TCP/IP BOOT-PROM Changes to User's Manual 1.50

This manual was written, illustrated, and produced using FrameMaker publishing software on a NeXT workstation. Proofs were rendered by a NeXT computer using the Dots software and printed on a 600 dpi printer. Final pages were transferred directly from a NeXT disk to film.

Project team: Malcolm Agnew, Ralf Büttner, Dirk Köppen

Printed in Germany.

21.Feb 1994, covers TCP/IP BOOT-PROM software version 1.50.

Changes

This manual describes changes made to the *TCP/IP BOOT-PROM User's Manual* version 1.50 dated 22.Dec 1993. The changes are:

- New support BBS address.
- New options have been added to the BPUTIL program.
- Options have been removed from the BPSHELL program.
- A new program BPDISK has been added to the distribution.
- A new program BPSECURE has been added.
- New features of the NetWare BOOTPD.NLM.

Using HIMEM.SYS on MS-DOS 6.2

Take care to include the /TESTMEM:OFF switch if the HIMEM.SYS of MS-DOS 6.2 is used. Otherwhise the memory test will overwrite the RAM disk image. A sample usage is:

```
device=a:\dos\himem.sys /testmem:off
```

The TCP/IP BOOT-PROM support BBS

Updates and documentation can be obtained from the TCP/IP BOOT-PROM *BBS*:

- Via modem at +49 69 9894 1025.

New BPUTIL options

The BPUTIL.COM and BPUTIL.SYS programs have been enhanced by adding more tags and a new method of argument passing.

Arguments to BPUTIL

All arguments to BPUTIL can now be passed using tags. If BPUTIL finds

that an argument includes the special ID (#@) it first looks up that tag in the BOOTP reply and then uses the resulting string as its argument.

For example:

```
bputil -t 193.141.47.197 /tftpboot/network.drv network.drv
```

Transfers the file /tftpboot/network.drv from the server 193.141.47.197 to the current drive. You can now specify the name of the file to be transferred by storing the filename into a tag in the BOOTP reply:

```
bputil -t 193.141.47.197 #@t130################ network.drv
```

BPUTIL will then first look up the value of the tag t130 and use its contents as the filename to be transferred. This can be used for variables in CONFIG.SYS as CONFIG.SYS can not be patched before execution. Also the network driver to be loaded can be remote configureable:

```
device=bputil.sys -t #@sip############ #@t130############# netdrv.sys
device=protman.sys
device=netdrv.sys
```

Note that the file netdrv.sys must already exist as a dummy or empty file before you can overwrite it using the -t option. Otherwise MS-DOS will not find the file from CONFIG.SYS.

Transferring files from a TFTP server to a diskless client

The -t option of BPUTIL has been enhanced by an optional gateway argument. If a gateway is to be used, it must precede the TFTP server IP address:

```
bputil -t 193.141.47.199 128.1.3.12 /tftpboot/foo foobar
```

This will download the file /tftpboot/foo from TFTP server 128.1.3.12 using the gateway or router 193.141.47.199. If no gateway is given, BPUTIL expects the TFTP server to be on the local subnet:

```
bputil -t 193.141.47.197 /tftpboot/foo foobar
```

The gateway can be determined by the BOOTP reply. The #@gw0 tag holds the first gateway IP address that is given by the BOOTP reply:

```
bputil -t #@gw0############## 128.1.3.12 /tftpboot/foo foobar
```

If no gateway was given, the #@gw0 tag is empty.

New vendor tags

The following tags have been added to `BPUTIL`:

tag	RFC1048 BOOTP vendor field
`#@ts0#`	First time server IP address.
`#@tsf#`	All time server IPs, separated by spaces.

tag	RFC1533 BOOTP vendor field
`#@bca#`	Broadcast IP address.
`#@nid#`	NIS domain name.
`#@ni0#`	First NIS server IP address.
`#@nif#`	All NIS server IPs, separated by spaces.
`#@nt0#`	First ntp server IP address.
`#@ntf#`	All ntp server IPs, separated by spaces.

New tag alignment

The alignment of the string which is generated out of the tags contents can be specified. Additional to the alignment specifiers + and –, `BPUTIL` can now replace tags by moving all unnecessary spaces to the end of the line. The new specifier `*` (asterisk) is used for that.

Note - `BPUTIL` does not change the size of a patched file to be compatible with DOS when `BPUTIL` is executed in `CONFIG.SYS`. Therefore all spaces are moved to the end of the line.

The following examples demonstrate the usage of the `*` specifier:

```
#@chn#########|
#@chn-########|
#@chn+########|
#@chn*########|
#@chn+########|#@chn-########|
#@chn+########|#@chn*########|
My hostname is #@chn*########. I am diskless !
Concatenated: #@chn*#########@chn*########
```

After patching this file using `BPUTIL -a`, the resulting file looks like:

```
diskless      |
diskless      |
      diskless|
diskless|
      diskless|diskless      |
      diskless|diskless|
My hostname is diskless. I am diskless !
Concatenated: disklessdiskless
```


Changes to the BPSHELL program

The `BPSHELL` program does no longer support the `Optimize` option within the `BootImage->Create` window and the `Format` option within the `BootImage->Restore` window.

The new program `BPDISK` can be used to create, restore and modify boot images.

New program BPDISK

A new program `BPDISK` has been added to the TCP/IP BOOT-PROM distribution. `BPDISK` can be used to create, restore and modify boot images without the need of a diskette drive. `BPDISK` can directly read and modify DOS filesystems in the boot image.

The features of `BPDISK` are:

- Can insert or extract single files or complete subdirectories from a boot image.
- Can create optimized boot images which only allocate the storage space of the included files.
- Can create boot images of all common MS-DOS diskette formats including 1.2, 1.44 and 2.88 MB images.
- Can be used in DOS batch files to create or update multiple boot images.
- UNIX version is available as add-on.

Sample BPDISK usage

To create a new boot image use the `-F` option. The following example will create a 1.44 MB type boot image `ramd.X` on drive `F:` but allocates only the space of the included files. The bootsector and system programs like `IO.SYS`, `IBMBIO.COM` and `COMMAND.COM` are read from diskette drive `A:`, but may also be read out of an already existing boot image:

```
C:\> bpdisk -d f:\ramd.X -F 1440,A:
```

The next commands copy the file `C:\BPROM\AUTOEXEC.BAT` into the boot image. The file `C:\BPROM\START.BAT` is copied into the subdirectory `\NETWORK` in the boot image. If the subdirectory does not exist, `BPDISK` will create it:

```
C:\> bpdisk -d f:\ramd.X -I \autoexec.bat,c:\bprom\autoexec.bat
C:\> bpdisk -d f:\ramd.X -I \network\start.bat,c:\bprom\start.bat
```

Files can also be copied out of a boot image:

```
C:\> bpdisk -d f:\ramd.X -O \network\start.bat,c:\tmp\foobar
```

To delete a file out of a boot image use:

```
C:\> bpdisk -d f:\ramd.X -E \autoexec.bat
```

Complete subdirectories can be copied into the boot image:

```
C:\> bpdisk -d f:\ramd.X -i \network
```

To view the contents of a boot image execute:

```
C:\> bpdisk -d f:\ramd.X -D
```

Note - All options given to `BPDISK` are case sensitive !

The -C option

To erase all files from a boot image the `-C` option can be used. The following example will remove all files from a boot image:

```
C:\> bpdisk -d ramd.X -C
```

The -d option

This option points to the filename of the boot image. The filename can be an already existing image or an image that should be created. The filename can also include a drive letter e.g. `f:\tftpboot\ramd.X`.

If the `-d` option is not given, `BPDISK` takes the image that is defined by the environment variable `BPDISK`.

The -D option

The `-D` option shows a recursive directory listing of all files in the boot image given by the `-d` option. Example:

```
bpdisk -d ramd.X -D
```

The -E option

This option removes a file from the boot image. This example deletes the file `autoexec.bat` from the boot image `ramd.X`.

```
bpdisk -d ramd.X -E autoexec.bat
```


This deletes the file `hosts` out of the `etc` directory from the boot image:

```
bpdisk -d ramd.X -E \etc\hosts
```

The -F option

This option formats (creates) a new boot image. The name of the boot image to be created must be given by the `-d` option or by environment variable `BPDISK`. If the boot image already exists, it will not be overridden.

The first argument specifies the format of the boot image. This can be one out of: `160`, `180`, `320`, `360`, `640`, `720`, `1200`, `1440` or `2880`. `BPDISK` will not allocate the space needed for the complete boot image, this will be done later when files are copied into the boot image.

The second argument points to a file or a drive which holds the bootsector and the system files for the boot image. The two arguments must be separated by a comma.

Example:

```
bpdisk -d ramd.X -F 1440,A:
```

This creates a 1.44 MB type boot image which takes the bootsector and system files from the diskette drive `A:`. The bootsector and system files can also be read out of an already existing boot image:

```
bpdisk -d ramd.X -F 1440,f:\tftpboot\dos.X
```

If the following system files exist on the diskette drive or the existing boot image, they will be copied into the new boot image:

- `IO.SYS` and `MSDOS.SYS`
- `IBMBIO.COM` and `IBMDOS.COM`
- `COMMAND.COM`

The -I option

This option copies a file into an existing boot image.

The name of the file in the boot image is given by the first argument, the source of the file that is to be copied into the boot image is given by the optional second argument. If the second argument is missing, the name of the first argument is taken for the second argument except that the path is removed.

Example:

```
bpdisk -d ramd.X -I config.sys,config
```

This copies the file `config` into the boot image using the name `config.-sys`. The following examples do the same:

```
bpdisk -d ramd.X -I \config.sys,config
bpdisk -d ramd.X -I /config.sys,config
```

This copies the file `c:\files\hosts` into the boot image using the name `\etc\hosts`:

```
bpdisk -d ramd.X -I \etc\hosts,c:\files\hosts
```

This copies the file `hosts` out of the current directory into the directory `etc` in the boot image. Note that the path of the destination has been removed in order to make up the name of the source:

```
bpdisk -d ramd.X -I \etc\hosts
```

The -i option

This copies all files of the directory given by the argument and the files in its subdirectories into the boot image. If you want to copy all files from the current directory use `.` (dot) as the argument.

Example:

```
bpdisk -d f:\tftpboot\ramd.X -i .
```

Copies all files and directories out of the current directory into the boot image at `f:\tftpboot\ramd.X`.

Example:

```
bpdisk -d ramd.X -i a:\
```

Copies all files from diskette drive `A:` into the boot image `ramd.X`.

The -M option

The `-M` option creates a new subdirectory in the boot image.

These examples create a new subdirectory `dos` in the boot image:

```
bpdisk -d ramd.X -M dos
bpdisk -d ramd.X -M \dos
bpdisk -d ramd.X -M /dos
```


This creates a new subdirectory `subdir` in the already existing directory `dos`. All parent directories must exist:

```
bpdisk -d ramd.X -M \dos\subdir
```

The -O option

As option `-I` but copies out of the boot image.

The -o option

As option `-i` but copies out of the boot image.

The -P option

The `-P` option pads (adds) dummy data to the end of the boot image. This is useful to create space if a program will write files to the boot image.

This creates 64 KB additional space to the boot image:

```
bpdisk -d ramd.X -P 64
```

The -T option

The option `-T` shows the contents of a file in the boot image given by the `-d` option.

Example:

```
bpdisk -d ramd.X -T autoexec.bat
```

This copies the contents of the file `autoexec.bat` out of the boot image to the screen.

The following example copies the contents of the file `hosts` in the subdirectory `etc` out of the boot image to the screen:

```
bpdisk -d ramd.X -T \etc\hosts
```

The -v option

The `-v` option gives more technical information about `BPDISK` operations. It can be used together with all options or alone.

The following example shows information about the boot image structure when the boot image is created:

```
bpdisk -d ramd.X -F 1440,A: -v
```

New program BPSECURE

If the network boot failed and the system has been booted from the internal hard disk, the user is able to execute programs from the hard disk. This can be prevented by using the `BPSECURE` program.

`BPSECURE` is padded to the DOS `EXE` or `COM` program and executes the program only if a TCP/IP BOOT-PROM and a RAM disk exists. You can install `BPSECURE` as in the following examples:

```
C:\> copy /b bpsecure.bin+net.exe net.exe
C:\> copy /b bpsecure.bin+login.com login.com
```

Note - The current implementation of `BPSECURE` allows the execution only if a RAM disk exists. It does not execute the padded program after `BPUTIL -r` has been executed.

New features of the NetWare BOOTPD.NLM

If the NetWare `BOOTPD.NLM` can not find an interface to send the BOOTP reply to, it broadcasts the reply over all interfaces which have been enabled by the `-i` option..

The length of the command line on NetWare is limited, also the editor can not hold that many characters in one line that are needed to pass all arguments to `BOOTPD.NLM` if multiple network adapters are used.

A new option `@` can be used to pass a filename to `BOOTPD.NLM` that holds all arguments. This is a sample file named `sys:\etc\bootpd.arg`:

```
-d -d -d -d -t 0 -s
-i 193.141.47.195/255.255.255.240
-i 193.141.47.211:3 -i 193.141.47.219/255.255.255.248
```

Arguments can be separated by new lines ore spaces. `BOOTPD.NLM` can then be loaded using the following command:

```
load bootpd @sys:\etc\bootpd.arg
```

TCP/IP BOOT-PROM
Changes to
User's Manual 1.50