches a text file for lines which contain matches to user specified patterns. These lines are displayed. Optionally, GREP will create a modified version of the file by replacing each occurrence of a matched pattern with another pattern. The search may optionally be restricted to Pascal identifiers and reserved words. GREP can be set to process an entire Pascal program, including files contained in the program through the $I Pascal "include" directive. Optionally, GREP will treat matching upper and lower case letters as identical for purposes of finding search patterns.

GREP accepts a set of search patterns (and replacement patterns if appropriate). It is then possible to apply these patterns to a series of files, automating the task of making multiple editing changes to multiple files.

Among the other uses for GREP are as an aid to matching 'BEGIN's, 'RECORD's, and 'CASE's with 'END's (displaying lines which match these key words immediately indicates the program's nesting structure and will reveal errors) and to easily determine where particular identifiers are accessed in a program.

Help is available to all of GREP's prompts by typing ? followed by the return key. GREP may be aborted by typing <esc> followed by the return key to any prompt.

OPERATION

GREP is operated by e(Xecuting GREP. GREP prompts:

```
Input file [.TEXT] -
```

for the name of the first file to search. At this time, the Pascal Tools II diskette can be removed from the system without affecting the operation of GREP. GREP will automatically add the .TEXT suffix to the file name provided if not supplied by the user (only .TEXT files are valid input files to GREP). GREP may be terminated by entering return to the input file prompt.

GREP then repeatedly prompts for patterns for which to search:

```
   Search for pattern -
```

Enter any series of characters terminated by return. Entering just return terminates the prompting for search patterns. (Note: Patterns which begin with the question mark character are interpreted as help requests. Patterns which begin with <esc> are interpreted as abnormal termination requests. Patterns which begin with the plus or minus sign are interpreted as option requests. It is not possible to search for any of these patterns.) If the replace option has been set (see OPTIONS below) GREP will prompt for the string to replace the search pattern (assume that the user entered abc as the search pattern):

```
   Replace 'abc' with pattern -
```

Once again, enter any series of characters terminated by return. Entering just return will cause the search pattern to be replaced by the empty string (deleted) in the file which is created with the replacement patterns. (Note: none of '?', '+', '-', nor <esc> may be initial characters in replacement patterns.)

The current search patterns and replacement patterns are displayed when help is requested by entering ? to any of GREP's prompts.

Having terminated inputing the series of search patterns by entering return to the search pattern prompt, GREP will do one of several things. If the replacement option is set (see below) and search patterns have been entered without replacement strings, GREP will go back and get replacement strings for these patterns. If the replacement option is set, GREP will require the name of a file (or device) on which to place the file with the designated replacements. The prompt for this is:

```
   Output file to receive file with replacements -
```

The file name is taken exactly as typed, with no automatic ending supplied by GREP. In the event that the replacement option is not set, GREP will not prompt for replacement patterns or an output file.

The input file is then searched for lines with matching patterns which are displayed on the screen. Having completed the search, GREP prompts for the next input file. All of the options, patterns, and replacement strings are retained for the next input file.

OPTIONS

Options may be specified to GREP at any time that user input is requested. Options are set or reset with + or - followed by the appropriate option letter. At any time, the current value of the options is displayed when help (?) is requested. The following options are recognized by GREP:

R	controls whether patterns are to be replaced (default -R).
T	controls whether pattern is limited to Pascal tokens (default -T).
C	controls whether upper and lower case letters match (default -C).
I	controls whether Pascal $I includes are processed (default -I).
Q	suppresses display of matching lines (default -Q

The R option determines whether a replacement file is to be created or whether GREP is just being used to display matched lines. If the R option is set, GREP will require an output file for each input file and a replacement string for each search pattern.

The T option can be used to restrict the searches to Pascal identifiers and reserved words. If +T is set, Pascal comments and string constants are skipped in the search. Also, for example, a real number of the form '10.3E5' will not be matched by the pattern 'E5'.

Setting the +T option also insures that only a single replacement will be attempted for a given occurrence of an identifier or reserved word in the source. For example, suppose the search pattern 'abc' is to be replaced by the string 'was once a followed by b followed by c' and the search pattern 'once' is to be replaced by 'xyz'. If +T is set, an occurrence of 'abc' in the input file will be replaced with 'was once a followed by b followed by c' and the 'once' placed into the replacement file will not be replaced with 'xyz'. If -T is set, the replacement of 'once' by 'xyz' might or might not occur (result is undefined).

Similarly, if the replacement string contains the search string, as in the case of 'abc' being replaced by 'was once abc', GREP will make the replacement once under the +T option, but would attempt to repeatedly (forever) under the -T option. (The user is given a warning by GREP not to issue nonterminating replacements, if it is attempted anyway, GREP will terminate abnormally.)

Setting the C option to +C causes GREP to treat upper and lower case letters as identical for purposes of searching for pattern matches. The replacement pattern is used exactly as entered regardless of whether the C option is set.

Setting the I option to +I causes GREP to include files indicated Pascal $I directives. The +I option only works if Pascal tokens are being processed (+T option set). Setting +I automatically sets +T, although if -T is subsequently set, the +I option will not operate. GREP only recognizes $I directives which are the first and only option in a Pascal comment. Moreover, the comment should terminate after the file name and the line which contains the comment should terminate directly after the end of the comment (no intervening spaces). If the +I option is set, the file name is indicated with each line which is displayed.

Setting the quiet option, +Q, suppresses the listing of the matching lines on the screen. This is desirable if the screen is designated as the output file for the replacement strings or if the primary objective is to create a replacement file and the output is considered unnecessary.

NOTES

Search patterns are scanned in the reverse order from the order in which they are entered. Therefore, if replacements are to be made, and an erroneous input was entered, it can be overridden by a subsequent search pattern and replacement string.

If the -T option is set, each pattern is repeatedly replaced until it no longer matches. (For example replacing ' ' with ' ' will cause all multiple blanks in a row to be replaced with a single blank. Once a pattern fails to match, the next pattern (in reverse order) is tried until it fails to match, etc. Under the +T option, exactly one replacement takes place.

Patterns and replacements strings may be up to 80 characters in length. Input lines and output lines in the replacement file may be up to 255 characters in length. About 90 patterns may be entered to GREP before the system runs out of space.

CMP
Binary File Compare, Disk Verify

CAPABILITIES

CMP reports the bytes which differ between pairs of arbitrary (.CODE, .TEXT, or data) files or that they are equal. CMP can be used to compare two volumes (diskettes) to verify that they contain exactly the same data. This is preferable to bad block scanning after transferring one diskette to another diskette since it not only checks that the blocks are correctly formatted, but also that the data was written correctly.

CMP also accepts "wild card" file designations. This allows file by file comparisons of groups of files with differing names or groups of files on different devices.

Help is available to all of CMP's prompts (except where noted below) by typing ? followed by the return key. CMP may be aborted by typing <esc> followed by the return key to any prompt (except as noted below).

OPERATION

CMP is operated by e(Xecuting CMP. CMP repeatedly prompts for pairs of file names to be compared. The prompt for the first file is:

```
First input file -
```

and the prompt for the file to be compared to this file is:

```
Second input file -
```

After the first input file prompt, the Pascal Tools II diskette (and if desired the Pascal system diskette) can be removed from the system without affecting the operation of CMP. Typing just return to either prompt terminates the execution of CMP normally. The entire file name must be provided by the user, including .TEXT or .CODE file name extensions, but "wild card" file names are accepted (see below). The pairs of files to be compared must both be available on line to CMP simultaneously; however, neither the Pascal Tools II diskette nor the system diskette need to be available. Remember to replace the Pascal system diskette into its drive before terminating CMP if you have removed it.

In order to compare two diskettes, enter just the volume names as input file names (such as #5: or TOOLS2:).

CMP will accept both the first and second file names in response to the first input file prompt if the names are separated by a comma.

If the pair of files being compared are identical binary files, CMP goes on silently to the next file compare. If differences are detected, a message is printed on the CONSOLE: and, unless the differences relate to file length, the user is queried as to whether CMP should continue, abort, or go on to the next file compare. After reporting the byte position of the detected difference and the byte value from each file, the user may enter <esc> followed by return to go on to the next file comparison, or just return to continue the current comparison.

WILD CARD FILE DESIGNATIONS

File names provided to CMP may contain the "wild card" designators =, ?, and $. These characters are interpreted in the same manner that most versions of the filer interpret them.

In the event that the ? designator is used, CMP will prompt before each file compare. Responding Y causes the files to be compared (no return is necessary). Any other response causes CMP to go on to the next file comparison. It is not possible to abort CMP, obtain help, or set options from the prompt which results from using the ? file designator.

OPTIONS

Options may be specified to CMP at any time that user input is requested except when CMP is prompting as a result of the ? wild card. Options are set or reset with + or - followed by the appropriate option letter. At any time, the current value of the options is displayed when help (?) is requested. The following options are recognized by CMP:

Q makes CMP run quietly (default -Q).

NOTES

CMP makes no attempt to resynchronize matching bytes after a difference is detected, even if the difference is as simple as one extra byte in one of the files.

When comparing .TEXT files using CMP, files which are "identical" in a text file sense may not be identical in a binary sense. Specifically, .TEXT files contain several possible representations for initial blanks in lines and may have differing amounts of "zero filler" at the end of block pairs and still have identical text information. Of course, CMP detects these differences. Moreover, CMP examines the first block pair of text files and detects differences which are due to differing "edit environments." DIFF, available in Pascal Tools I, is recommended for comparing .TEXT files for identical text information.

When comparing two diskettes using their volume name, CMP may report that they are different even if each file is identical and the directory on each prints identically. This is due to the diskette bytes which are in the unused portion of the directory and diskette bytes which are not in files. CMP should report diskette volumes to be identical directly after a complete diskette to diskette transfer since all of these areas are copied by the filer from one diskette to the other.

A common use for the "wild card" file designation is to compare all files on two diskettes in pairs by name (#4: = to #5:$ or #4: = to #5: = or selectively #4:$ to #5:$. In this way, the differences in the directory structure, placement of the files on the diskette, and empty areas on the diskette do not interfere with full diskette comparisons.

FILEGEN
Arbitrary File Generator

CAPABILITIES

The FILEGEN utility allows for the straightforward production of files with arbitrary, possibly nonprintable, characters in them. FILEGEN can produce a file format which is exactly like .TEXT file format except for the embedded nonprintable (possibly zero) characters or it can produce a standard date file format (see notes). FILEGEN processes an input program in .TEXT file format, presumably produced by the Apple Pascal Editor and interprets certain character sequences as arbitrary character codes.

One of the uses of FILEGEN is to produce exec files for Version 1.1 of Apple Pascal directly, rather than by "example." In order to do this, the user must be familiar with the character codes which are issued by the CONSOLE: keyboard. The ECHO tool (also contained in Pascal Tools II) is very useful for determining these codes.

Another use for FILEGEN is to get backspace characters into a file which is to be printed so that underlining or other overstriking can be accomplished. Similarly, it is sometimes desirable to put cursor control characters into files which are to be directed to a screen device.

Help is available to all of FILEGEN's prompts by typing ? followed by the return key. Entering < esc > followed by return to any prompt aborts FILEGEN.

OPERATION

FILEGEN is operated by e(Xecuting FILEGEN. FILEGEN prompts:

```
Input file [.TEXT] -
```

for its input file, see input file format below. The Pascal Tools II diskette may be removed from the system at this time without affecting the operation of FILEGEN. The input file must be a .TEXT file, and FILEGEN will provide the .TEXT suffix automatically if necessary for the user. Next FILEGEN prompts.:

```
Output file -
```

for the name of the file on which to place the output. If the output file name supplied ends in .TEXT, FILEGEN will produce a file format as close to .TEXT file format as possible, considering that nonprintable characters are normally not permitted in .TEXT files (see notes). If the output file name does not end in .TEXT, FILEGEN produces a data file format described below (see notes).

INPUT FILE FORMAT

Characters in the input file are passed directly to the output file with the following exceptions:

1. !! is passed to the output as !
2. !integervalue! is passed to the output as chr (integervalue). For example, !66! is passed to the output as B.
3. The DLE character at the beginning of .TEXT file lines and the blankcount which follows it are converted to an appropriate number of blanks.

If a ! is detected in the input file, it must be followed by either another ! of by an integer value and a terminating !. No blanks, pluses, or minuses, are permitted between the exclamation points. FILEGEN issues error messages if the input format is incorrect.

NOTES

Remember the following points when producing exec files for Version 1.1 of Apple Pascal: The file produced should have a name which ends in .TEXT; don't forget to put the initial and terminating characters into the file; and the file must be an exact script for what is to happen from the terminal. It is very useful to use PATCH to look at a few exec files before attempting to produce your own using FILEGEN.

Files processed by FILEGEN are in .TEXT format which has the following characteristics:

—The file name ends in .TEXT.
—The first two blocks of the file do not contain data and are ignored.
—The file is an even number of blocks long.
—Each block pair, starting with blocks 2/3 consists of full lines followed by some number of trailing zeros which pad out the end of the block pair.
—A line consists of the DLE character (character code 16), followed by a byte indicating 32 + the number of initial blanks, followed by a string of (nonzero, non chr (13) characters, followed by the return character (character code 13). The DLE and blank count are optional.
—If the file has been produced by the text editor, or a system compatible with it, the final block pair will have an "extra" blank line at its end. FILEGEN passes this blank line on to the output. It can be avoided by placing the editor cursor at the end of the last meaningful line in the file.

In general, when writing a file from a Pascal program, it will be in .TEXT format providing its name ends in .TEXT. Pascal I/O does not, however, prohibit the placement of zero characters in the middle of a file which will disrupt the operation of FILEGEN (as well as the system text editor).

If the output file name ends in .TEXT, FILEGEN produces a file with the following format:

—The file name ends in .TEXT.
—The first two blocks of the file are zeros. This is interpreted by the editor as a reasonable edit context if the output of FILEGEN is edited.
—The file is an even number of blocks long.
—Each block pair, starting with blocks 2/3 consists of full lines followed by some number of trailing zeros which pad out the end of the block pair.
—A line consists of a string of (non chr (13)) characters, followed by the return character (character code 13).

If the output file name does not end in .TEXT, FILEGEN produces a datafile with the following characteristics:

—Data begins at the start of the file.
—Data is a stream of characters. The end of file is indicated by the character count of the last block of the file found in the directory entry of the file (which is available to Pascal programs through the "eof" function).

ECHO
Keyboard Character Code Indicator

CAPABILITIES

ECHO provides an easy means of determining which internal character codes correspond to the various CONSOLE: keystrokes on a particular system. This information is useful when preparing input files for FILEGEN.

OPERATION

ECHO is operated by e(Xecuting ECHO. Operating instructions are automatically displayed.

DISCLAIMER
OF ALL WARRANTIES AND LIABILITY

Advanced Business Technology, Inc.
12333 Saratoga-Sunnyvale Road
Saratoga, California 95070

ADVANCED BUSINESS
TECHNOLOGY, INC.