

# PC-DOS 7.1

## User's Guide

### The versions

There have been nine recorded releases of PC-DOS 7.10. This versions corresponds with the list given by Vernon Brooks, the chief of the PC-DOS 7.00 and 2000 releases.

directory	bldlevel	Date	Features	Comment (source of files)
B071		1998.04.30		PC-DOS 2000 files distributed with PC-DOS 7.1
B101		1999.?? .??	B	Symantec boot disk
B110	1.10	2000.08.01	B	(unknown source)
B111	1.11	2001.01.12	BX	Ghost 8.00 [GSS 1.0] bootsvr.sys
B119	1.19	2002.07.15	BS	Ghost 8.00 (2003) [GSS 1.0] geneeral
B126	1.26	2003.03.20	BH	Thinkpad secure disposal disk
B128	1.28	2003.07.22	BHF	Rnr31_rr disk (thinkpad)
B129	1.29	2003.09.11	BX	Ghost 11.50 [GSS 2.5] cddvdimg.bin
B130	1.30	2003.09.26	BS	Ghost 11.50 [GSS 2.5] general
B132	1.32	2003.12.01	BHFA	Server Guide Toolkit 1.101 and 1.037
B134	1.34	2004.02.27	E	SGKT utility (ibmcdet.sys only)
U132				Non-server specific SGKT utilities.

B     System files: IBMDOS.COM, IBMBIO.COM, COMMAND.COM  
S     Boot sectors: BOOTSECT.COM, BSFLOPPY.COM  
X     These files need to be extracted from a boot-image file, given in the comments.  
H     XMM manager Himem.sys 3.15  
F     FDISK32.EXE  
E     El-torito cd-rom driver IBMCDDET.SYS  
A     Other Dos files: BLDLEVEL.COM, DEBUG.COM, DYNALOAD.COM,  
       FORMAT.COM, FORMAT32.COM, MSCDEX.COM, NOINT25.COM  
GSS   Ghost Solutions Suite. DOS was dropped in version 3.0 and later.

MSCDEX reports a version of 2.30.

The command.com in LZ-DOS is a patched version of build 1.19, with a 32-byte header file.

### Installing PC-DOS 7.1

The various PC-DOS 7.1 sources provide pre-made boot images, but no install utility. Given that this DOS is intended for whole-of-partition activities, rather than as a desktop utility, a number of safety features have been disabled.

One should exercise a good deal of caution with these files.

The supplied utilities for making DOS are as follows:

1. Boot-sector files in GSS distributions. (B119, B130) and DISKDOS.EXE (U132).
2. SYS.COM command (B071)
3. MAKEBOOT.BAT command (B132)

#### 4. LOADDSKF.EXE command (U132)

A utility that can prepare a bootable diskette image, given its boot sector, could be used to prepare a boot-image file. This is how the GSS solutions do this.

MAKEBOOT.BAT prepares a general fat32 format, and copies the boot files to the root directory.

SYS.COM does not know about FAT32, although it can be used to prepare a bootable floppy disk. PC-DOS 7.10 will boot from a generic PC-DOS boot-sector.

LOADDSKF is used to copy pre-selected images onto a floppy disk. The server guide distributions B132 contain disk images that use PC-DOS 7.0 boot sectors.

Blank diskettes that contain the B119 and B130 bsfloppy boot-sectors appear as zip files in these directories. One then copies the relevant files onto these diskette images.

The DISKDOS utility in U132 can be used to read and write boot sectors to floppy disks and hard drives, from existing files.

## IBM ServerGuide Scripting Toolkit

User's Reference Version 1.3.07

The following is an extract of the ServerGuide Scripting Toolkit documentation. To this I have added sections on some extra commands. The utilities are in the U132 directory.

The added commands are: \_BOOT\_HD.NOZ, BLDLEVEL.COM, DISKDOS.EXE, IBMCDDET.SYS, NOINT25.COM. The section on CLINI.EXE is large, and has been moved to the end of the section.

\_BOOT\_HD.NOZ is a zero-byte file created to

\_BOOT\_HD.NOZ

When the boot media is a CD, the boot media cannot be altered using the altboot option because the CD is non-writable. To prevent the installation process from stopping and prompting the user to remove the boot media when using a CD, a dummy file must be created on the boot diskette image. This method is available as a base option for IBM PC DOS 7.1 only.

Before saving the boot image to the CD in your CD creator software, create a zero-byte length file on the boot diskette image called \_BOOT\_HD.NOZ. When the DOS kernel sees the \_BOOT\_HD.NOZ file, it displays the Press the ENTER key to boot from CD or DVD message and sets a 5-second timer. The user must then press any key, within 5 seconds, to boot from CD; otherwise, the system defaults to booting from the hard disk drive.

You must then boot from the CD, perform any necessary updates, configure ServeRAID, and create an operating system partition. Once this is done and the Windows files are being copied to the operating system partition, the installation can continue unattended using the \_BOOT\_HD.NOZ file you created.

(This feature appeared in build levels 1.26 and later.)

## ALTBOOT.EXE

The altboot.exe utility permits the ServerGuide Scripting Toolkit to bypass startup (boot) of the diskette drive in the target server during the deployment process. This is done by modifying the boot sector on the diskette so that the diskette is ignored or the first active partition on the first hard disk is explicitly started (booted).

The altboot.exe utility has the following command-line syntax:

```
altboot </h|/r|/?> [/f:filename]
```

Parameter	Description
/?	Displays all parameters
/h	Saves a backup image of the boot sector on the diskette, and forces the target server to boot from the hard disk regardless of the startup (boot) sequence configured in the target server BIOS
/r	Restores the boot sector on the diskette from the diskette image
/f:filename	Specifies a file name for the backup image of the boot sector, where filename is the name of the file. The default file name is bootsec.img. This parameter is only valid when the /b or /r parameter is used.

The altboot.exe utility returns the following values to indicate status:

Value	Description
0	Success
1	File error (invalid boot-sector image, file access errors)
2	Diskette drive or diskette error
3	Command-line parameter error

The following examples illustrate altboot.exe utility usage.

Example	Description
altboot /f:a:\saveboot.sec	Saves an image of the boot sector as a:\saveboot.sec
altboot /r	Restores the boot sector on the diskette from the default file name a:\bootsec.img
altboot /r /f:a:\test.img	Restores the boot sector on the diskette from the a:\test.img file
altboot /h /f:a:\floppy.img	Explicitly starts (boots) the target server from the hard disk and saves an image of the boot sector as a:\floppy.img

## BLDLEVEL.COM

The bldlevel.com utility returns the version number (build level) embedded in the file. It uses a standard format in the style of OS/2 programs. It reports only the files that have a build-level included.

The bldlevel.com utility has the following command-line syntax:

Bldlevel [drive][path]filespec

Parameter	Description
Drive	The drive containing the files
Path	The relative or absolute path
Filespec	One or more files, including wildcards.

## DISKDOS.EXE

The diskdos.exe utility copies a boot sector file to or from a disk. This can be used to prepare a bootable floppy disk or hard drive given the bootsector.

The diskdos.exe utility has the following command-line syntax:

Diskdos /F=filename /D=drive[:] [/R=R|W] [/V]

Parameter	Description
/f=filename	The name of the file that is used to create or store the boot-sector
/d=drive	The letter, with optional colon, that the boot-sector is to be read or written.
/r=r w	Read (ie boot-sector to file), or write (ie file to boot-sector)
/v	Verbose, shows the name of the file, drive, and the boot sector listing

## FDISK32.EXE

The fdisk32.exe utility manages and configures partitions of hard disks in the target server. The ServerGuide Scripting Toolkit uses this utility during the deployment process to configure hard disks before installing operating-system files.

The fdisk32.exe utility has the following command-line syntax:

fdisk32 [/status|d</pri:n|ext:n|log:n|mbr|delete:all>]

Parameter	Description
/status	Displays the status of all fixed drives in the target server
d	Specifies the hard disk number to be configured, where d can be any positive integer.
/pri:n	Creates a primary DOS partition of size n in MB on the specified hard disk
/ext:n	Creates an extended DOS partition of size n in MB on the specified hard disk
/log:n	Creates a logical drive of size n in MB in the extended partition on the specified hard disk

/mbr	Writes a new master boot record to the hard disk
/delete:all	Deletes all partitions on the specified hard disk

The following examples illustrate fdisk32.exe utility usage.

Example	Description
fdisk32 1 /delete:all	Deletes all partitions on the hard disk 1
fdisk32 1 /pri:4096	Creates a 4096 MB (4 GB) primary DOS partition on hard disk 1
fdisk32 1 /ext:4000	Creates a 4000 MB extended DOS partition on hard disk 1
fdisk32 1 /log:2000	Creates a 2000 MB logical drive in the extended partition on hard disk 1

## FINDRAM.EXE

The findram.exe utility determines the drive letter of the RAM drive that was created by IBM PC DOS 7.1 when the target server started (booted). The ServerGuide Scripting Toolkit calls this utility during the deployment process automatically, when required. The findram.exe utility returns a numerical value from 3-26, which corresponds to the letters C-Z.

## FORMAT32.EXE

The format32.exe utility formats a FAT32 partition on the hard disk of a target server for use with IBM PC DOS 7.1. The ServerGuide Scripting Toolkit uses this utility during the deployment process to format hard disks before installing operating-system files.

The format32.exe utility has the following command-line syntax:

```
format32 <drive:> [/v:label] [/q] [/autotest]
```

Parameter	Description
drive	Specifies the drive to format on the target server
/v:label	Specifies an optional volume label for the specified drive
/q	Performs a quick-format of the specified drive
/autotest	Does not display any prompts during the formatting process

The following examples illustrate format32.exe utility usage.

Example	Description
format32 c: /q /autotest	Formats the c: drive on the target server using a quick-format and requiring no interaction

## IBMCDDET.SYS

The ibmcdet.sys driver allows DOS to find the booted cdrom through the el-torito chain. Thus it can be used on an image file on a bootable cdrom.

## LINECOMB.EXE

The linecomb.exe utility is a 16-bit DOS executable that combines the lines of a file into one single line. This is used in Linux-scripted installations to create a compatible syslinux.cfg file for the deployment scenarios.

The linecomb.exe utility has the following command-line syntax:

`linecomb source_file output_file`

Parameter	Description
<i>source_file</i>	Specifies the name of the source file to be scanned.
<i>output_file</i>	Specifies the name of the output file.

The following example illustrates linecomb.exe utility usage.

Example	Description
<code>linecomb multiline.txt single.txt</code>	This command takes each individual line in multiline.txt and combines them into one line, each separated by spaces. The output file name is single.txt.

The linecomb utility issues the following return codes to indicate status:

- 1 - Source file not found
- 2 - Unable to create target file

## LOADDSKF.EXE

The loaddiskf.exe utility uses a diskette image file to create a DOS-startable (bootable) diskette. The file must contain a diskette image. The destination diskette is checked to make sure that it is the same format as the one from which the diskette image was originally created. The destination diskette must be blank. The ServerGuide Scripting Toolkit calls this utility during the deployment process automatically, when required.

Restriction: A known problem can prevent the loaddiskf.exe utility from working properly under Windows 2000. The problem occurs when the utility or the disk images are in directories that contain long file names.

## NOINT25.COM

Use this utility to disable the old DOS absolute disk read (int 25h) and execute a program. Windows 98 has intentionally disabled this old API and some programs such as the Windows 98 version of CHKDSK will not run under PC DOS 7.1 without this function being disabled.

The noint25.com utility has the following command-line syntax:

noint25.com [program [parameters]]

Parameter	Description
<i>program</i>	The program requiring int-25 calls to be hidden
<i>output_file</i>	Parameters passed to the program

## REBOOT.COM

The reboot.com utility restarts (reboots) the target server. The Scripting Toolkit uses this utility during the deployment process to restart the target server, when required. This utility does not perform a disk reset or flush hard disk cache. There are no command line parameters for the reboot.com utility.

## SAVEDSKF.EXE

The savedskf.exe utility creates a diskette image file from a DOS-startable (bootable) diskette. The destination is checked to make sure that there is enough free space to save the diskette image. Sectors above the highest sector containing data, as indicated by the file-allocation table, are not written into the diskette image. The ServerGuide Scripting Toolkit calls this utility during the deployment process automatically, when required.

## SAVESTAT.EXE

The savestat.exe utility enables you to store and retrieve up to five values in CMOS, using persistent-state information, on the target server. This utility is useful to pass information to the deployment process after a restart (reboot) occurs, such as where in the deployment process to continue after the restart.

The 16-bit version of savestat.exe runs only in DOS. The 32-bit version of savestat.exe is for use on Microsoft Windows 2000 Server, Windows Server 2003, and the 32-bit version of Windows Preinstallation Environment 2005. The x64 version of savestat.exe is for use on the 64-bit version of Windows Preinstallation Environment 2005.

Values are returned using the errorlevel DOS environment variable or the return code in Windows so that you can create a batch file to branch according to the value returned. The savestat.exe utility that comes with the ServerGuide Scripting Toolkit has the following command-line syntax:

savestat </setn=value|/getn|/reset>

Parameter	Description
/setn=value	Saves an integer value, value, to the nth location in persistent-storage memory, where n can be any number 1-5 and value can be any number 0-254. The return values are: 0 if successful 1 if not successful
/getn	Retrieves a value currently set in the nth location in persistent-storage

	memory, where n can be any number 1-5. The return value is the number stored, or 255 if not successful.
/reset	Resets all persistent-storage memory to zero values. The return values are: 0 if successful 1 if not successful

Note: Savestat.exe is incompatible with emm386

The following examples illustrate savestat.exe utility usage.

Example	Description
savestat /set2=100	Stores the value 100 in the second persistent-storage memory location
savestat /get2  if errorlevel 100 goto end if errorlevel 1 goto level1  :level1 call level1.bat  :end	Retrieves the value of the second persistent-storage memory location and branches in the batch file according to the value returned

## SCRUB3.EXE

The scrub3.exe utility performs secure data disposal on the target server hard disks. This utility erases all data on one or more hard disks.

Attention: You cannot retrieve data from a hard disk after running this utility on the target server, so make sure you do not need the data anymore before including this utility in a deployment scenario.

The scrub3.exe utility has the following command-line syntax:

```
scrub3 [/?] [/Q=NO] [/S=NO] [/D=<drive|ALL>>] [/L=level|/W=writes]
```

Parameter	Description
/?	Displays all parameters
/Q=NO	Causes the utility to display a maximum number of messages during the process
/S=NO	Prevents the scrub signature from being written to the hard disk that is being erased
/D=<drive ALL>	Specifies the drive on the target server to erase, or all drives when ALL is specified
/L=level	Defines the level of security to be used when cleaning the target server hard disk. Valid values are:  1 - (Limited security) The master boot record and some sectors are overwritten once. 2 - (Medium security) All sectors are overwritten once. 3 - (High security) All sectors are overwritten four times. 4 - (U.S. Department of Defense-compliant security) All sectors are overwritten seven times.

/W=writes	Defines the number of times each sector on the target server hard disk is overwritten. Valid values for writes are any positive integer.
-----------	--

The following examples illustrate scrub3.exe utility usage.

Example	Description
scrub3 /L=1 /D=ALL	Overwrites the master boot record, the first 100 sectors of each partition, and the last two sectors on every hard disk in the target server with a 0x0000 pattern
scrub3 /Q=NO /D=2 /L=2	Overwrites every sector on hard disk number 2 with a pattern 0x0000, but does not erase any other hard disk on the target server, while displaying status messages on the screen
scrub3 /D=ALL /L=3	Overwrites every byte on every sector of every hard disk in the target server four times This is a very secure method for data disposal, but will take a long time to complete.

## SLEEP.EXE

The sleep.exe utility pauses the deployment process on the target server for a specified amount of time. After the time-interval elapses, the target server resumes the deployment process.

You can exit before the time-interval elapses by pressing any key on the target server.

The sleep.exe utility has the following command-line syntax:

sleep [n]

Parameter	Description
n	Specifies n in seconds for the target server to pause before resuming the deployment process

The following examples illustrate sleep.exe utility usage.

Example	Description
sleep 10	Pauses for 10 seconds before resuming the deployment process

## TSHUDWN.EXE

The tshudwn.exe utility is a 16-bit DOS shutdown utility.

The tshudwn.exe utility has the following command-line syntax:

tshudwn

## CLINI.EXE

The Command Line INI utility can perform the following functions:

1. Write information to an INI file:
  - a. Add new sections, items, or values
  - b. Remove sections, items, or values
  - c. Change existing sections, items, or values
  - d. Comment or uncomment sections, items, or values
2. Read information from an INI file:
  - a. Read items and store all or part of the value as an environment variable
  - b. Read items and check all or part of the value for strings, substrings, or tokens
3. Merge information from one INI to another.

Three versions of the Command Line INI utility come with the ServerGuide Scripting Toolkit:

- A 16-bit version for DOS.
- A 32-bit version for use on Microsoft Windows 32-bit operating systems and the 32-bit version of Windows Preinstallation Environment (Windows PE) 2005. The 32-bit version was formerly named `clini32.exe`.
- A 64-bit version for Windows x64 operating systems and for Windows PE 2005 (x64).

Storing a value as an environment variable is done by creating a batch file that contains a command to set the environment variable. You must then call the batch file to set the environment variable. By default, the batch file is named `cliniset.bat`. If the batch file already exists, it is deleted and recreated with the new information.

In addition to setting values, the `clini.exe` program can append values to existing items in an INI file. By default, no delimiter is used to append values. A delimiter can be specified, if required. Appending values provides the ability to 'build' values in the INI file by issuing multiple commands. When reading values from an INI file to set an environment variable, the values can be tokenized to specify a particular token.

The `clini.exe` program checks the number of characters on the command line against the current limits of 127 characters for the DOS version and 255 characters for the Windows version. A message is displayed if the characters exceed the limit. The `/O` parameter overrides character-limit checking.

The `clini.exe` utility has the following command-line syntax:

```
clini <filename> <[filename2 [/ES] [/A/U/P]]> <[/S:section] [/I:item]  
[/V:value[/A:value/U:value/E:variable /=:string/C:string/CT:string]]> [/B:file_name]  
[/D:delimiter] [/T:n] [/R] <[/CMT/UCMT] [/AI] [/CC:character]]> [/NS] [/N] [/O]
```

Parameter	Description
filename	Defines the fully qualified path to the INI file to process
filename2	Defines the fully qualified path to an INI file to merge information into

	from filename. All values in filename are copied into filename2, replacing the value of any preexisting items in filename2.
/ES	Specifies to merge only the items or values in the empty section.
/A	Specifies to append values from items in filename to the items in filename2 instead of replacing them. An optional delimiter can be specified using the /D:delimiter parameter.
/U	Specifies to uniquely append values from items in filename to the items in filename2 instead of replacing them; only if the value doesn't already exist. An optional delimiter can be specified using the /D:delimiter parameter.
/P	Specifies that the data in filename2 is persistent. If duplicate items are found, they are not replaced.
/S:section	Specifies the name of the section within the INI file to write or to read
/I:item	Specifies the name of the item within the INI file to write or to read
/V:value	Specifies the value to write to the INI file
/A:value	Specifies the value to append to the existing item in the INI file. The /I parameter is required to use the /A:value parameter.
/U:value	Specifies a unique value to append to the existing item in the INI file, only if this value does not already exist for the item. The /I parameter is required to use the /U:value parameter.
/E	Convert multiple Items to Environment Variables. The Item name is used for the environment variable name. Use the /NS parameter to replace any spaces in the item names with underscore characters when creating the Environment Variables, if spaces are not desired.
/E:variable	Specifies the environment variable used to store the value of the item from the INI file. The /I parameter is required to use the /E:variable parameter. If the item specified by the /I parameter does not exist, or the section specified by the /S parameter does not exist, the environment variable has no value in the batch file created by clini.exe. If the environment variable exists on the system, it is deleted when the batch file runs.
/=:string	Verifies that the value of the item is equal to string, returning a value of 0 if true and 100 if false.
/C:string	Verifies that value of the item has string as a substring, returning a value of 0 if true and 100 if false.
/CT:string	Verifies that the value of the item has string as one of the tokens, returning a value of 0 if true and 100 if false. The default delimiter is a comma unless the /D:delimiter option is specified.
/B:filename	Defines the fully qualified path and file name of the batch file to create for setting the environment variable. The default is CLIniSet.bat if no file name is specified for this parameter. This parameter is only valid when the /E parameter is used.
/D:delimiter	Specifies a delimiter to use when appending values to an item in an INI file or reading tokens from an INI file. This parameter is not valid if the /V parameter is used. The /D parameter is valid only with the /A, /U, or /E parameters. Using the /D parameter without one of these three parameters results in a syntax error.
/T:n	Specifies the token in a delimited value to set as the specified environment variable, where n is a positive integer. The default delimiter is a comma unless otherwise specified with the /D parameter. This parameter is only valid with the /E parameter.
/R	Removes the specified section, item, or value from the INI file. Removing the last item in a section also removes the section.
/CMT	Specifies to Comment out the line indicated by the Section, Item, or Value parameter, if it exists in the INI file. It also allows use of the /AI parameter.
/UCMT	Specifies to Uncomment the line indicated by the Section, Item, or Value

	parameter, if it exists in the INI file. It also allows use of the /AI parameter.
/CC:character	Specifies the comment character to use when commenting or uncommenting lines. If omitted, the default comment character is the semicolon. This parameter is only valid with the /CMT or /UCMT parameters.
/AI	Specifies to explicitly treat the /V parameter as the value to all items when commenting or uncommenting. This parameter is only valid when using the /CMT or /UCMT parameters.
/N	Deletes an existing INI file and creates a new INI file. This parameter is not valid with the /E parameter.
/NS	Omits spaces around ? = ? when writing items into INI files. By default, the clini.exe utility concatenates spaces around ? = ? when writing items.
/O	Overrides the command-line character count. The number of characters on the command line is automatically determined by this utility. An error message is displayed when the character limit is reached, unless you override this feature. The DOS version is limited to 127 characters, while the Windows version is limited to 255 characters.

The clini.exe utility returns the following values to indicate status:

Value	Description
0	Success or true
1	Syntax error
2	Program error
3	Destination is read-only
4	Current working directory is read-only
5	File not found
100	False

- End of Document -