Written by Ben Castricum

May 30, 1995

This is the documentation belonging to and explaining the use of:

UNP V4.11

Executable file restore utility

---> PDF-Conversion by Thomas Antoni - thomas@antonis.de - www.qbasic.de <---

TABLE OF CONTENTS:

Disclaimer

Under NO circumstances I can be held responsible for any damage caused by files in this or any other package containing programs written by me. (That should do it :-)

What is UNP ?

UNP's main purpose is to restore executable files to their original state. However it can do more than that. UNP can optimise EXE-headers, remove debug information, convert files from one structure to the other, scan directories for compressed files, reveal hidden viruses and even make files that didn't run anymore run again.

General info

Before you start using UNP, I would like to point out a few things which you might take into consideration.

Compressed EXE files containing an overlay may not work correctly after they have been decompressed. Decompression expands the code size of the EXE file which also means that the overlay moves up. Some programs do not check where the overlay currently is but just use a constant to get the overlay. If this is the case, most anything can happen.

When you use UNP to convert a file to another structure, please take into consideration that the converted program never runs under the exact same conditions as it did before. Though these differences are likely not to

cause any problems with most programs, there are always programs which expect just that what is changed by conversion.

UNP can do just about anything with files. This definately includes messing up your files. For that reason it is always a good idea to have a backup of the files your are going to process. Someone suggested to let the -b (create backup) option turned on by default. Although this is a good idea, it's still not 100% reliable.

UNP is not case sensitive in anyway, nor does it care about extensions. This however does not mean that it is possible to convert files which are reported by UNP to be "binary (.COM)" can all be converted to .EXE files. Files which are not really .COM files (e.g. .BAT or .GIF) will not run or view the picture when converted and executed.

How to use UNP

If you type UNP without any parameters then you will get the built-in help screen of UNP which is explained below.

- Commands - These are 1 character long and only one can be specified on the command line. It does not really matter where you put it. If no command is specified, the E command is used.

c = convert to COM file
Some .EXE files can be converted to .COM files. You can do this by using this command. Please note that the resulting file will not automaticly have a .COM extention. You should only convert a file when you know exactly what you are doing (see general info section).

d = make current options default
Using this command enables you to specify the default options yourself. Simply type the options on the commandline you would like to have as default and use this command. UNP will modify itself to the settings as default. For example to let UNP always create a backup use

```
UNP d -b+
```

UNP stores the new settings in itself, which means that UNP is self-modifying. With most anti-virus programs, this causes some alarm to go off. Check your anti-virus program documentation on how to solve this problem (see also: Hey! UNP is compressed!)

e = expand compressed file (default)
This command expands the compressed file. If you do not specify a command, UNP will use this by default. Using this command without a filename will result in unpacking all files in the current directory

i = show info only
If you just want some information about the file, this is the command to use. UNP will show all information like the E command but will not decompress or write the file back.

l = load and save
This command loads a .COM or .EXE file but does not expand it. It will be written back just like a decompressed file would be written back. This is useful in case you want to remove an overlay, irrelevant header data or optimize the relocation items.

m = MarkEXE, insert a file in header
MarkEXE is a small utility supplied with PROTECT! EXE/COM V5.0. This program can add a piece of text to an EXE file in such a way that when the file is shown on screen the user can see that piece of text. The 'M' command does not exactly do the same as MarkEXE. First it inserts the file before the relocation items, this way any EOF markers in the relocation items won't screw it up. Second, UNP does not place the same piece of text at the end of the code, since I see this as more or less screwing up the file.

o = copy overlay
A new (and probably rarely used) command is the overlay copy command. With this you can get the overlay from some .EXE file and append to some other .EXE file. The idea behind this is that when you use LZEXE as

compressor, the overlay is removed from the file. With this command you can place the overlay back.

s = search for compressed files
When you use this command, only a small list of compressed files matching the Infile wildcard will be generated. To save some space on the screen, the pathname of the file will not be show. But since UNP does not work recursive, it should not be a problem.

t = trace executable
My first attempt to a general unpacker can be found in this command. Actually there are 2 different implemtations. The implementation used for .COM files will single-step through a program and checking every instruction if the original program has been restored. If UNP thinks it has, it will stop and write the file back. Unfortunately this is a very slow process. The .EXE implementation also single-steps through the file but it checks every step to see if a known packer has been revealed. If it has found one, it will remove it and write it the resulting file back. If the program has not been compressed with a known packer, sooner or later some interrupt will be used which UNP will detect and abort the tracing.

x = convert to EXE file
Some compressors can only compress .EXE files (like LZEXE). With this command you can convert a .COM file to an .EXE file. The resulting file will not be written back with an .EXE extension by default. As with the .EXE to .COM conversion, be sure you know what you're doing. Not all programs can be converted.

- Options - Even more fun can be achieved with specifying options on the command line. Options can be passed sepparated (like -a -b -c) but can also be combined (like -abc). After each option there can be one of the characters "-", "+" or "?". The first turns switches off, the second turns them on and the third.. well it turns them on as well. But the real purpose of the question mark is to force UNP to ask if it should do something. Currently only the -K switch supports this. Options which are not followed by one of the mentioned characters work as toggles, which means that using an option twice will undo the previous (eg. -a -a has no result). However once an option has been turned on with the question mark (like -a?) then you can only turn it off by appending a - (like -a-). Still got it? :)

-? = help (this screen)
Suprisingly enough, this switch will let UNP show the built-in helpscreen. Any other switch or command used on the same line will be ignored.

-a = automatic retry
It is possible that some files have been processed with some program more than once. This switch will make UNP to process the file again when it was changed. Useful when you want to uncompress a file which also has been Immunized by CPAV.

-b = make backup .BAK file of original
If you want to keep a backup of your original file (very wise) use this switch. The original file will be renamed to a file with a .BAK extension.

-c = ask for confirmation before action
This will force UNP to ask you if you want to remove the routine UNP found on the file each time it has recognized some program's work.

-f = optimise fixups (like HDROPT.EXE)
Relocation items, also known as fixups, are stored in the .EXE header in two parts; 16 bits for the segment value and another 16 bits for offset. Since DOS only uses 20 bits for addressing, the fixups may contain some redundant data. Optimising the fixups does some arithmetic stuff which will move as much as possible of the address into the offset and fills the segment value with zeros. This is the same as the program HDROPT.EXE supplied with PKLITE does.

-g = merge overlay into image
This dirty switch allows you to merge an overlay into the image of an .EXE

file. I can't think of any reason why someone should use it but it's here.

-h = remove irrelevant header data
Most linkers add useless data to the .EXE header. This switch removes all such useless information, thus shrinking the header size. This switch also skips the header rebuilding code with files like PKLITE.

-i = interception of I/O interrupts
By default UNP watches several DOS interrupt to check if the program is running as expected. Any unexpected call to such an interrupt will make UNP abort the process. If you have any weird TSRs resident you might have to use this switch.

-k = pklite signature handling; - = don't add, + = add always, ? = ask
With this switch you can handle the pklite signature. There are 3 possibilities :
-k- = don't add
The pklite signature will not be added, this will also be the case if you only use -k (to stay dislite compatible).
-k+ = add always
Always add the pklite signature, this is the default of UNP so you can just as well leave the -k switch away if you want this.
-k? = ask
When you use this, UNP will ask you each time it has found a signature (like UNP V3.01 or earlier did).

-l = use large memoryblock
When UNP loads a program it allocates a block with a size of the required memory with about 32k extra for safety. Some programs require even much more memory than they tell DOS they need. If such a file is decompressed by UNP it definately will go wrong. Two things can happen in such a case. The program detects the absence of enough memory and will attempt to notify the user by writing a message on screen. This will probably result in a "(INT 21) Unexepected call to DOS" error (see messages) and UNP will abort gracefully. Or worse, the program does not check at all and will try to decompress anyway. This will probably result in a system crash or a memory allocation error. If you have got a file which requires more memory than it tells DOS, use this switch. After identifying the compressed program, UNP will increase the allocated memory block to 15/16 of the maximum size of that block.

-m = MORE alike output
On request this switch has been added. It should pause about every screen full of information similar like DOS's MORE.EXE.

-n = numbered Outfiles
Also on request is the possibility the have UNP remove several routines in one run but keeping a copy of every version. This switch will assign a number to the files it writes the new file to. If the file already is numbered, it will increment that number. If not, the number 1, possibly with leading underscores,will be assigned to it.

-o = overwrite output file if it exists
If you want to have the destination file overwritten, you can avoid the question for permission by specifying this switch on the command line.

-p = align header data on a page
It is said that .EXE files with a header size that is a multiple of 512 bytes load faster (this could make sense since a sector is also 512 bytes). This switch will expand the header to the nearest multiple of 512 bytes, filling it with zeros.

-r = remove overlay data
If something is appended to an .EXE it is called an overlay. This switch will let the file size of the outfile be the same as the load image. So anything that was appended to the file will be thrown away. An overlay can be used for all kinds of data, so removing this can result in throwing away something useful.

-u = update file time/date
By default UNP sets the time/date of the destination file to the same

time/date as the original source file. If you want to have it updated to the current time/date use this switch.

-v = verbose
When you use this switch UNP will give you some additional information. I added this switch for debugging purposes.

-- = program's commandline
Anything after this switch will be passed to the program to be decompressed. This way you can pass along any required parameters (like passwords) for the Tracing command.

Messages

UNP has 6 kinds of messages other than the usual information it can display:

- Questions - Even with new smart routines programmed into UNP4 it still needs to ask a few things now and then. Who said that computers are smarter than you? Anyway, you can expect the following questions:

Add code to fake PKLITE decompression (y/n)?
This question arises when UNP detects that a signature has been placed into the program's PSP and the -K switch has the '?' value. (for more info, read the "notes on compressors" part)

Continue (y/n)?
When UNP considers a program abnormal it wil display a warning with the reason why it thinks so and will ask you if you want to continue anyway.

Remove this routine from file (y/n)?
You have requested confirmation for each action UNP takes (see -C option) and this is the result.

Program is protected, please enter password:
Some programs have the ability to scramble executable files with a password. Unfortunately I have not succeeded in breaking all protection schemes using this. So for certain programs you might be prompted for the password

File FILENAME.EXT already exists. Overwrite (y/n)?
When UNP wants to write to the destination file and discovers the file already exists, it will ask if you want to overwrite the file. You can avoid this question by using the overwrite option (see -O option).

- Informal messages - By placing UNP in verbose mode (see -V option) UNP will display additional information about anything that might be interesting. Note that informal messages allways start with "INFO -".

DOS Version X.XX[, running under Windows.]
Some system information, this has no effect on UNP.

Commandline = ...
This indicates what options are passed for the Init procudere the the main module. This is influenced by UNP's commandline.

Program's commandline = "...".
If you have specified anything for the program's commandline (see -- option), it will be echoed here.

Using FILENAME.EXT as temp file.
The name of the temporary file UNP will use. This is composed of the TEMP environment variable and some constant defined in UNP.

Anti-virus program TbScanX detected.
UNP has detected the resident anti-virus program TbScanX and will use it to scan the files before it loads them (also see -s switch).

Wildcard matches X filename(s), stored at XXXXh.
The wildcard specified on the commandline is resolved to a number of files and these names has been stored at the specified segment.

Program loaded at XXXXh, largest free memory block: X bytes.
   Indicates at which segment UNP is loaded and how large the largest
   available memoryblock is.

Adding 'PK' signature to fake PKLITE decompression.
   When UNP automaticly adds the code to fake PKLITE decompression
   (see -K option), it will display this message.

Increasing program''s blocksize to X bytes.
   In certain cases UNP will increase the memory given to the program which
   UNP wants to decompress. This can solve problems with programs which
   do not check if they have enough memory. This can be forced with the -L
   option (see -L option).

- Warnings - These messages indicate something is wrong but UNP can live with it. Warnings will always start with "WARNING -".

Infile and Outfile are same, Outfile ignored.
   After UNP has resolved the wildcard it has found out the the file to be
   processed is the same as the destionation file. Since this is the default
   operation of UNP it will ignore the destination file.

Outfile specified, -B option ignored.'
   When you have specified a destination file you can't create a backup. This
   is because the backup is created by renaming the original file. When the
   destination file is also specified there would be no original program
   left.

-N option overrules -B option, -B option ignored
   You can't number your files and have a backup created as well. It's about
   the same reason as mentioned above.

Invalid or missing stored header information.
   Some files store the original header somewhere inside the compressed file.
   When UNP has detected this and the info does not seem to be correct it
   will display this warning.

- Errors - UNP has discovered something wrong and cannot continue with the current action. It will continue with the next file (when available).

(INT 10h) Unexpected use of video interrupt, action failed.
(INT 20h) Unexpected program termination, action failed.
(INT 21h) Unexpected call to DOS, action failed.
   UNP watches several interrupts to ensure things are going as expected.
   When UNP loses control it will sooner or later detect one of the
   interrupts it watches and abort the current action. If you think nothing
   went wrong and you got this message anyway, you can disable the interrupt
   watching (see -i switch).

Cannot convert, file already is a COM file.
Cannot convert, file has relocation items.
Cannot convert, initial CS:IP not FFF0:0100.
Cannot convert, file is too large for COM.
Cannot convert, file contains internal overlay.
   Convertion of a .EXE file to .COM file has to meet several conditions.
   When one of these is not met the program will show which one and abort the
   action.

- Dos error - Your operating system does not allow something UNP would like to do. Simple things like a read-only file or disk full will cause such a error. UNP will quit if such an error is encountered. These messages start with "DOS ERROR - " and end with the DOS error code.

unable to open file ... (error x)
unable to create file ... (error x)
unable to read from file ... (error x)
unable to write to file ... (error x)

- Fatal errors - Something seriously wrong has happened. The program will abort. These messages will start with "FATAL ERROR - ".

No files found matching

UNP could not resolve the wildcard you specified on the commandline to any file. You might want to check the filenames.

Decompressing many files into one.
The Infile wildcard matches more than one file and you have also given a destination filename on the commandline.

Output path/file must not contain '*' or '?'.
You can't use wildcards in the destination filename.

Outfile required for specified command.
The command you specified requires 2 filenames and you only gave one.

Specified command does not require filenames.
The command you specified does not allow any filenames at all!

(INT 00h) Divide overflow generated by CPU.
(INT 23h) Ctrl-C or Ctrl-Break pressed by user.
These interrupts are considered very important and UNP will quit as fast as possible when one of these occur.

Not enough memory to ...
UNP could not allocate enough memory for something.

Memory Control Blocks destroyed.
UNP now checks for this special memory error since this error is probably caused by a progam that has been giving too few memory. UNP will abort but the system will most likely halt immediately after that. You might want to try giving the program more memory (see -l switch).

Notes on compressors

There are a few things about compressors that might usefull to know:

AVPACK V1.20
This Russion compressor has many similaritys with PKLITE. The PKLITE routines are used to unpack this compressor. However, it is not as good as PKLITE. It reports that files with a size which is a multiple of 512 bytes contains an overlay and also it only stores the first 20h bytes of an exeheader making it impossible to do a complete restore. It does have some extra option like encryption. UNP can uncompress encrypted files like these although it does not recognize them as such. If you know you got an encrypted AVPACKed file you can use "UNP T" to unpack.

COMPACK V4.4
This program does not really contain a bug but more an incompatibility error. On 486s, programs compressed with this version of COMPACK will crash. This is a result of the self-modifying code COMPACK uses. Somewhere at the end of the decompression routine of COMPACKed programs there is a far jump to the decompressed program. Initially this jump points to 0:0 but is adjusted not much earlier before the execution of this instruction. On 386s or lower the prefetch queue is small enough to allow this self-modifying code. On 486s however, the read-ahead buffer is much larger so the jmp has already been read when the adjustment takes place. The result on 486s is that the jmp 0:0 is actually executed, most likely causing a system crash. UNP places a breakpoint before the execution of this instruction which flushes the read-ahead buffer and the program can be saved with the correct entrypoint.

EXEPACK
Ever got the message "Packed file is corrupt"? Then you are probably using a memory manager and have lots of conventional memory free. Old versions of Microsoft's EXEPACK require atleast one segment (64k) below it to be able to unpack the program into memory. If you have a lot of free memory, let's say above 600k, then programs can be partially loaded in the first segment. This causes EXEPACK to generate this error. UNP loads an EXEPACKed file high enough to unpack it and can decompress it without any trouble.

MEGALITE V1.5
Like AVPACK, this compression looks very much like PKLITE. This version

however contains an instruction which changes 1 byte in the decompression routine. I have not been able to find out what the use of this instruction is. All it seems to do is screw up the code. The instruction which causes this is: DEC BYTE PTR DS:[SI+012Ch].

MR-LITE
This utility seems to be floating around in certain circuits. It is written to reduce the size of PKLITE size even more. All it does is simply rewrite the header and leave all useless information away. In fact, it does the same as "UNP l -h". Unfortunately it does not do this very well. One of the fields in the .EXE header reports the amount of memory required by the application. This value is kindly set to 0 by MR-LITE. Because DOS by default allocates all memory available, you will not immediately detect this bug. But when unpacking it with UNP you will very likely get the message "Memory Control Blocks destroyed.". It is adviced to unpack such a file with "UNP E -l" and if you want recompress it, optionally you can optimize the header with a "UNP L -h". (for more info, see -l switch)

PKLITE V1.00á
Although this program is probably rarely used, I implemented some code that fixes a bug that appears in this version of PKLITE only. When certain programs are compressed, PKLITE moves the last 512 bytes of the image into an overlay. Compressed programs will be decompressed by UNP and checked for an overlay of 512 bytes. If such an overlay has been found, UNP includes the overlay into the newly created image. This has the same result of what would have happened when "PKLITE -x" would have been used to restore to program.

PKLITE V1.14+ Professional
These versions of PKLITE have some small piece of code in the decompression routine that adds a so called signature into the PSP. This allows programs to check if they are still compressed with PKLITE. When such a program is unpacked UNP by default adds a small piece of code into the PSP to fake the decompression. One of the programs that check for such a signature is the PKZIP V2.04g program. (see also -k switch)

PKLITE V1.15
This version does not seem to detect OS2 or Windows files anymore and will compress them like normal EXE files. Files will however not run correctly, even when UNP has uncompressed them again.

PKTINY
A small utility has been written to prevent recognition (and unpacking) of TINYPROGed files. The trick this program uses is very simple. TINYPROG has the ability to leave some space in the beginning of an .EXE file. By filling this space with a PKLITE header and modifing some code to let the program still run correctly, it tries to fool unpackers. If UNP detects the modified code it tries to get around it and continue with the TINYPROG check.

SHRINK V1.00
This compressor uses the basic RLE (Run Lenghth Encoding) compression algorithm to decrease the size of a program. Unfortunately the program contains (at least) 2 bugs. One of the bugs is when the RLE byte is found followed by a 00 while decompressing, a 00 is placed in the program which should be the RLE byte. The second bug is that the last byte of the compressed file is not written to disk. Both of these bugs are triggered when all 256 bytes appear at least one time in the file. UNP is able to correct the first bug, causing most program to work again. However the second bug is unrecoverable and UNP give a warning if it detects this bug. It is always better to decompress it, even if the last byte is missing.

Registering UNP

Having tried several forms of registration for UNP, I have decided to use the following registration method. First, since a lot of support has come from the low end user I decided to release UNP as cardware to the public domain. It's always nice to know your program is appreciated, and what's the price of a simple card compared to the registration fees asked by several others? So if you're a happy user of UNP fill in your registration postcard of

something in your neighbourhood today. However, I have spend a lot of hours on this program and since it can be useful for commercial purposes I decided that for commercial use a registration of $1 per copy is required. Why so cheap you might wonder. Well, I don't want the price to prevent you from registering. I do not have to make profit out of it, I am just a student who has written a program to teach myself more about DOS. I just as well could have been writing viruses but instead I have chosen this. Please note that non-commercial users are allowed to send me money anyway! If it is enough to buy and mail a disk, you can expect a free special registered version!

Hey! Unp is compressed!

Yes, starting with V4.11 of UNP I will use a compressor to make sure lamers won't just change the version number and upload it to some BBS just to get their ratio higher. UNP is compressed with DIET V1.45f and processed with a program I call DSHIELD to prevent decompressing. The traps used are not too difficult to figure out, but the idea behind it was just the prevent the lamers from hacking. If you succeed in unpacking it, then you are probably an experienced programmer. I am sorry but the protection seems to be neccesary.
Due to this protection it might be possible that some anti-virus programs which use heuristic scanning consider UNP infected by a new or unknown virus. If you also use the D command to alter (some of) UNP's default settings, you might get a warning as well. The D command causes UNP to alter it's own .EXE file. Check your documention that came along with your anti-virus software on how to solve this incompatibility.

What UNP can remove

Quite a lot actually. A list follows:

AINEXE V2.1
ANTIBODY
AVPACK V1.20
AXE V2.2
CENTRAL POINT ANTI-VIRUS V1, V1.1
COM2CRP V1.0
COMLOCK V0.10
COMPACK V4.4, V4.5
CRYPTA V1.00
CRYPTCOM
DELTAPACKER V0.1
DIET V1.00, V1.00d, V1.02b, V1.10a, V1.20, V1.44, V1.45f
ENCRCOM V2.0
EPW V1.2, V1.21, V1.30
EXELITE V1.00aF
EXEPACK V4.00, V4.03, V4.05, V4.06
F-XLOCK V1.16
ICE V1.00
IMPLODE V1.0 Alpha
KVETCH V1.02á
LINK /EXEPACK V3.60, V3.64, V3.65, V3.69, V5.01.21
LZEXE V0.90, V0.91, V1.00a
MCLOCK V1.2, V1.3
MEGALITE V1.18a, V1.20a
OPTLINK
PACKEXE V1.0
PACKWIN V1.0à
PASSCOM V2.0
PGMPAK V0.13, V0.14, V0.15
PKLITE V1.00á, V1.00, V1.03, V1.05, V1.12, V1.13, V1.14, V1.15, V1.20, V1.50
POJCOM V1.0
PRO-PACK V2.08, V2.14
PROCOMP V0.82
PROTECT! EXE/COM V1.0, V1.1, V2.0, V3.0, V3.1, V4.0, V5.0
SELF-DISINFECT V0.90á
SHRINK V1.0
SCRNCH V1.00, V1.02
SYRINGE
TINYPROG V1.0, V3.0, V3.3, V3.6, V3.8, V3.9

TURBO ANTI-VIRUS V7.02A, V9.40
UCEXE V2.3
USERNAME V2.00, V2.10, V3.00
WWPACK V3.00, V3.01, V3.02

I have left out a couple of names not really worth mentioning.

What UNP cannot remove

SPACEMAKER V1.03
EPW V1.2, V1.21, V1.30 - EXE only
USERNAME V2.00, V2.10, V3.00 - EXE only

CONTACTING ME

Please note that registrations must be send to my home adress, not to my E-mail adress. A card really is a card, not a scanned picture or some piece of text.

My address:
Ben Castricum
Van Loenenlaan 10
1945 TX Beverwijk
The Netherlands

E-Mail: valid until june '95
benc@htsa.hva.nl

I am not sure when exactly my account will be disabled, but I'll try to get a new account somewhere as soon as possible.

THE HELP SCREEN

The Help screen has the following contents. You can use this as a quick-reference card

```
*******************************************************************************
UNP 4.11 Executable file restore utility, written by Ben Castricum, 05/30/95

usage: UNP command [options] [[d:][\path]Infile] [[d:][\path]Outfile]

  commands:
c = convert to COM file                m = MarkEXE, insert a file in header
d = make current options default       o = copy overlay
e = expand compressed file (default)   s = search for compressed files
i = show info only                     t = trace executable
l = load and save                      x = convert to EXE file

  options followed by their current setting
-?-= help (this screen)                -l-= use large memoryblock
-a-= automatic retry                   -m-= MORE alike output
-b-= make backup .BAK file of original -n-= numbered Outfiles
-c-= ask for confirmation before action -o-= overwrite output file if it exists
-f-= optimize fixups (like HDROPT.EXE) -p-= align header data on a page
-g-= merge overlay into image          -r-= remove overlay data
-h-= remove irrelevant header data     -u+= update file time/date
-i+= interception of I/O interrupts    -v-= verbose
-k+= [-|+|?] pklite signature handling -- = program's commandline
*******************************************************************************
```

-- End of UNP V4.11 documentation --