GENERAL SYNTAX RULES

Any characters following '*' or ';' are regarded as comments and are ingnored by the system. When processing the configuration file, the system does not distinguish between small and capital letters. All numeric values inside the configuration file may be written either in decimal or in hex modes. A hex numeric constant should always begin with a number and end with 'H' or 'h'. The feature is useful when defining memory segments addresses and determining key scan-codes in some commands.

At the beginning of any CONFIG.xxx file line the following prefixes or combinations of those may be used: CPU88, CPU86, CPU286, CPU386, CPU486, PC, AT, XT, PS1, PS2. These prefixes mean "for the xxx processor" and "for the xxx type of computer", respectively. It is possible to use '+' at the end of the CPUxxx prefixes, meaning "for the xxx processor and higher".

CPU88 - CPU486	- processors i88 - i486 and compatible
PC	- original PC
XT	- IBM PC/XT compatible
AT	- IBM PC/AT compatible
PS1	- IBM PS/1
PS2	- IBM PS/2

For example,

```
     CPU386+ PS2 echo "Running on PS/2 i386/i486/P5/.. computer"
```

system, when processing the above line, will echo:

```
     Running on PS/2 i386/i486/P5/.. computer
```

Some commands have optional parameters. A special symbol is used as a rule to denote such a parameter. By default, the symbol is '/'. It is useful to keep in mind that the key symbol may be defined (changed) with the help of SWITCHAR (sys) command. Once done so, the new symbol should be used instead of '/', for example:

```
          SWITCHAR = -
          DRIVPARM = -d:0-c
```

As the system makes several passes when processing the configuration file, the SWITCHAR command should be located at the very beginning of the file.

The following conventions are used for syntax description:

```
     <...> - obligatory parameter;
     [...] - non-obligatory (optional) parameter;
     |     - "or" - shows the alternatives.
```

For example,

```
        DRIVATTR </D:dd> <[/T][/C][/S]>
```

implies that the argument of the command is a line always containing '/D:dd' and at least one of '/T' ,'/C', '/S'.

```
        BREAK = <ON|OFF>
```

BREAK command should have either 'ON' or 'OFF' as an argument. Also, 'YES' and 'NO' respectively can be used in the above case.

CONFIGURATION FILE SECTIONS

The configuration file consists of commands which may form several sections. The system itself processes only the user-chosen or predefined sections. Each section begins with a definition of its name and ends either with a next section name definition or with an end of file. Section name should be put in the '[' ']' square brackets. Comments within a section name should begin with '@'. For example,

```
        [ SectionName @ This is a comment ]
```

Section name is defined by its first 16 characters. All the others, including any spaces, are ignored.

A section name should be unique, except for the special names like: COMMENT, PTSDOS, COMMON. If more than one section has the same name, the system treats the resulting set as a whole (does not apply to the TRACE).

The following section names have been predefined:

```
COMMENT        - a section with this name is ignored by the system.
PTSDOS, COMMON - commands within such a section are considered global
                 and are always executed by the system.
BOOTMGR        - BOOT MANAGER parameters description section.
```

CONFIGURATION FILE PROCESSING

When booting, the system first loads the boot manager for processing sections and some commands like VIDEOMODE etc. The boot manager tries to recognize all the sections in a configuration file and, if it finds selectable sections, displays the list of these sections with a prompt to select one of them for booting. This boot menu can contain up to ten selectable sections to be user-selected. The sections COMMENT, PTSDOS, COMMON, BOOTMGR and sections with names beginning with '$' can not be selected from a boot manager prompt, but they can be enforced by FORCE command. The boot menu doesn't appear, if there are none of the selectable sections. Mode of operation of the Boot Manager can be defined with the following commands in the BOOTMGR section:

AUTOCHECK = <ON>|<OFF> ;

By default, AutoCheck = Off. When "ON", boot manager will use the user's input prerecorded into a file or CMOS memory, instead of a dialogue.

STORAGE = <FILE>|<CMOS> ;

By default, Storage = FILE. At the first call or on pressing <Esc> during booting, Boot manager will display a prompt for user selection. If a user presses '1','2',...,'0', the choise will be realized but not saved for further bootings. If any of the above keys is pressed in combination with 'Alt', then the Boot Manager will ask the permission to process each line of the configuration file (See TRACE command). If the user inputs 'F1' - 'F10', the boot manager will save the number either in a file or in CMOS, depending on STORAGE.

FILENAME = [drive:][path]filename

Specifies the filename for storing users input. By default, FILENAME = AUTOBOOT.SYS

CMOSADDR = <CMOS register address in decimal format>

Specifies CMOS register for storing users input. By default, CMOSADDR = 52

FORCE = <section name>[,<section name>[, ...]]

Specifies a section or a list of sections, equivalent to those named "COMMON".

IGNORE = <section name>[,<section name>[, ...]]

Specifies a section or a list of sections which are to be ignored by the system.

```
COLOR  = <cc | < BLACK..WHITE [,BLACK..WHITE]>>
         [on]
         [ bb | BLACK..WHITE [,BLACK..WHITE] ]
```

colors -
BLACK,BLUE,GREEN,CYAN,RED,MAGENTA,BROWN,LIGHTGRAY
DARKGRAY,LIGHTBLUE,LIGHTGREEN,LIGHTCYAN,LIGHTRED
LIGHTMAGENTA,YELLOW,WHITE
Where, cc - foreground color, bb - background color.
Specifies boot manager prompt color.

DEFAULT = <section name>

TIMEOUT = <period in seconds>

By default, TIMEOUT = 5 (sec). DEFAULT specifies a user selection 'by default'. TIMEOUT gives a time delay before the section is selected according to the DEFAULT command.

Surely enough, the system will immediately make the right choice if the AUTOCHECK=YES and correct user information does exist. However, if such information does not exist or for any reasons (for instance, due to any modification in CPNFIG.xxx file) is unacceptable, then a standard prompt shall be given by the system, where the default selection will be marked with '+'. If no measures are taken after the TIMEOUT period or if you press [Enter], the system will load the marked section, but this choice will not be saved for further booting.

CONFIGURATION FILE PROCESSING (continued)

By entering the global trace mode (processing each line of the configuration file after pressing Alt-1..9 in the BootMgr prompt) it is possible, by pressing Alt-Y or Alt-N to answer all the prompts related to the global trace mode (no effect on the TRACE command itself).

If in the [BOOTMGR] section it is specified:

```
storage=FILE
autocheck=YES
```

then the date of CONFIG.xxx creation will be stored together with information on the selected section. So, after any modification of the file, the system will consider the section selection information as incorrect and will assume that it should be rewritten.

EXAMPLES OF CONFIGURATION FILE

One may, in absence of compatibility with MS-Windows and other programs, and by using non-documented DOS features, free more memory for applications by modifying system configuration. For example, let us consider the following configuration file:

```
  WINDOWS = NO            ; Enabling usage of non-standard system data
                          ; and code allocation strategy
  COMAREA = STANDARD      ; Usage of memory area starting from 50:00
  DOS     = HIGH          ; HMA enabling
  DEVICE  = HIMEM286.SYS  ; XMS driver loading
  FILES   = 40            ; Maximum number of open files
  BUFFERS = 20            ; Number of input/output buffers
```

On starting-up System and running MEM, the following message will be displayed on the screen:

```
        655,360 bytes total conventional memory
        650,368 bytes free conventional memory
        650,304 largest executable program size

      3,407,872 bytes total contiguous extended memory
              0 bytes available contiguous extended memory
      3,316,736 bytes total XMS memory available
      3,316,736 largest XMS memory block size

       PTS-DOS resident in High Memory Area
```

In this case you have 650368 bytes of free memory available for applications.

Extended command set and improved syntax of the configuration file allow a user to fine-tune the System according to own needs. For example, if several system configurations are required, then the following configuration file should be written:

```
[Common]                 ; The common section is always executed
 Device = himem386.Sys   ; Loading XMS driver

[C1@Maximum Memory]      ; Section C1 - maximum free memory configuration
 ComArea    = Standard   ; Usage of memory area starting from 50:00
 Windows    = No         ; Enabling usage of non-standard system data and
                         ; code allocation
 Force      = $DosStd    ; Linking content of the $DosStd section to
                         ; current one

[C2@Compatibility Mode]  ; Section C2 - compatibility mode configuration
 ComArea    = High       ; Disable usage of memory area from 50:00 to 70:00
 Windows    = Yes        ; Standard system data and code allocation
                         ; strategy
 Force      = $DosStd    ; Linking content of the $DosStd section to
                         ; current one

[$DosStd]                ; Section $DosStd - internal, used for linking to
                         ; other sections
 Files      = 40
 Buffers    = 20
 DosDir     =  C:\PTSDOS
 Shell      =  C:\PTSDOS\command.com C:\PTSDOS /p C:\AUTOPTS.BAT
DOS        =  High Umb
 Clock      =  Real
 IDTChk     =  Off
 Abort      =  On
 ShiftState =  Off
 SoftReboot =  On
 LastDrive  =  k
 Country    =  RU,866, C:\PTSDOS\COUNTRY.SYS
 Keyboard   =  RU LoadOnCall ActBorder 56 DefBorder 0 HotKey 02 54
 Display    =  RU Text Preload High
```

EXAMPLES OF CONFIGURATION FILE (continued)

```
[BootMgr]                ; Special section BootMgr. Description and system
                         ; of applicable commands may be found in section
                         ; "CONFIGURATION FILE PROCESSING"

 AutoCheck = Yes
 Color     = White on Black
 Storage   = File
 FileName  = AutoBoot.Sys

[C3@OldDOS]
 SysBoot   = C:\oldboot.sys  ; Indicates the boot sectors address of the
                             ; system installed before PTS-DOS on this
                             ; computer

;<< EOF >>
```

Upon first System start-up (and subsequent start-ups, if the [Esc] key is pressed during the first loading) the following prompt shall be displayed:

```
 Select section number to be active
--------------------------------------
[F]1. C1           Maximum Memory
[F]2. C2           Compatibility Mode
[F]3. C3           ÄldDOS
--------------------------------------
```

By pressing a corresponding numeric or function key, the System configuration shall be changed accordingly.

If a numeric key is pressed in combination with the [Alt] key, the System will step-by-step process the commands of a corresponding section. Therefore, pressing [Alt-1] in the previous example, will cause the following System prompt:

```
        (Y/N)? >Device = himem386.Sys
```

Pressing <Y> will install the himem386.sys driver, <N> - will cancel the operation. TRACE command may be used for toggling the trace mode in certain lines of configuration file.

If HMA is available, the System partially allocates in it depending on the mode which is set up by command as WINDOWS or BUFFERS. Command SYSGEN may be used for direct specifying the allocation area of a system part (code, operative data, I/O buffers, file tables etc.).

SYSTEM CONFIGURATION COMMANDS:

- ABORT
- BREAK (sys)
- BUFFERS
- CLOCK
- COMAREA
- COUNTRY
- CPU
- DEVICE
- DEVICEHIGH
- DISKETTE
- DISPLAY
- DOS
- DOSDIR
- DRIVPARM
- DRIVATTR
- ECHO (sys)
- ERREXE
- EXECA20OFF
- FASTOPEN
- FCBS
- FILES
- FORCE
- HMAREA
- IDTCHK
- IGNORE
- INSTALL
- INSTALLHIGH
- LASTDRIVE
- LOCKS
- LOADFIX (sys)
- KEYBOARD
- MEMORY
- QUIET
- SETVER
- SAVENAME
- SCREEN
- SHELL
- SHELLHIGH
- SHIFTSTATE
- SHUTDOWN
- SOFTREBOOT
- STUBA20
- SWITCHAR (sys)
- SYSBOOT
- SYSLOAD
- SYSGEN
- SYSTOUMB
- TRACE
- VGASCREEN
- VIDEOMODE
- VIRTDMA
- VXD
- WINDOWS
- ZOMBIEDRV

The commands COMMENT, MULTITRACK, REM, STACKS, SWITCHES together with any their parameters are ignored by the System and have been left for the compatibility with other DOS versions.

ABORT

Purpose: Sets the emergency program exit option activated by the [SysReq] key.

Syntax: ABORT = <ON|OFF>

By default: ABORT=ON

Note:

In certain cases when a program halts, it is sufficient to take measures on exiting the current process in order to resume normal system operation. If the ABORT is ON, you may do the above by pressing [SysReq]. Nevertheless, such exiting is not always correct as the program could have allocated some part of extended and/or expanded memory, and DOS can not release it automatically, or temporary files may have been created which can be deleted just on normal completion of the program. Besides that, the aborted program could have made changes to the interrupt vectors table, making further operation impossible.

COMAREA

Purpose: Sets the DOS communication area location.

Syntax: COMAREA = <STANDARD|HIGH>

By default: COMAREA = HIGH

Note:

COMAREA = STANDARD instructs the system to use the memory area between 0050:0000 and 0070:000 for its internal purposes. As the PTS-DOS uses this memory in not completely the same way as with other DOS versions, some programs will not operate if COMAREA=STANDARD. When WINDOWS=ON, as it is set by default, the COMAREA=STANDARD is not executed and it is only permissible in case if

```
WINDOWS=OFF
COMAREA=STANDARD
```

Allocating communication area higher than standard requires 530 bytes additionaly, which are always in the lower memory area.

DEVICEHIGH

As the DEVICE command above, but the System will try to load the specified driver to the UMB. If UMB is absent at the time of the command execution, or there is no enough memory of this type, the System will assume the command is DEVICE.

DOSDIR

Purpose: Sets an additional search path for auxillary system files.

Syntax: DOSDIR = [drive:]path
            where, path is a directory name

By default: DOSDIR =\

Note:

Some configuration file commands require corresponding special files, like display.cpi, keyboard.sys, country.sys. By default, the system will first search the root directory and in case of failure, will continue searching in the directory specified by the DOSDIR command.

ERREXE

Purpose: Sets the level of checking code to be loaded for its validity.

Syntax: ERREXE = <ON|OFF>

By default: ERREXE = OFF

Note:

When ERREXE = ON, the system will not load modules if:

1. The size of the part to be loaded exceeds the total filesize.
2. In case of fixup overflow.

Some programs containing these errors otherwise function normally, therefore an user is left free to decide the systen reaction to such occurences.

FILES

Purpose: Sets the number of file open operations accessible via system calls.

```
Syntax:   FILES =<xx>[,yy][HIGH|LOW][STANDARD|CHANGEABLE|FIXED]
```

Parameters:

```
xx    -  the maximum number of concurrently open files,
         the default value is xx=20.
LOW   -  use the standard scheme for allocating files.  In this case the
         yy parameter is not used while xx ranges from 8 to 255.
HIGH  -  instructs the system to use the extended memory for allocating
         file tables. yy sets the file tables number of a cache, allocated
         in the lower memory, while xx ranges from 8 to 60000.
STANDARD or CHANGEABLE - specifies whether the code servicing dynamic
         changes of the xx value when running an application program, is
         to be loaded.
FIXED -  do not load the above described code.
```

By default: FILES = 20,8 LOW STANDARD

Note:

When allocating files in the extended memory, the value of FILES may exceed 255 with additional system call required for a program to open more than 255 files. These calls in fact set the value that is added to the argument specified with a system call. This feature is extremely useful in multitask enviroments that require a large number of files concurrently opened by different tasks. Still, very large values of the parameter slow down the system performance when opening a file, since the system has to look up all the file tables in order to define the reccurence of such operation.

IGNORE

Purpose: Inhibits system processing of specified sections.

Syntax: IGNORE = section[,section[,...]]

Note:

Not used, as a rule, and serves only to complement the FORCE command. As the FORCE and IGNORE commands involve only single-pass parsing, only the names of subsequent sections should be used as their parameters.

LOCKS

Purpose: Sets the number of system file locks, accessible in system calls.

Syntax: LOCKS = xx
            Where, xx is the number of locks ranging from 4 to 92, or 0

By default: LOCKS = 4

Note:

The lock principle is used primarily with networked software. In absence of such, the best memory saving choice is LOCKS = 0.

QUIET

Purpose: Inhibits output of messages by drivers and TSRs to STDOUT when using commands DEVICExxx = ... and INTSALLxxx = ... during system start-up.

```
Syntax:         QUIET = <ON|OFF>
By default:     QUIET = OFF
```

Note:

The command redirects all STDOUT output to the NUL device, preventing initial loading messages made by device drivers and resident programs during system start-up. Programs, as a rule, use the STDOUT for displaying messages during normal operation and the STDERR in case of errors. But, there are programs using STDOUT for error messages. Such messages won't be seen, if QUIET= ON, this produces an impression that the system "hangs" when a program or driver waits for data input from the console. The QUIET command, therefore is recommended for use only after the system configuration has been throroughly checked up.

QUIET=YES together with LOGO=OFF, as a rule fully supresses all the screen output.

SHELL

```
Purpose:    Sets an alternative command processor name.

Syntax:     SHELL = [drive:][path]filename [parms]
            Where, filename is a command processor name,
                   parms is the command line to be passed to the program.

By default: SHELL = command.com  /P
```

Note:

Refer to the description of the Command.com and the initial value of the COMSPEC environment variable. In case of failing to find the specified shell or an error while running it, the system will prompt:

"Shell ? _"

In this case, the name of the program together with its parameters should be specified, e.g. "\cmd\command.com /p/e:512", or alternatively, the [Ctrl-Break] may be used for cancelling the start-up.

SOFTREBOOT

```
Purpose:    Quick system restart, skipping the POST BIOS.

Syntax:     SOFTREBOOT=<ON|OFF|x,y>
            Where, x is the trigger keys state,
                   y is the key scan-code.

By default: SOFTREBOOT OFF
```

Note:

In some cases, it is convenient to quickly restart the system, e.g. when experimenting with the configuration file parameters. That's when SOFTBOOT=ON should be used. The default quick restart key combination is [Ctrl-Alt-Ins]. Another combination can be specified using the x and y parameters.

Warning! When trying to restart with the processor running in the protected mode, a sound signal warns that the action is impossible. As the SOFTREBOOT=ON involves closing all the files, the SHUTDOWN=OFF cancels it.

SYSLOAD

Purpose: Loads a system from another hard disk partition.

Syntax: SYSLOAD = <partNO|?>

```
    partNO - (1 - 4) - number of the partition with boot sector
             of the system to be loaded,
    '?'    - question mark - the system will display all the partition
             numbers and information on the operating systems contained
             there.
```

Note:

An alternative UNIX-type operating system with a file structure different from that of DOS may be loaded by specifying the corresponding hard disk partition. If the partition numbers are known, they may be directly specified in the command. Specifying the question mark will make the system to display information on the partitions and wait for user input of the desired partition number. Incorrect section choise (incorrect code or a non-bootable partition) may "hang" the computer.

VGASCREEN

Purpose: Controls the mode of a VGA-compatible display

Syntax: VGASCREEN = <<ENABLE|DISABLE>|<ON|OFF>>

Parameters:

ON - activate the screen

OFF - disactivate the screen

ENABLE - enable writing to the video RAM

DISABLE - disable writing to the video RAM

Note:

It worth to remember, that when OFF, the ENABLE and DISABLE commands are ignored.

The command allows to disactivate the screen or to organize video ROM with VGA-compatible displays. Fully corresponds to the analogous BIOS calls.

WINDOWS

Purpose: Sets the system compatibility level when working with MS-Windows.

Syntax: WINDOWS <ON|OFF|STANDARD|286|ENHANCED|386> [/V:version]
Where 'version' sets the Windows version number, e.g., 3.0 or 3.1

By default: WINDOWS ON /V:3.0

Note:

MS-WIndows requires additional DOS code and system data located in the lower areas of the memory, depending on its version and mode (real/standard/enhanced). To achieve the best results, both mode and version number should be specified. For example:

```
 WINDOWS STANDARD           Assumes that MS-WINDOWS 3.0 is to be used in
 WINDOWS STANDARD /V:3.0    standard mode.
```

STANDARD and 286 are synonyms, just as the ENHANCED and 386.

```
 WINDOWS 286 /V:3.1     MS-Windows 3.1 in standard mode
 WINDOWS 386 /V:3.0     MS-Windows 3.0 in enhanced mode
```

WINDOWS=ON assumes the version is 3.0 with the mode set maximum according to the type of processor used. For the 286-class machines, it is STANDARD, while for 386 and up - ENHANCED. It is not recommended to use the SYSGEN command together with the WINDOWS command, as the system generation may be incompatible with that required by MS-Windows. WINA20.386 (a driver supplied with the system) is required for running MS-Windows 3.0 in the enhanced mode (see the VXD command description).

BREAK (sys)

Purpose: Sets a check-up for pressing [Ctrl-Break].

Syntax: BREAK = <ON|OFF>

By default: BREAK=OFF

Parameters:

ON - sets BREAK = ON. The system checks if the [Ctrl-Break] keys' being pressed before executing any DOS command and breaks the execution, if those keys are pressed.

OFF - sets BREAK = OFF. In this case the system controls [Ctrl-Break] keys' pressing only during:
- standard input operations,
- standard output operations,
- standard printing operations,
- operations with standard auxiliary devices (adapters).

Running the BREAK command without parameters leads to the BREAK current condition being displayed.

COUNTRY

Purpose: Allows PTS-DOS to use different formats of time, date, national currency symbols, thousands and decimal separators, conventions on conversion of lower case letters to upper case depending on the country specified.

```
Syntax:     COUNTRY = <XXX>[,[YYY][,[drive:][path]filename]]
            Where, XXX      - country code or abbreviation
                   YYY      - new code page
                   filename - file containing the required information.
```

By default:

If there is no COUNTRY command in the configuration file, the system uses the code page for the United States.

If no filename is specified as an argument, DOS assumes it to be COUNTRY.SYS and looks for it either in the root directory or the directory specified by DOSDIR.

Note:

This configuration command identifies to PTS-DOS which country character set is to be used. The following table lists all countries supported by PTS-DOS. The table also lists the code pages that may be used with a specific country code.

Country	Abbreviation	Code	Code pages
USA	US,USA	1	437, 850
Canada	CF	2	863, 850
Latin America	LA	3	437, 850
Russia	RU	7	866
Netherlands	NL	31	437, 850
Belgium	BE	32	437, 850
France	FR	33	437, 850
Spain	SP	34	437, 850
Italy	IT	39	437, 850
Switzerland	SG,SF	41	437, 850
United Kingdom	UK	44	437, 850
Denmark	DK	45	865, 850
Sweden	SV	46	437, 850
Norway	NO	47	865, 850
Germany	GR	49	437, 850
English (International)	-	61	437, 850
Portugal	PO	351	860, 850
Finland	SU	358	437, 850
Arabic	-	785	437
Israel	-	972	437
Japan	-	081	932,850,437
Korea	-	082	934,850,437
China	-	86	936,850,437
Taiwan	-	88	938,850,437

DISKETTE

Purpose: Sets permissible diskette types.

Syntax: DISKETTE = <EXTSIGN|DOS3>

By default: DISKETTE = DOS3

Note:

System support of a floppy disk implies that when changing a diskette, all the internal data structures associated with the specified device should be renewed, and a check on all the information possibly not recorded onto a previous diskette (for example, files which have been modified were not closed) should be carried out. Diskette format programs designed for older DOS versions didn't create a unique record in the boot sector, that helps detect a change of diskettes. Without such records the system compares diskettes by their formats. It is of course a non-reliable comparison and there is always a chance of data loss when mishandling diskettes. The DISKETTE = EXTSIGN instructs the system to work only with diskettes having an unique label.

Warning! Many format programms don't support the extended diskette boot sector format. Using DISKETTE = EXTSIGN is recommended only with disk drives without change line.

DRIVPARM

Purpose: Amends or adds parameters set by the System for a physical disk device at system boot.

Syntax: `</D:dd>[/C][/F:ff][/H:hh][/N][/S:ss][/T:tt]`

Parameters:

/D:dd - dd is a physical device number, ranging from 0 to 255.

/C - indicates that the drive with change line is used. With such a drive which detects whether a diskette has been removed, the System assumes that a diskette has not been changed and doesn't check for this state, if there is no signal from hardware. See the documentation for your disk drive to find out whether it has a change line.

/F:ff - ff specifies the block device type. By default, ff=2, possible values given below:
0 - 160/180/320/360K
1 - 1,2M
2 - 720K (3.5" disk)
5 - Hard disk
6 - Tape drive
7 - 1.44M (3.5" disk)
9 - 2.88M (3.5" disk)

/H:hh - specifies the maximum number of heads, the default is: hh=2.

/N - specifies that a disk is nonremovable.

/S:ss - sets ss as the number of sectors per track for a given device.

/T:tt - sets the number of cylinders on a disk.

EXECA20OFF

Purpose: Sets the lines state "A20" at the start of executing a freshly loaded program.

Syntax: EXECA20OFF = <YES|NO>

By default: EXECA20OFF = YES

Note:

Everything described below is applicable only when the system is using the HMA (See the DOS=HIGH command).

By default, the system tries to make the HMA area invisible before transferring the control to the loaded program's first instruction, imitating the address space of 88/86 processor PCs. It is worth mentioning that some programs designed with old versions of the Microsoft LINK, work incorrectly if loaded lower than the first 64K of memory. In such cases the error is unnoticed if the higher 64K of the first megabyte are located below (XT type). The system automatically fixes the code of such programs (See the LOADFIX command), but not always, like when, for instance, the "bad" program is packed with an utility as PKLITE etc. Therefore, if there are no old programs involved or if the "A20" line is being automatically controlled by the computer, it is better to set EXECA20OFF = NO, which will slightly accelerate loading of smaller programs.

FORCE

Purpose: Makes the specified section(s) available for system processing.

Syntax: FORCE = section[,section[,...]]

Note:

A section may be defined for system processing either by user selection when running the boot manager or with the help of this command. For example,

```
[ PtsDos ]
Force = $Drv
 ...
[ $Drv ]
device = himem386.Sys
```

So, when processing the previously defined PTSDOS section, the system will find the FORCE = $Drv command and process the $Drv section resulting in the himem386.sys driver being installed.

As the FORCE and IGNORE commands involve only single-pass parsing, only the names of subsequent sections should be used as their parameters.

INSTALL

Purpose: Allows executing some programs at the stage of configuration file processing.

Syntax: INSTALL = [/e] [drive:] [path]filename [parms],

Parameters:
parms - command line that will be transferred to a program
/e - creates a program environment allocated in the higher memory.

Note:

When loading a program, the system creates an unique environment available to this program and located ,as a rule, before its beginning. Running resident programs which free the memory before exiting, leads to memory fragmentation. Using the /e key provides a programm enviroment located after the program code. Therefore, no memory fragmentation occurs when freeing the environment block.

LOADFIX (sys)

Purpose: Corrects loading of programs designed with old versions of the LINK.EXE program.

Syntax: LOADFIX = <ON|OFF>

By default: LOADFIX = OFF

Note:

Since the old LINK.EXE versions all contained a bug, the following message is displayed when modules created with such a linker are loaded into quite low areas of memory:

Packed file is corrupt

The problem mostly occurs when DOS is loaded into the high memory and may be solved either by the LOADFIX external command or in this case, by LOADFIX = ON. The latter automatically detects incorrectly working programs. The memory located below the loaded program is free and may be used on enquiry; the system loads additional code of approximately 500 bytes.

If the message "Packed file is corrupt" is displayed even when LOADFIX=ON, the LOADFIX external DOS command may be used for solving the problem.

SETVER

Purpose: Allows to assign to different executable programs different numbers of DOS version.

Syntax: SETVER = {Yes|No|On|Off}[,<filename>]

where: <filename> - file name (with path) containing information about DOS versions reported to programs (by default SETVER.SYS).

By default: SETVER=Off

Example:

```
SETVER = On,c:\ptsdos\setver.sys
```

SETVER.SYS file structure (text form):

```
<filename1> x.xx
<filename2> y.yy
................
<filenameN> z.zz
```

According to this file the number of version x.xx will be reported to the <filename1> program, the number of version y.yy will be reported to the <filename2> program, z.zz will be reported to the <filenameN> program. The PTS-DOS distribution diskette contains an example of the file SETVER.SYS.

SHELLHIGH

```
Purpose: Sets an alternative command processor name with an attempt of
         loading in into the UMB.

Syntax:  SHELLHIGH = [drive:][path]filename [parms]
         Where, filename - command processor name
                  parms  - command line to be passed to the program.


Note:   As the SHELL command but tries to install the shell into
        the UMB - upper memory blocks.
```

STUBA20

Purpose: Defines a special driver that increases performance of the system located in high memory.

Syntax: STUBA20=<ON|OFF>

By default: STUBA20=OFF

Note:
A special XMS support driver is required for allocating the system in high memory. Any interrupt intercepted by the system leads to a HMA access request driver call. When STUBA20=ON, the system controls the driver calls that enable or disable HMA, so that no additional calls are made. The result is a slightly improved system performance.

Warning! Not recommended for use when running programs that control the high memory, bypassing the XMS driver. STUBA20=ON is incompatible with MS-Windows.

SYSGEN

Purpose: Sets the PTS-DOS component allocation strategy.

```
Syntax:  SYSGEN <[/Cx][/Sx][/Lx][/Dx][/Vx][/Rx][/Fx][/Tx]>,
         Where, x is:  'L' - forces to allocate components always in
                             conventional memory,
                       'H' - enables using the HMA.
```

Note:

Please, refer to BUFFERS /L or /H - on the strategy of allocating block device I/O buffers, and to COMAREA - on specifying the communication area. The following abbreviations are used with the command:

```
   C - system code
   S - system data
   L - locks data
   D - disk parameters data
   V - data on the additional devices
   R - current directories structure
   T - floppy disk parameters table
   F - file tables
```

Setting, for example, /CH enables the system code to be allocated in the high memory when DOS=HIGH; /FL forces the system to allocate the system file table in the lower memory.

The default strategy depends on the WINDOWS command in the first place. If WINDOWS=OFF, then all data and codes are enabled for loading into the high memory, excluding additional data on the installed block device drivers.

If WINDOWS=386/V:3.0, then the file structure should be in the lower memory, while WINDOWS=386/V:3.1 enables placing the structure in the high memory.

Warning! The command is for the experienced user only. System faults may occur with some application programs if the command parameters are incorrect (incompatible).

VIDEOMODE

Purpose: Sets the current video mode during system start-up.

```
Syntax:   VIDEOMODE =[SOFT]<mm|MONO|CO80|BW80|CO40|BW40>
          Where, mm is the decimal number of the current mode
```

Note:

The SOFT prefix adds 128 to the video mode number set by the user in order to save the contents of the video buffer. This operation is only applicable to EGA video adapters or compatibles.

ZOMBIEDRV

Purpose: Allows bufferization for file deletion on the specified disks.

Syntax:

```
ZOMBIEDRV = drv[:][,drv[:][, ...]]
```

Note:

The main purpose of this command is to create a queue of files to be deleted. Due to this queue, it is now possible to restore recently deleted files with no data loss. The procedure is the following: when a file is deleted, the system moves it to the \ZOMBIE directory, created before-hand by the MKZOMBIE external DOS command. Directory size is specified on its creating and it defines the queue size. When the directory is full, the system will delete an old file and will write a new one in its place. Files from this directory are also deleted when there is no more space on a disk for new files. As the path is not saved when moving a file, the filenames may be recurring, as could be seen with DIR command. Only the first file is processed. Error messages may occur when running disk data check-up utilities because of some files with the same names.

BUFFERS

Purpose: Sets the number of buffers allocated in RAM to buffer disk operations and I/O operations with other types of block devices.

Syntax: BUFFERS = <XX> [,YY] [/X] [/L|/H]

```
XX - number of buffers, from 8 to 99;
YY - not used by PTS-DOS, left for compatibility with other
     DOS versions;
/X - not used by PTS-DOS, left for compatibility with other
     DOS versions;
/L - instructs the System to allocate buffers only in the
     lower (up to 640+) memory;
/H - enables usage of HMA for allocating buffers.
```

By default: BUFFERS = 8 /H

Note:

A disk buffer is a block of memory which is used for buffering I/O operations with block-type devices, that can transfer data only in blocks of fixed size.

For applications such as word processors, a number between 10 and 20 is quite sufficient for the system. If you are going to use the additional caching program, then the default setting will provide normal performance. Increasing the number of buffers by one, takes up additional 530 bytes of memory if the DOS is not loaded into the high memory, but if it is, you should look through the SYSGEN command description for explanation of the strategy of allocating DOS in memory.

You should try various settings of BUFFERS to find out an optimal I/O speed/available memory ratio.

CPU

Purpose: Slightly improves the performance of a PC.

Syntax: CPU = <FAST|STANDARD>

By default: CPU = STANDARD

Note:

This command increases the memory refresh cycle, resulting in a little higher performance of a PC.

DISPLAY

Purpose: Sets the display font.

```
Syntax:   DISPLAY = <xx|ID> [GRAPHICS|TEXT]
                       [FONTS<yy|STANDARD>]
                       [PAGES zz] [HIGH|LOW]
                       [FILE [drive:][path]filename]
                       [PRELOAD|LOADONCALL]
                       [RESIDENT|NONRESIDENT]
                       [USEHARDWARE|USEBIOS]
          Where, filename - font description file
                 zz       - number of pages supported
                 xx       - page code
                 ID       - country identifier
```

Note: By default, the following optional parameters are assumed:

```
        GRAPHICS =  FONTS STANDARD, PAGES 1, LOW \
                    \ FILE Display.Cpi, LOADONCALL, RESIDENT
```

This command allows to load an alterantive display font, if the videoadaptor supports this feature. As a rule, these are: EGA, VGA (SVGA), MGA, XGA and analogous. If the RESIDENT parameter is present, the system will load a code that will save the specified font during change of the videomode and make it possible to interactively switch to it by pressing Ctrl-Shift-F1...F9 , where F1-F9 define active page (F1 is the BIOS supported font). For more details, refer to the external DISP command description.

PRELOAD, LOADONCALL define the initial state: in the first case a new font is loaded during processing the DISPLAY command from the first page, in the second case switching between fonts is possible either in the interactive mode, or with the help of CHCP and DISP commands. FILE sets the path and filename for the file containing binary font description. By default, the filename is Display.Cpi. HIGH and LOW set the type of memory used for allocation of font images, the former sets the usage of extended memory, while the latter concerns the lower area memory.

PAGES sets the number of various code page combinations, supported by the system. PAGES1, the default value, is sufficient for a twin-language interface. It is recommended to use higher values in combination with HIGH, which will save on system memory.

FONTS sets the number of types for each font. Normally, different matrix fonts are required to support all the available videomodes, e.g. 8x8 ,8x14, 8x16. Their number is defined by the type of the videoadapter. But, if a soft font is to be used in one of the possible modes, FONTS should be specified to save the system memory.

Supported matrices	Adapters	FONTS 1	2	3
8x8, 8x14, 8x 16	VGA	8x16	8x8, 8x16	8x8,8x14,8x16
8x8, 8x14	TGA	8x14	8x8, 8x14	8x8, 8x14
8x8	other	8x8	8x8	8x8

Soft font matrices

GRAPHICS, TEXT instruct the system to support a national font either both in graphic and text modes or only in text mode. When using GRAPHICS,

more system memory is required, as a rule.

When processing the DISPLAY command, the System can only load fonts for a single code page (country). On specifying with PAGES a value larger than 1, loading other code pages becomes possible, but only using the DISP -external DOS command. A two-letter country identifier code may be used instead of the code page. In this case, the system will choose the corresponding page code automatically. See COUNTRY for more information on this feature.

Available code pages:

437	United States of America
850	Multilanguage interface
860	Portugal
863	Canadian-French interface
865	Norway
866	Russia

USEHARDWARE and USEBIOS define the way the videoadapter is programmed. USEHARDWARE instructs the system to program VGA/EGA adapter registers directly. With this mode there will be no display flickering when switching between fonts interactively with keys Ctrl-Shift-Fx. USEBIOS instructs the system to use the BIOS video for display fonts loading. High level of compatibility with existing hardware and software is achieved here.

DRIVATTR

Purpose: Sets additional physical parameters of a disk device.

Syntax: DRIVATTR =</D:dd> <[/T][/C][/S:xx]>

Parameters:

/D:dd - defines the device logical number

/T   - specifies that the device controller can process input-output enquiries that require addressing several cylinders. Applicable only to IDE controllers, as a rule.
WARNING! Not recommended to use when running disk cache programs, e.g. SMARTDRV.EXE and when loading MS-Windows 3.1 in enhanced mode.

/C   - Disables system interference over diskette access control. With this key PTS-DOS will not support non-standard diskette formats.

/S:xx - sets the number of rotations per second for a diskette. At boot, the system sets the following values:
for a 360K drive - 5 r.p.s.
for a 1.2M drive - 6 r.p.s.
for 3,5" drives 720K/1.44M/2.88M - 5 r.p.s.
The parameter is used for calculating GPL - the filling gap between sectors on a diskette track.

FASTOPEN

Purpose: Sets the number of buffers for organizing a fast file search cache.

Syntax: FASTOPEN = xx
Where, xx - ranges from 4 to 99 and defines the number of buffers.

By default: FASTOPEN = 4

With a complex directory structure, the system has to look through a significant number of directories when searching for a file deeply located and if path to this file is not found in any of the buffers, the number of which is set by the FASTOPEN. Each buffer takes up about 80 bytes of system memory. Maximum values for the number of buffers should be used when working with file-extensive applications such as DBMS.

HMAREA

Purpose: Sets an initial HMA segment address.

Syntax: HMAREA = <xxxx|STANDARD>
                Where, xxxx - segment address

By default: HMAREA = STANDARD

Note:

The STANDARD constant is equivalent to 0FFFFh. By default, the HMA is located between FFFF:0010 and FFFF:FFFF and an XMS driver is required for its support. But, if 64K of memory are available between A000 and FFFF ,it may be used as HMA with no additional driver.

It is not recommended to use this command if the HMA size is different than 64K, since some programs utilize HMA for their own purposes and may not correctly work with HMAREA.

INSTALLHIGH

Analogous to the INSTALL command except that it uses the UMB for allocating program code.

KEYBOARD

Purpose: Sets an alternative keyboard layout.

```
Syntax:     KEYBOARD=[XT|AT|PC]<STANDARD|ID>
                     [SIMPLE]
                     [HOTKEY ss cc]
                     [TEMPHOTKEY cccc]
                     [PRELOAD|LOADONCALL]
                     [ACTBORDER xx]
                     [DEFBORDER yy]
                     [USEBORDER]
                     [MAPS nn]
                     [,[codepage [,[drive:][path]filename]]

            Where, ss   - shift state
                   ßß   - scan code
                   ßßcc - scan code
```

By default: KEYBOARD = STANDARD

Note:

When defining some "extended" keys it is necessary to set upper byte of the cccc as 0E1h or 0E0h. For instance, the <Ctrl> scan code should be defined as 57373 or 0E01Dh.

To change the keyboard layout from the one supported by BIOS, it is necessary to include KEYBOARD = ID, where ID is a two letter country code (See the COUNTRY command). The new layout may be substituted for another in an interactive mode by using the KEYB external DOS command. Specifying STANDARD is equivalent to stating no KEYBOARD command at all.

The XT, AT and PC prefixes specify the type of keyboard interface and are recommended only in case of faults arising from using the KEYBOARD command.

SIMPLE means absence of "dead keys". Dead keys don't lead to display of some character after being pressed separately. If a letter key follows the dead one, then the specified letter is displayed with an accent or "umlaut".

You can switch from one layout to another in interactive mode with the help of the following keys:

<Ctl-Alt-F1> - set the initial layout
<Ctl-Alt-F2> - set the loaded layout,

or with the DOS KEYB command. Also, a hot key can be specified by stating HOTKEY ss cc, where ss is the shift state and cc - the scan code. In this case you may shift between states by pressing the specified key. It is also possible to define a TEMPHOTKEY hot key used for shifting the layout only while it is held pressed.

Display border may indicate the current status with its color, if the USEBORDER parameter is specified. ACTBORDER xx defines the active state border colour (white by default), DEFBORDER yy defines the initial state border colour (black by default). Parameter values are treated depending on the type of display: for CGA, it is one of the 16 colors, for EGA and the like, one of the available palette colors (one of 64 or more colors).

MAPS nn defines the maximum number of layouts loaded with KEYB command. nn may range from 1 to 9, default value is 1.

NOTICE: Border color indication does not apply to MDA and Hercules displays, both not having the option.

The system automatically installs a new keyboard layout when KEYBOARD command is processed. PRELOAD is used for making an new layout active after computer loading. LOADONCALL points out that a new layout will be installed

KEYBOARD (continued)

by pressing hotkeys. Codepage means exactly what it says.

Filename (By default, KEYBOARD.SYS) specifies name of the file containing the keyboard layout data.

SAVENAME

```
Purpose:     Controls the option of saving the first character of a deleted
             file or directory name.

Syntax:      SAVENAME = <ON|OFF>
By default:  SAVENAME = OFF
```

Note:

When SAVENAME=ON, the system saves the first character of a deleted file or directory name. The saved first character of a filename can be used by the UNERASE command.

SHIFTSTATE

```
Purpose:    Sets the initial state of the trigger keys.

Syntax:     SHIFTSTATE OFF|STANDARD|
                           |<[SCROLLLOCK][NUMLOCK][CAPSLOSK]>
                           |<[KEYRATE=rval] [KEYDELAY=dval]>

            Where,
            rval=1..32 chars./sec., or: MAX and MIN
            dval=1..4             , or: MAX and MIN

By default: SHIFTSTATE STANDARD

Note:      STANDARD - no action,
           OFF      - resets all the keys,
           SCROLLLOCK, NUMLOCK, CAPSLOCK - sets the active state for
                      the corresponding keys.
```

KEYRATE and KEYDELAY specify the auto-repeat rate of a pressed key in characters per second and the delay before the auto-repeat, respectively. Much in parallel with the MODE CON command.

SWITCHAR (sys)

Purpose: Sets the switch character.

Syntax: SWITCHAR = c

By default: SWITCHAR = /

Note:

To switch to a UNIX-like command writing style, specify:

```
SWITCHAR = -
```

All the parameters as /A/B/C should be then written as -A-B-C. '/' instead of '\' should be used in the file path. The first line of the configuration file is the best place for this command, because any other place might lead to ambiguity with parameters, since the system does several passes when processing the file.

SYSTOUMB

```
Purpose:    Enables usage of UMB for internal system data and code.

Syntax:     SYSTOUMB = <ON|OFF>
By default: SYSTOUMB = OFF
```

Note:

If SYSTOUMB = ON, the system will attempt to allocate first the HMA memory, then the UMB and in case of more need, the conventional memory.

With some UMB servers running on 386+ computers in protected mode and with SYSTOUMB = ON, restarting the computer by pressing [Ctrl-Alt-Del] might lead to system halt, as the server first switches off the UMB, then returns to real mode operation and jumps to a code located previously in the UMB, but not located there any more.

VIRTDMA

```
Purpose:    Defines the server ability to support the virtual DMA service.

Syntax:     VIRTDMA = <ON|OFF>
By default: VIRTDMA = OFF
```

Note:

By default, the system intercepts the interrupt 13h (disk I/O) and executes I/O to the high memory - A:000:000 and above, by using an internal driver located in the lower memory. If the UMB supports the virtual DMA service, a condition almost always fulfilled, then specifying VIRTDMA=ON makes I/O to higher memory run faster, e.g. when loading a program using the INSTALLHIGH command. This is also recommended for usage with a UMB-server that operates on hardware level by reprogramming the chipset, for example, LASTBYTE.

CLOCK

Purpose: Specifies the type of system clock, installed in the computer.

Syntax: CLOCK = <REAL|STANDARD>

By default: CLOCK = STANDARD

Note:

In the default setting, DOS derives the current time and date using a timer available with any personal computer. The initial state is then set either by the real-time clock or equal to 00:00.00 1-1-1980. Setting CLOCK = REAL makes the system to load a code, that directly works with the real-time clock on time and date enquiries. In this mode less system memory is used and the date range extends back to the year 1553. It is recommended to set CLOCK = REAL in all cases when no compatibility with old DOS versions is required with the symbol device CLOCK$.

DEVICE

Purpose: Installs specified driver into the system list.

Syntax:

```
DEVICE = [drive:][path] filename [arguments]
Where, arguments are all parameters of a driver.
```

Note:

If you purchase a new device, like a mouse or a scanner, you will receive software device driver with that device which services it. To install this driver, the DEVICE command should be used where the driver name and location should be specified.

DOS

Purpose: Makes DOS to load itself into the HMA on PCs with 286+ processors, also instructs DOS to use or not to use the upper memory area (UMB).

Syntax: `DOS = < [HIGH|LOW] [UMB|NOUMB] >`

By default: `DOS = LOW,NOUMB`

Note:

Use this command to increase the size of the free lower memory.

Options HIGH and LOW define whether DOS will use the 64K HMA or not. This option requires installation of a XMS server. The PTS-DOS package includes two XMS support drivers - Himem286.Sys and Himem386.Sys for 286 and 386+ processors, respectively.

Options UMB and NOUMB define whether DOS will use the UMB blocks or not. When DOS=UMB, a TSR or a driver may be installed into the UMB using the DEVICEHIGH=, INSTALLHIGH= commands or with the LOADHIGH DOS command. The option requires installed UMB server, e.g. EMM386, 386MAX, QEMM386.

Options may be used combined in a single line. Line:

```
DOS=HIGH,UMB
```

is equivalent to the lines:

```
DOS=HIGH
DOS=UMB
```

ECHO (sys)

Purpose: Displays a message on the screen during the first pass of a configuration file.

Syntax: ECHO = "message"

Where, message is any text you like, embraced with the quote marks <">.

Note:

As the command is executed during first pass of the configuration file, there is no synchronization between the ECHO command and other commands, like DEVICE=.

FCBS

Purpose: Sets the emulation level of FCB system calls, usual for the old system versions.

Syntax:
```
FCBS = <xx[,yy]|ON|OFF>
```

By default: FCBS = ON

Note:

The yy parameter has been left for the sake of compatibility with previous DOS versions. xx values other than 0 correspond to the ON state, xx=0 - to the OFF state.

FCBS=OFF saves about 1.5K of system memory. Only the Find First/Find Next/Create system calls are then available trough FCB. Other calls that are supported by the standard system interface, are possible only when FCBS = ON.

There are programs that employ undocumented FCB fields. In these cases, as a rule, messages of errors in DOS file system occur. To cure the problem, the CHKDSK external DOS command or the system- independent programs as Symantec's Norton Disk Doctor, should be used.

IDTCHK

Purpose: Allows to detect boot viruses in certain cases.

Syntax: IDTCHK = <ON|OFF>

By default: IDTCHK = OFF

Note:

When IDTCHK = ON, the system tests the computer memory for key interrupt code receivers. In case of suspicious interrupt pointers a warning message is displayed and the user has a choice of either continuing the boot or cancelling it.

LASTDRIVE

```
Purpose:    Sets the maximum number of accessible logical drives.

Syntax:     LASTDRIVE = ll
            Where, ll ranges from 'A' to 'Z'

By default: LASTDRIVE = max('E', last installed drive)
```

Note:

The ll value defines the last valid logical drive that the system may access. The maximum value is derived when booting. The command is useful with a network environment because, new drives may appear in the process of work. Additional drives are also required for the SUBST external DOS command, allowing different logical drives to be used for access to parts of the physical drive.

Notice that the system allocates memory for each logical drive, so only the necessary maximum value should be speciifed.

MEMORY

Purpose: Sets system memory allocation strategy.

Syntax: MEMORY=<[BESTFIT|FIRSTFIT|LASTFIT] [NOUMB|UMB]>

By default: MEMORY=FIRSTFIT NOUMB

Note:
This command is executed before the pass processing some commands like INSTALL. The memory allocation strategy can be defined trough system calls.

```
FIRSTFIT - use the first acceptable memory block
BESTFIT  - choose the block closest in size to that inquired
LASTFIT  - use the last of the acceptable blocks
UMB      - use UMB (only if DOS = UMB)
NOUMB    - search for free memory only in the lower memory
```

Hence, the commands

```
    MEMORY=FIRSTFIT UMB
    INSTALL=filename
```

are equivalent to

```
    INSTALLHIGH=filename
```

although, in the first case it is impossible to perform the simple INSTALL command in view of the global action of the MEMORY command.

SCREEN

Purpose: Sets the type of display interface.

Syntax: SCREEN=<[STANDARD|ANSI][FAST|SLOW][DIRECT]>

By default: SCREEN=STANDARD FAST

Note:

SCREEN=ANSI should be specified for supporting ANSI screen output standard. In this case, only the console write operations support ESC-sequences with no keyboard input support of the latter. ANSI.SYS (supplied with the system) should be used for the full ANSI service. When writing to the screen, the system either uses BIOS or directly writes into the video-buffer. The system is forced to write into the video-buffer when DIRECT is stated and the current video mode number is 3 or 7. The approach is recommended for the slower machines of the PC XT type. Besides, there is a possibility of deliberately processing console write system calls with the mode set by FAST or SLOW. In the first case, output rate is high while in the second, it is lower, but with no special code taking up additional memory. The difference in speed is negligible with 386 and higher machines while quite detectable with XT-type machines.

SHUTDOWN

Purpose: Sets the system response to [Ctrl-Alt-Del].

```
Syntax:     SHUTDOWN = <ON|OFF>
By default: SHUTDOWN = ON
```

Note:

When ON, the system will close all the files open for writing operations and save all the buffers to the disk in response to [Ctrl-Alt-Del]. No action is undertaken in the opposite case.

As the SOFTREBOOT=ON involves closing all the files, the SHUTDOWN=OFF cancels it.

SYSBOOT

```
Purpose:   Loads another DOS-compatible system.

Syntax:    SYSBOOT = <bootimagefile> [bootdrive[:]]

Parameters:

bootimagefile - file with the specified system boot sector image,
bootdrive     - disk to be regarded as the initial for the
                specified system.
```

Note:

A file containing the boot sector image of the system used prior to the PTS-DOS may me created by running the SYS external DOS command with the /U key.

If a file containing the boot sector image has incorrect data or does not correspond to the initial start-up disk, the system may "hang".

TRACE

Purpose: Sets/resets trace mode when processing configuration file.

Syntax: TRACE = <OFF>|<ON [DEFAULT=<YES|NO>][TIMEOUT=secs] >

Note:

Global trace mode may be activated by selecting a main section 'Alt-1..0' (See Configuration file processing). All sections to be processed shall be traced then. TRACE = ON action is limited to the beginning of the following section.

DEFAULT defines the default reply on the system request on the expiry of the TIMEOUT time (in seconds) or on pressing the [Enter] key. With DEFAULT specified, the request form shall be:

(Y/n/E) > or **(y/N/E) >**

depending on the parameter value: in the first case (as well as in the case when TIMEOUT is specified with no DEFAULT parameter), the DEFAULT=YES, while in the second case, DEFAULT=NO. In rest of the cases the prompt is:

(Y/N/E) >

If TIMEOUT is defined, the first triangle bracket of the prompt will be bright white and blinking, otherwise, its color will be identical to the color of other prompt characters.

If the global poll mode is selected (ALT-1..9 is pressed) then all the user inputs can be programmed by pressing ALT-Y or ALT-N:

[ALT-Y] - always "Yes"
[ALT-N] - always "No"

The automatic input is not used with the TRACE command.

On pressing the [E] key in reaction to the system prompt, the current line may be edited. It is worth mentioning that new line length may not exceed that of the original line. At the start of editing, [F3] may be pressed to restore the initial line. To insert a line, the following new configuration file command may be specified:

TEMPLATE

The purpose of this command, which is itself ignored by the system, is for a user to fill the line it occupies with something useful.

For example:

```
[Section1]
 ........      No trace mode here

Trace = On
 ........      User confirmation of each line

Trace = Off
  ........      No trace mode once again

 [Section2]
```

VXD

```
Purpose:    Sets the alternative path and name for a virtual device.

Syntax:     VXD=<STADARD|ON|OFF|[drive:][path]filename>
            Where, 'filename' is a virtual device name.

By default: VXD=STANDARD  or VXD=ON
```

Note:

The system assumes that the virtual device is located in the root directory and bears the WINA20.386 name. The device is required for running MS-Windows 3.0 in the enhanced mode. If such a device is absent, the system located in high memory will display an error message when starting MS-Windows 3.0 in the enhanced mode. VXD=OFF may be specified, although it might lead to system faults when working under MS-Windows 3.0.

No virtual device is required with MS-Windows 3.1 or higher. The WINA20.386 is supplied together with the system.

Some non-standard DOS commands are listed in the following pages

MKZOMBIE COMMAND

Purpose: Creates a ZOMBIE directory on specified disk.

Syntax: MKZOMBIE drv[:] [cnt]

```
drv - is a disk, where the directory will be created,
cnt - the length of the directory in entries (the length of the
      directory by default is 1 cluster)
```

Type: External command

Note:

See the description of the configuration command ZOMBIEDRV.

Description:

The main purpose of the command is to create a directory for files to be deleted. The command works in the following way:

1. A new directory is created on a disk with the help of this command. Here you can specify the length of a directory in entries, available in it. It defines size of deleted files queue. It is not recommended to specify too great a number, for it may lead to significant delays in System work. The default size - 1 cluster is enough.

2. The directive ZOMBIEDRV dsk: is written to the CONFIG.SYS file, where "dsk" is a disk where you have created ZOMBIE directory.

Instead of deleting a file, the System moves it into the directory \ZOMBIE, previously created by the given command. When the directory is completely full, the System will delete an old file and write a new one instead. In case of lack of space for new data, old files will be also deleted in the directory. Names in the directory can be duplicated, because the path is not saved during file transition. This can be seen by running the DIR command. If you work with files, the first one will be processed in this case. When running utilities, that check integrity of data on a disk, error messages are likely to appear.

RDZOMBIE COMMAND

Purpose: Deletes the ZOMBIE directory in the specified drive.

Syntax: RDZOMBIE drv[:]

where 'drv' is the drive where you want to delete the ZOMBIE directory.

Type: External command

Note:

See the description of the external DOS command MKZOMBIE and the configuraion command ZOMBIEDRV.

Description:

To delete the ZOMBIE directory perform the following:

1. If the command ZOMBIEDRV is specified in CONFI.SYS for the desired disk, you should cancel this command in config.sys file or change its argument to another disk, reboot system to force the system to recognize the new config.sys file.

2. After that you should run the RDZOMBIE command, specifying the disk where ZOMBIE directory is to be deleted.

ZOMBIEDRV

Purpose: Allows bufferization for file deletion on the specified disks.

Syntax: ZOMBIEDRV = drv[:][,drv[:][, ...]]

Note:

The main purpose of this command is to create a queue of files to be deleted. Due to this queue, it is now possible to restore recently deleted files with no data loss. The procedure is the following: when a file is deleted, the system moves it to the \ZOMBIE directory, created before-hand by the MKZOMBIE external DOS command. Directory size is specified on its creating and it defines the queue size. When the directory is full, the system will delete an old file and will write a new one in its place. Files from this directory are also deleted when there is no more space on a disk for new files. As the path is not saved when moving a file, the filenames may be recurring, as could be seen with DIR command. Only the first file is processed. Error messages may occur when running disk data check-up utilities because of some files with the same names.

RH[RUNHIGH] COMMAND

Purpose: Performs the same actions as the commands RUN and LH**[LOADHIGH]** simultaneously.

Syntax:
```
RUNHIGH filename [arg_list]
or
RH filename [arg_list]
```

Type: Internal command

Parameters:

filename - the name of loaded program,
arg_list - the list of arguments of the loaded program.

PUSHDIR COMMAND

Purpose: Allows to save directories in the stack.

Syntax: PUSHDIR {PUSH/POP/PURGE}

Type: External command

Parameters:

```
PUSH  - push the current disk and directory;
POP   - pop the current disk and directory;
PURGE - Purge the disk/directory stack.
```

HISTORY COMMAND

Purpose: Supports the "history" of commands entered from the keyboard.

Syntax: HISTORY [/H|/L] [/X|/C] [size]

Type: External command

Note: Keeps a resident part, sometimes in conventional memory.

Description:

Some commands perform the input from the keyboard using the DOS function. One of the examples of such a program is COMMAND.COM. Working with this program it is very handy to have an opportunity to recall one of the recent command lines. Exactly, what this command does. You will be able to look through all recent commands by pressing arrow keys [Up] and [Down]. Having chosen the required string, you can continue its editing, or input it by pressing [Enter].

Parameters:

```
/H     permission to use HMA;
/L     contrary to the previous one, the resident code will be situated
       in the conventional memory;
/X     use the extended memory for data;
/C     use the conventional memory for data;
size   defines the size of data buffer (When you use the extended memory
       for data, it is rounded to kilobytes)
```

Running the command without arguments is equal to:

```
HISTORY /H/X 512
```

FF COMMAND

Purpose: Finds all matching files and changes directory.

Syntax:

```
[/C[HANGE]] [/P[ROMPT]] [/M[ORE]] FILELIST
or
[/?][/H[ELP]]
```

Type: Internal command

Parameters:

```
/C or /CHANGE       changes to the directory containing the found file.
/P or /PROMPT       prompts before changing the directory.
/M or /MORE         displays found files by pages.
/? or /H or /HELP   shows this help.
```

BEEP COMMAND

Purpose: Sets a TSR, that changes the simple sound signal, generated by DOS after [Ctrl-Break] has been pressed for a complex one.

Syntax: BEEP [/M:mm][/T]

Type: External command

Description:

/M switch defines the index number of a generated signal. mm ranges from 1 to 3. Having defined /M:0 as the parameter, you can switch off the signal. If you want check or choose the necessary signal, you should press the /M switch together with the /T switch. Herewith you will hear the signal while the resident won't be loaded.

UPCOMP COMMAND

Purpose: Compares and updates files basing on previous version.

Syntax: UPCOMP <command>[/<sw>] <diff_list> [<old_names>] [<new_names>]

Type: External command

Description:

The procedure of comparison is the following: the program looks for fragments coinciding in both versions and creates the comparison protocol. This protocol contains records about the common fragments and about unique byte sequences, that are absent in the old version. Based on this protocol, the program then takes the unique new fragments from the comparison protocol and adds them to already existing in old version coinciding fragments. If the difference between versions is not very great, the size of protocol is considerably less than the size of files compared, so updating files with Upcomp program takes less time than copying new version. Using this program you can minimize costs of commercial networks and modem lines.

As Upcomp program does not use data compressing, you may use any additional archiver and save more disk space. (But do not try to compare archived files with Upcomp, because archive copies of even very similar files may have no common fragments at all.) Small size of program makes it possible to include it also into archived package, and you may easily automate updating procedure using selfunpacking archives with consequent .BAT file starting.

Starting this command without parameters displays brief information about the command.

Parameters:

<command> - the mode of Upcomp, which may be one of the following:

- c (compare) - Compare files and create comparison protocol, which name is specified in <diff_list> parameter;
- u (update) - Update version, using old one and the comparison protocol;
- l (list) - Display the contents of comparison protocol;

/<sw> - keys defining the method of comparison. This parameter may be located in any place of the command line. Default key is /m2. Key /m0 corresponds to the best quality of comparison but at the lowest rate. In this case the comparison may take about one minute per every 100 Kb. Key /m4 reduces the time of comparison but perhaps also reduces the number of coincidences found.

<diff_list> - is the Path and filename of comparison protocol.

<new_names> and <old_names>
- the pathes and filenames for new and old versions respectively. You may use *.* instead of listing all directory contents. If only one of these parameters is specified, it is assumed to be the "old_names" and its contents is compared with the content of current directory. If both parameters are absent, the default disk and directory are assumed and in 'u' mode the program will update current directory contents in accordance with already existing comparison protocol. If you wish to save your old version, we recommend you to specify different directories for old and update versions.

Note: If the file specified in <old_names> does not exist or has been corrupted, error message is displayed and then the program continues updating the files following the corrupted one. Error message also occurs if your old file was compressed by any archiver or was checked with antivirus utility. You should restore initial state of your files before using the Upcomp command.

UPCOMP COMMAND (continued)

Example:

```
      UPCOMP C DATA.LST C:\OLD_PCX\*.* C:\NEW_PCX\*.*
```

This command creates the file with comparison protocol for two catalogs specified.

LOADFIX COMMAND

Purpose: Loads the program higher than the 64k of memory and runs it.

Syntax: LOADFIX filename [arg_list]

Type: External command

Parameters:

filename - the name of loaded program
arg_list - the list of arguments of the loaded program

Note:

See the description of the configuration command LOADFIX (sys).

Description:

As the old versions of the link-program LINK.EXE contained an error, the following message may appear when loading modules created by this program into a relatively low memory area:

Packed file is corrupt

It occurs mainly when DOS is loaded in High Memory Area (DOS=HIGH command in config.sys file). In order to solve this problem use LOADFIX=ON command in config.sys file or external DOS command LOADFIX.

All these infos were taken from http://www.phystechsoft.com/ptshelp/007.html and http://www.phystechsoft.com/ptshelp/067.html