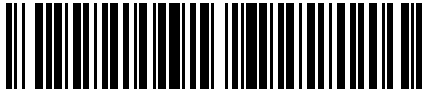
PDT 3200

Technical Reference Guide



70-31468-01
Revision A — July, 1997

Symbol Technologies, Inc. One Symbol Plaza, Holtsville N.Y. 11742

PDT 3200
Technical Reference Guide

70-31468-01
Revision A
July, 1997

Symbol Technologies, Inc.
One Symbol Plaza
Holtsville, N.Y. 11742
http://www.symbol.com

Contents

Chapter 1. Default Disk Driver Organization and Unit Configurations

Chapter 2. System Utilities

Chapter 3. Resetting the PDT 3200

Chapter 4. Using PC Cards

Chapter 5. ROM-DOS Commands

About This Manual

This manual is a supplement to the *PDT 3200 Product Reference Guide*. It is provided as a reference for system administrators, developers, and programmers who want to create end-user solutions for the PDT 3200 terminal.

This reference guide contains technical information about the PDT 3200's system configuration, disk drives, utilities, and DOS commands. It also provides information about using PC cards with the PDT 3200 and resetting the unit.

Related Publications

- *PDT 3200 Quick Reference Guide, p/n 70-31467-xx*
- *PDT 3200 Product Reference Guide, p/n 70-31466-xx*
- *PDT 3200 Run Time Library Reference Manual, p/n 70-31577-xx*
- *CRD 3200 Cradle Quick Reference Guide, p/n 70-31469-xx*

Conventions

Keystrokes are indicated with the angle brackets as follows:

ENTER	Identifies a key.
ALT+X	Identifies a simultaneous key combination.
BKSP, SHIFT, ON	Identifies a key sequence.

Escape sequences described do not contain angle brackets characters. For example:

ESC[3;4f	Indicates the sequence: escape character, left bracket, numeral 3, semicolon, numeral 4, and letter f.

Typeface conventions used include:

Italics	Indicates first time a new item is used. A definition follows the italicized terms. Italics also indicate book titles or information that must be replaced by an actual value. Italics also express menu titles.
Syntax	Indicates text entered by the user.
Screen	Indicates a text displayed on a screen or terminal.

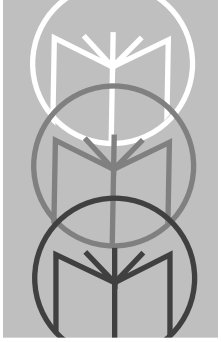
Note: Indicates tips or special requirements.

Caution

Indicates conditions that can cause equipment damage or data loss.

WARNING

Warnings indicate procedures that are potentially dangerous and should therefore be performed only by Symbol-authorized repair personnel.



Chapter 1 Default Disk Driver Organization and Unit Configurations

Chapter Contents

Introduction

This chapter details the default contents of the disk drives and the organization of system software on the PDT 3200.

File System Format

Overview

The PDT 3200's default file system format is as follows:

- Drive A is a read-only flash drive. The contents of this drive cannot be changed.
- Drive B is a read-only flash drive that you can update by using the core update program, COREXFER.EXE, from a host machine. (Normal operation will not require updating of this drive.)
- Drive C is a resident flash disk drive that is accessed with the FLASHDSK.SYS device driver. You can use this drive to store applications and data.
- Drive D is a RAM disk accessed with the VDISK.SYS device driver. This drive can be used to store applications as well as data. However, you should take great care in selecting files for this drive. As with any RAM drive, its contents will be lost if power is removed.
- Additional drives are mapped to ATA flash PC cards or to network drives via radio frequency (RF).

This organization provides an easy-to-use, extensible system that allows a high degree of performance, usability, and customization. The following sections more fully describe the contents and intended uses of each of the PDT 3200's drives.

Drive A

Drive A is a disk image that is located in flash along with the BIOS and ROM-DOS. A read-only drive, it is intended to remain secure throughout the life of the unit. Drive A contains the following files:

- AUTOEXEC.BAT, the first file in the startup sequence
- COMMAND.COM, the DOS shell
- CONFIG.SAF, the first file in the safe-boot sequence
- CONFIG.SYS, the first file in the boot sequence
- REV.COM, the utility that reports the firmware revision

The CONFIG.SYS and AUTOEXEC.BAT files contain only basic commands that the unit needs to run correctly (see Figure 1-1 and Figure 1-2). Both files chain to their respective counterparts on the B drive, as described in the next section.

The CONFIG.SAF file (see Figure 1-3) is used in the safe-boot sequence. For more information about the CONFIG.SAF file, see page 1-5.

```
REM ***********************************************
REM Initial CONFIG.SYS file for starting the system.
REM ***********************************************
NEWFILE = B:\CONFIG.SYS
```

Figure 1-1. CONFIG.SYS File on Drive A

```
@echo off
REM ************************************************
REM Initial AUTOEXEC.BAT file for starting the system.
REM ************************************************
VER
PATH=A:\
B:\AUTOEXEC.BAT
```

Figure 1-2. AUTOEXEC.BAT File on Drive A

```
REM *********************************
REM Initial CONFIG.SYS for safe boot.
REM *********************************
NEWFILE = B:\CONFIG.SAF
```

Figure 1-3. CONFIG.SAF File on Drive A

Drive B

Drive B is a disk image that is located in flash. A read-only drive, it is intended to be used to store mandatory utilities and initialize the boot process. During normal use this drive will not be updated. Drive B includes the following files:

- AUTOEXEC.BAT, the second file in the startup sequence
- CFGDEV.SYS, the system parameter configuration device driver

- CONFIG.SAF, the second file in the safe-boot sequence
- CONFIG.SYS, the second file in the boot sequence
- COREUPD.COM, the utility that updates the firmware image
- DECODE.SYS, the bar code decoding device driver
- FLASHDSK.SYS, the resident flash disk device driver
- FORMAT.COM, the disk-reformatting utility
- FUNCTEST.COM, the functional test for system components
- LD.BAT, the batch file for RESPONSE.BAT processing
- LOCK.COM, the utility for locking or unlocking the resident flash disk and RAM disk
- ORGANIZE.COM, the utility for reorganizing flash disk data
- PM.COM, the power-management utility
- VDISK.SYS, the RAM disk device driver
- XFER.EXE, the serial transfer utility

The CONFIG.SYS file (see Figure 1-4) will load the FLASHDSK.SYS driver to create the C drive and the VDISK.SYS driver to create the D drive. It will then load DECODE.SYS and CFGDEV.SYS before chaining to its counterpart on drive C. The AUTOEXEC.BAT file (see Figure 1-5) will chain to its counterpart on drive C, except during a safe boot.

The purpose of the CONFIG.SAF file (see Figure 1-6) is to provide a minimal boot configuration that preserves the flash and RAM drives. When the safe-boot sequence is initiated, the CONFIG.SAF file is processed instead of the CONFIG.SYS file. This prevents chaining into the drive C initialization files by omitting the NEWFILE = C:\CONFIG.SYS command. It also eliminates the loading of the DECODE.SYS and CFGDEV.SYS drivers.

```
REM ****************************************
REM Standard CONFIG.SYS for building system.
REM ****************************************
REM ***************************
REM Create flash and RAM drives
REM ***************************
DEVICE = B:\FLASHDSK.SYS 4096
DEVICE = B:=VDISK.SYS 4096 /e
REM *********************************
REM Load decode/configuration drivers
REM *********************************
DEVICE = B:\DECODE.SYS
DEVICE = B:\CFGDEV.SYS
NEWFILE = C:\CONFIG.SYS
```

Figure 1-4. CONFIG.SYS File on Drive B

```
@echo off
REM ****************************************
REM Standard AUTOEXEC.BAT for starting system.
REM ****************************************
PATH=%PATH%;B:\
C:
IF EXIST A:\CONFIG.SAF IF EXIST C:\AUTOEXEC.BAT C:\AUTOEXEC.BAT
```

Figure 1-5. AUTOEXEC.BAT File on Drive B

```
REM *********************************
REM Standard CONFIG.SYS for safe boot.
REM *********************************
REM ***************************
REM Recover flash and RAM drives
REM ***************************
DEVICE = B:\FLASHDSK.SYS 4096
DEVICE = B:\VDISK.SYS 4096 /e
```

Figure 1-6. CONFIG.SAF File on Drive B

Drive C

Drive C is a resident flash disk drive that is accessed via the FLASHDSK.SYS device driver. By default, this drive is configured to have full read and write access. The default CONFIG.SYS and AUTOEXEC.BAT files reside here. You can modify these files to customize the system for an application. They will be called after the CONFIG.SYS and AUTOEXEC.BAT files on drives A and B are processed.

The PDT 3200 Configuration Utility gives you the ability to customize what software tools and utilities are placed on the unit. By default, the utility installs this software to the C drive. If the unit is intended to be used with PC cards, then the Phoenix PC card drivers are installed in the PCM subdirectory on the C drive, and the proper entries will be transferred into the CONFIG.SYS and AUTOEXEC.BAT files. The exact configuration of files and the contents of the CONFIG.SYS and AUTOEXEC.BAT files will change according to the configuration of the unit.

For RF capability, additional files for network connectivity would be required. These might include a Telnet program or peer-to-peer networking tools, depending on what you choose to install.

If something happens to a unit and data integrity becomes questionable, use the CHKDSK utility to detect and correct errors on drive C. (CHKDSK.COM does not come installed on the PDT 3200. You can download it using XFER or the PDT 3200 Configuration Utility.) You can also use ORGANIZE.COM to recover unused sectors.

Drive D

Drive D is a RAM disk drive. By default, this drive resides in extended memory and is sized 1MB less than the total RAM in the PDT 3200 unit (either 1MB or 3MB).

Drive D can be used for short-term data storage. Programs that need to be loaded into memory and then removed from memory quickly can also be located there. The drive can also be used for scratch disk space or temporary files.

The PDT 3200 preserves the data on drive D between warm or cold boots by checking for an existing RAM disk. However, only minimal checking is performed on any disk that is found; if something happens to a unit and data integrity becomes questionable, you should use the CHKDSK utility to detect and correct errors on drive D. (CHKDSK.COM does not come installed on the PDT 3200. You can download it using XFER or the PDT 3200 Configuration Utility.)

Drive E

Drive E is the ATA flash card. The PC card looks like a hard disk drive to the operating system and the user. More flexible than flash disk drive C, it can be used for safer and more permanent bulk storage of batch data than the RAM disk (drive D). This drive exists only on systems configured to use ATA flash cards.

Drive E could also be a peer-to-peer or client-server network drive that is accessed through an RF or Ethernet network card link. This option allows the developer to make many network drives (drives E, F, etc.) available to applications.

System Software

BIOS and DOS

The PDT 3200 uses a modified version of General Software's BIOS with Datalight's ROM-DOS. Both products are burned into system flash in a single 256K image. The A drive is included in the image, physically addressed just below the BIOS image.

ROM-DOS uses the Datalight COMMAND.COM processor. This processor is fully Microsoft 6.2 compatible, except that it occupies about half the space in memory. In addition, drive C contains Datalight DOS files in a DOS subdirectory.

Device Drivers and Utilities

The following device drivers are always present on the PDT 3200's drive B:

- CFGDEV.SYS
- DECODE.SYS
- FLASHDSK.SYS
- VDISK.SYS

The following utilities are always present on drive B:

- COREUPD.COM
- FORMAT.COM
- FUNCTEST.COM
- LOCK.COM
- ORGANIZE.COM
- PM.COM
- XFER.EXE

See page 1-5 for descriptions of these drivers and utilities. See chapter two for more information about them.

PC Card and RF Networking Software

PC card device drivers and utilities as well as RF networking solutions can be installed to drive C (the flash drive) or drive D (the RAM drive). You can use the PDT 3200 Configuration Utility to download the necessary files to the PDT 3200.

Unit Configurations

Several configurations are possible for the system software on the unit, depending upon the type of hardware that is to be supported. Two standard configurations are listed below, along with directory structures and default CONFIG.SYS and AUTOEXEC.BAT file descriptions.

Default Configuration

The configuration created by DEFAULT.CFG in the PDT 3200 Configuration Utility will create a C drive that contains PALPRO.EXE, part of the Symbol Portable Applications Library (PAL). The data files will be stored on drive D. PC card drivers able to support ATA cards will be loaded.

```
FILES = 30
REM ------------------------------------------------------
REM --- Phoenix Card and Socket services for accessing PC Cards
REM ------------------------------------------------------
device = c:\pcm\cnfignam.exe /NORMAL
device = c:\pcm\pcmssit.exe
device = c:\pcm\pcmcs.exe
device = c:\pcm\pcmata.sys
```

Figure 1-7. CONFIG.SYS File on Drive C for Default Configuration

```
REM ------------------------------------------------------
REM --- The following lines set up default parameters for some
REM --- environment variables. These lines may be modified or
REM --- overridden in the USER section below.
REM ------------------------------------------------------

set prompt=$p$g
set dircmd=/ogn /p
REM ------------------------------------------------------
REM --- The following section is for customized user entries.
REM ------------------------------------------------------
REM ------------------------------------------------------
REM --- Insert user-specific options and commands here.
```

```
REM ------------------------------------------------------
REM ------------------------------------------------------
REM --- The following lines add system components to the PATH
REM --- and run the main application executable, if one was
REM --- specified.
REM ------------------------------------------------------
IF EXIST c:\bparams.ini copy c:\bparams.ini PARAMS
set path=c:\;c:\dos;%path%
c:
cd\
upgstart
```

Figure 1-8. AUTOEXEC.BAT File on Drive C for Default Configuration

I/O PC Card Support

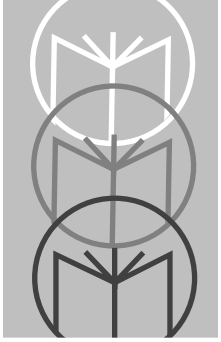
The PDT 3200 Configuration Utility has an option for support of I/O cards. This option will download the Phoenix Super Client Driver (PCMSCD.EXE) to the C:\PCM subdirectory on the PDT 3200. It will also add the command for loading of this driver to the CONFIG.SYS file. The CONFIG.SYS and AUTOEXEC.BAT files below were created by eliminating ATA card support and adding I/O card support to DEFAULT.CFG.

```
FILES = 30
REM ------------------------------------------------------
REM --- Phoenix Card and Socket services for accessing PC Cards
REM ------------------------------------------------------
device = c:\pcm\cnfignam.exe /NORMAL
device = c:\pcm\pcmssit.exe
device = c:\pcm\pcmcs.exe
device = c:\pcm\pcmscd.exe
```

Figure 1-9. CONFIG.SYS File on Drive C for I/O Card Support

```
REM ------------------------------------------------------
REM --- The following lines set up default parameters for some
REM --- environment variables. These lines may be modified or
REM --- overridden in the USER section below.
REM ------------------------------------------------------
set prompt=$p$g
set dircmd=/ogn /p
REM ------------------------------------------------------
REM --- The following section is for customized user entries.
REM ------------------------------------------------------
REM ------------------------------------------------------
REM --- Insert user-specific options and commands here.
REM ------------------------------------------------------
REM ------------------------------------------------------
REM --- The following lines adds system components to the PATH
REM --- and runs the main application executable, if one was
REM --- specified.
REM ------------------------------------------------------
IF EXIST c:\bparams.ini copy c:\bparams.ini PARAMS
set path=c:\;c:\dos;%path%
c:
cd\
upgstart
```

Figure 1-10. AUTOEXEC.BAT File on Drive C for I/O Card Support



Chapter 2 System Utilities

Chapter Contents

Introduction

This chapter describes the utilities that are designed to be run on a PDT 3200 unit. For each device driver or executable file, a brief description is given, followed by any command line parameters and their meanings. Any parameters listed may be supplied using either upper- or lowercase letters, and the hyphen (-) may be substituted for the forward slash (/).

CFGDEV.SYS

CFGDEV.SYS is located on drive B of the PDT 3200 unit. When CFGDEV.SYS is loaded by B:\CONFIG.SYS, a logical character-based device called PARAMS is created. This device receives programming strings that modify the configuration parameters that control the behavior of several system components. There are three ways for the developer to send programming strings to the PARAMS device.

- **Bar Code Scanning** Special bar code labels beginning with $+$- and ending with EE can be scanned using any bar code scanning device. The *PDT 3200 Product Reference Guide* contains many useful labels.
- **File Copy** Files containing the bar code programming strings can be created and copied to the PARAMS device. The PDT 3200 Configuration Utility creates a file called BPARAMS.INI that contains all of the configuration settings. C:\AUTOEXEC.BAT on the unit contains a command to copy this file to PARAMS (IF EXIST c:\bparams.ini copy c:\bparams.ini PARAMS). You may also create your own file containing configuration settings.

  Example: Use D2 defaults with CTRL-ALT-DEL sequence disabled.

  1. Create a MYPARAMS.INI file with the following text: $+$-D2E00EE. D2 sets the D2 defaults, E00 disables CTRL-ALT-DEL.
  2. Copy MYPARAMS.INI to the PARAMS device using the following DOS command: COPY MYPARAMS.INI PARAMS
- **Developer's Toolkit API** The API in the PDT 3200 Developer's ToolKit contains two function calls that allow you to send configuration strings to the PARAMS driver from within a C application. See CFG_Write() and CFG_Read() in the Developer's Toolkit.

Attempts to create, delete, or modify a file called PARAMS will conflict with the PARAMS device. You cannot eliminate the loading of CFGDEV.SYS except through the safe-boot process.

DECODE.SYS

This driver controls bar code scanning devices and decodes bar code labels into text. If special programming bar codes are read, the decoded information will be sent to the PARAMS device (see CFGDEV.SYS above). Otherwise, the resulting text is placed in the keyboard buffer for reading by the application. DECODE.SYS is located on drive B of the PDT 3200 unit and is loaded by the CONFIG.SYS file located on that drive. You cannot eliminate the loading of DECODE.SYS except through the safe-boot process.

PM.COM

Use this utility to control various power-management features of the PDT 3200. You can use it at the command line or in a batch file (including AUTOEXEC.BAT) to set or read power-management settings.

Syntax

```
PM [/b#] [/f#] [/h] [/i] [/r] [/t#]
```

Options

/b#

Sets the automatic timeout for the backlight. Replace the # with the number of seconds to wait after a keypress before automatically turning off the backlight. Each keypress restarts the timeout countdown. The range of acceptable values for # is 0 (off) to 255; the default is 15 seconds.

/f#

Sets the time interval for audio indication when the batteries are low. Replace the # with the number of minutes between sounding of the tone. The tone will not sound until the battery has been in the low state for the same number of minutes. The range of acceptable values for # is 0 (off) to 255; this option is off by default.

Note: The audio warning uses extra current, causing the batteries to drain faster. Therefore, you might want to set the interval for as long as possible or not use it at all.

/h

Displays a help file for the utility.

/i

Displays timer settings for auto-off, backlight timeout, and low-battery audio indication.

/r

Resets power management to the cold-boot defaults.

/t#

Sets the auto-off timeout. Replace the # with the number of seconds of nonuse before the unit should transition to its lowest power state. The range of acceptable values for # is 16 to 1032 or 0 for off; the default is 300 seconds (5 minutes).

FLASHDSK.SYS

This driver configures all of the unit's available flash memory as a read/write disk, called a flash disk. Reading from a flash disk is fast, but writing can be somewhat slower. This driver is located on the PDT 3200's drive B and is loaded by the CONFIG.SYS file located on that drive. You cannot eliminate the loading of FLASHDSK.SYS.

LOCK.COM

Use this utility to enable or disable write protection on flash or RAM drives. (This is the software equivalent of the write-protect tab on a floppy disk.) You can use LOCK to protect against accidental erasure of all files stored on the flash disk.

Whenever the PDT 3200 is rebooted, write protection on a drive will be disabled, which is the default state. To lock the drive upon bootup, include a LOCK command in the AUTOEXEC.BAT file on drive C.

Syntax

LOCK [*drive*:] [/l] [/u] [/y]

Options

drive

Specifies the drive letter of the drive which is to be locked or unlocked. If you do not specify a drive letter, the current drive is used.

/l (the letter L)

Lock the flash drive to prevent further disk writes.

/u

Unlocks the flash drive, allowing further disk writes.

/y

Suppresses output from being sent to the display.

Note: If both the /l and /u options are given, the last option specified will take effect.

ORGANIZE.COM

This utility reorganizes sectors on a flash drive to improve flash disk access. Data is relocated so that all free sectors are grouped together, reducing the overhead of freeing space for new data. ORGANIZE.COM has a greater effect on a flash disk as the disk gets closer to being full.

Note: The only PDT 3200 drive you can organize with this utility is the flash disk created with the FLASHDSK.SYS driver. ATA flash drives will not respond properly to this utility.

Syntax

ORGANIZE [*drive*:] [/y]

Options

drive

Specifies the drive letter of the flash drive which is to be reorganized. If a drive letter is not given, the current drive will be used.

/y

Suppresses output from being sent to the display.

FORMAT.COM

Formats a read-write disk drive. Only the flash and RAM drives may be formatted using this command. FORMAT uses the existing drive parameters to recreate the drive, but without any files stored on it. This provides a quick method for cleaning out an existing drive.

Syntax

```
FORMAT [drive:] [/y]
```

Options

drive

Specifies the letter of the drive to be formatted. If you do not specify a drive letter, the current drive is used.

/y

Suppresses output from being sent to the display, and bypasses the usual prompt to proceed with formatting.

XFER.EXE

The XFER utility gives you the ability to transfer files to and from a PC through the PDT 3200's serial port.

Syntax

XFER [*/option1* [*/option2*] ...] *filename*

Options

/1|2

Sets the communication port (COM**1** or COM**2**) to use.

/a

Receives data and appends it to the specified file. If you do not use this option, the file will be overwritten. This option is available only when used with the /p option.

/b#

Specifies the baud rate. Acceptable settings are 2400, 4800, 9600, 19200, 38400, 57600, and 115200; the default is 19200.

/d#

If used with the /p option, this option specifies the receive timeout, in seconds. Replace the # symbol with the desired number of seconds for the timeout.

If used with the /s option, this option specifies the delay between packets, in 55-millisecond intervals. Replace the # symbol with the desired number of delay intervals.

The range of acceptable values for # is 0 to 65,535. If neither /p nor /s is specified, this option is ignored.

/h

Displays a help file for the XFER command.

/o

Overwrites the specified file without prompting. If this option is not used and the filename exists, the following message and prompt will be displayed:
"The file exists. Overwrite? (Y/N)"

/p

Uses Symbol ACK/NAK protocol for the transfer.

/q

Runs the transfer "quietly." If you use this option, only the filename and a "Transmitting..." or "Receiving..." message is displayed. If you do not use this option, additional information will be displayed.

/r

Receives the specified file.

/s

Performs an ASCII dump. This can be specified in transmit mode (/t) only.

/t

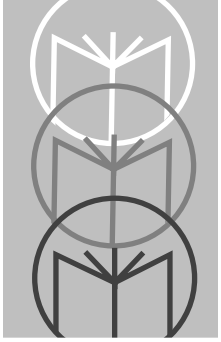
Transmit the specified file. This is the default and does not need to be specified.

/x

Uses Xmodem protocol for the transfer.

VDISK.SYS

This driver configures all of the unit's available extended memory as a read/write disk, called a RAM disk. This driver is located on drive B of the PDT 3200 unit and is loaded by the CONFIG.SYS file located on that drive. You cannot configure the parameters for this driver.



Chapter 3 Resetting the PDT 3200

Chapter Contents

Introduction

One of the critical features of any portable data collection device is its ability to protect against accidental data loss. In the event a PDT 3200 unit becomes “locked up” or otherwise in a state in which data can no longer be processed, you can use any of four methods to reset the unit. This chapter describes these methods, with a focus on the degree of data recovery that is available.

Warm Boot

The warm boot is one of two software methods for resetting a PDT 3200 unit. It is analogous to the CTRL-ALT-DEL key sequence available with IBM-compatible PCs, and the same key sequence is used to force a warm boot on a PDT 3200 unit. This method of resetting a unit should be used first to attempt to bring it back to a usable state.

Since a warm boot can be attempted only from a unit that has been turned on, certain assumptions are made. For example, the rigorous hardware tests that are performed as part of the cold-reset boot sequence are not all necessary; this means the unit restarts faster with a warm boot than with a cold boot.

During a warm boot, data written to the RAM drive (drive D), the flash drive (drive C), or an ATA flash PC card will remain intact. However, if an application is running on a unit before a warm boot, its state cannot be restored.

The CTRL-ALT-DEL key sequence can be disabled, making it difficult for an end user to reset the unit.

Cold Boot

The second software method for resetting a PDT 3200 unit is the cold boot. A cold boot should be used only if a warm boot is unsuccessful. A cold boot is analogous to the results of pressing the Reset button on the front of many desktop PCs or to turning the power off and then on.

To perform a cold boot on a PDT 3200 unit, first turn it off. Then press and hold the ALT and FN 2 buttons while pressing the power button. Release all three buttons simultaneously, and the unit resets.

Simply removing and inserting the battery pack does not force a cold boot, because the backup battery can provide minimal power to sustain the unit's operating state. The exception to this is when the backup battery is drained of all power, a condition that you should not let happen.

When a cold boot is performed, all transient data is lost, along with the state of any application that was running on the unit. Data written to the RAM drive (drive D), the flash drive (drive C), or an ATA flash PC card will remain intact. If a cold boot happens because of a dead backup battery, data written to a RAM drive may not be maintained.

A cold boot begins with a retesting of the hardware and then follows with the DOS boot sequence. The unit reboots DOS under the same conditions you would expect from a desktop PC: Device drivers included in the CONFIG.SYS file and applications included in the AUTOEXEC.BAT file are automatically loaded as part of the boot process. Any other programs that were loaded before the cold boot will not be reloaded automatically.

Safe Boot

During the DOS boot sequence, control is passed sequentially to the CONFIG.SYS files on drives A, B, and C. A similar process is followed for the AUTOEXEC.BAT files on each drive. (See Chapter 1 for more information.) It is possible for a driver or application loading from the CONFIG.SYS or AUTOEXEC.BAT file on drive C to hang the system. A cold or warm boot of the unit will not directly correct the problem, because the troublesome program will be started again as part of the boot sequence.

A method is required by which the CONFIG.SYS or AUTOEXEC.BAT file on drive C and the loading of the CFGDEV.SYS and DECODE.SYS drivers can be taken out of the normal boot sequence. This can be accomplished with the DOS Ctrl-C, F5, and F8 escape sequences.

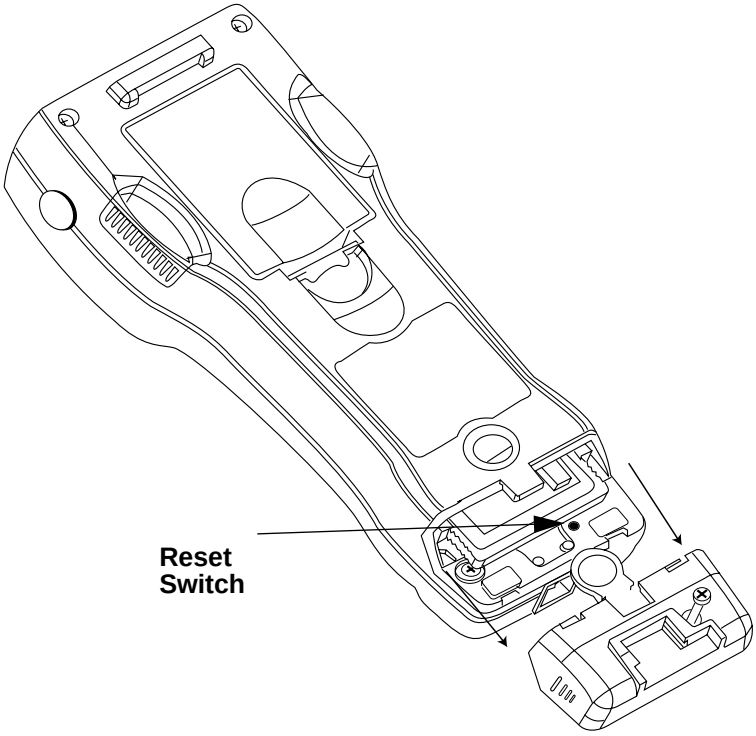
The PDT 3200 provides an alternative method that is less accessible to the end user. To bypass the CONFIG.SYS and AUTOEXEC.BAT files on drive C, reboot the unit (with either a cold or warm boot), and when the message "Wait…" appears on the display, immediately press the ESC and DEL keys at the same time. This causes alternative CONFIG.SYS and AUTOEXEC.BAT processing to take place on drives A and B without chaining into the files on drive C. The unit indicates safe-boot mode by emitting a low tone followed by a higher-pitched tone. The flash drive (drive C) and the RAM drive (drive D) will be preserved, and the boot sequence will place the unit at the C:\> prompt.

Note: If you do not hear the two tones indicating the safe-boot sequence, you probably did not press the ESC and DEL keys quickly enough. Reboot the PDT 3200 and press the two keys simultaneously as soon as you see the "Wait..." message on the display.

Hardware Reset

In the extremely rare situation where none of the rebooting methods is successful, you can use a reset mechanism that is located under the PC card slot cover (see Figure 3-1). Touch a metal device (such as a paper clip) to the two contacts on the reset mechanism. This will cause the PDT 3200 to begin a cold boot.

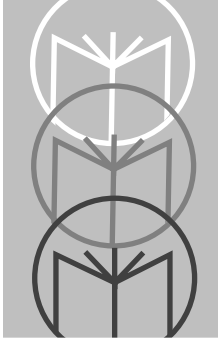
Figure 3-1. Location of the Hardware Reset Mechanism



Recommended Programming Practices

The programming methodologies used to develop DOS applications for a desktop PC also apply to creating applications for the PDT 3200. Some of these are more critical to success on a PDT 3200 than on a PC, because of the hardware environment of the PDT 3200. The following are some important guidelines for programming PDT 3200 applications:

- *Always check for reported error conditions.* Many of the BIOS and DOS service routines are useful only if the calling application checks for any flagged error conditions that signal a failure to comply with the original request. For example, not enough available memory or lack of disk storage space must be dealt with immediately; otherwise data may be lost.
- *Carefully choose how data will be stored.* A PDT 3200 unit has a number of data-storage options: RAM drive, flash drive, ATA flash PC card, remote storage via RF, and other PC card possibilities. You should be aware of the strengths and limitations of each storage method and plan to store critical and temporary data in appropriate locations so as to maximize both system performance and data security.
- *Disable the CTRL-ALT-DEL key combination.* The PDT 3200 allows for easy rebooting during the development process. To prevent the end user from resetting the unit in the field, disable the CTRL-ALT-DEL key combination. You can accomplish this by using bar code programming (see the *PDT 3200 Product Reference Guide*) or by calling an API function (see the Developer's Toolkit).



Chapter 4 Using PC Cards

Chapter Contents

Introduction

PC card support expands the feature set of the PDT 3200 to include:

- Supplemental storage capacity through ATA flash cards
- Modem support
- LAN connectivity
- Radio frequency (RF) communications

PDT 3200 supports Type 1 and Type 2 PC cards. Only one card of either type can be installed in the unit at any one time.

This chapter describes the use of PC cards with the unit. It does not provide details about specific cards, such as communications protocols for fax/modem cards. This type of information should be available from the PC card vendor.

For information about opening the PC card slot cover, inserting and removing PC cards, and card recognition, see the *PDT 3200 Product Reference Guide.*

Overview

The primary intended uses for PC cards with the PDT 3200 are expanded storage space and radio frequency (RF) communications. Using ATA flash memory cards enables you to configure a PDT 3200 for a theoretical maximum of 64MB of memory. The card appears as a hard drive to the user. RF communications cards allow you to connect the portable PDT 3200 unit to a traditional network and to access shared network resources in a transparent fashion. The PDT 3200 can also use network and fax/modem cards.

Use the PDT 3200 Configuration Utility to download the appropriate drivers for ATA and generic I/O cards.

Configuring Your PC Card

PhoenixCARD Manager Plus (PCM+)

To attain compatibility with as many PC cards as possible, the PDT 3200 utilizes card and socket services supplied by Phoenix Technologies. These card and socket services are implemented as installable device drivers and are loaded using the DEVICE= command in the CONFIG.SYS file.

Table 4-1 identifies the PhoenixCARD Manager Plus (PCM+) drivers needed for each type of card. Information about these drivers follows the table. Additional drivers or application programs may be required for some cards, particularly RF, LAN, and fax/modem cards.

Table 4-1. PCM+ Drivers Required for PC Cards

PCM+ Driver	ATA	Fax/ Modem	LAN	RF
CNFIGNAM.EXE	X	X	X	X
PCM.INI	X	X	X	X
PCMATA.SYS	X			
PCMCS.EXE	X	X	X	X
PCMSCD.EXE		X	See Below	
PCMSS.EXE	X	X	X	X

- **CNFIGNAM.EXE** identifies the section in the PCM+ configuration file (PCM.INI) that should be loaded as the default.
- **PCM.INI** is the configuration file for the PC cards.
- **PCMATA.SYS** is an ATA-IDE device driver. It communicates with the PCMCS.EXE driver to allow the system to access ATA-configured PC cards.
- **PCMCS.EXE** is the card-services driver. Required for all PC cards, it provides an interface between applications and the socket services. It also manages resource allocations for the PC cards.
- **PCMSCD.EXE** is referred to as the super client driver. It uses the PCM.INI file to configure and request system resources for generic PC cards as well as specific cards that have been included in the PCM.INI file. Some PC cards will have

specific client drivers to handle configuration; PCMSCD.EXE should not be used with those cards.

- **PCMSS.EXE** is the socket-services driver. Required for all PC cards, it provides an interface to the PC card hardware.

ATA Cards

ATA cards are flash memory cards that utilize the IDE interface. This allows the flash to appear as a hard drive. Any ATA PC card should work on the PDT 3200 unit with the appropriate drivers installed. The following ATA cards are approved for use on the PDT 3200:

- 2.6MB SunDisk Seagate FlashCard, Model ST72P5
- 4MB SanDisk PCMCIA PC CARD ATA
- 8MB SanDisk PCMCIA PC CARD ATA

Additional cards may be approved. Contact your Symbol Technologies representative for information.

Fax/Modem Cards

The PDT 3200 should recognize any generic fax/modem PC card. Fax/modem PC cards are configured to use COM2, I/O address 2F8h, and IRQ 3. The following fax/modem cards are approved for use on the PDT 3200:

- US Robotics Sportster 28.8 Faxmodem PC card with DataView Connector
- Xircom CreditCard Modem 28.8

Note: The PDT 3200 does not support fax/modem cards with XJack connectors.

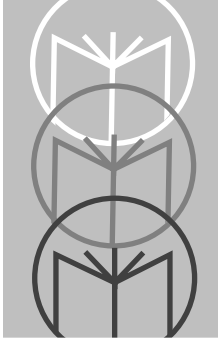
LAN Cards

LAN cards require assignment of an I/O window, a memory window, and an IRQ. If the LAN card will be configured by the super client driver (PCMSCD.EXE), additional entries to the PCM.INI file will be necessary. Many LAN cards come with their own resource-management system and do not require the PCMSCD.EXE driver.

Note: Because of the PDT 3200's memory-handling properties, some LAN cards may not work. Please contact Symbol Technologies before purchasing a LAN card.

RF Cards

Because of the special requirements of RF environments, Symbol Technologies is distributing PDT 3200 RF units to qualified customers only. Please contact your representative to see if you are qualified.



Chapter 5 ROM-DOS Commands

Chapter Contents

Introduction

This chapter lists the Datalight ROM-DOS commands that are available in the PDT 3200. ROM-DOS commands that differ from their MS-DOS equivalents are identified and described.

The information in this chapter is taken from Datalight's *ROM-DOS 6.22 User's Guide* and is used by permission.

Command Overview

Table 5-1 lists the Datalight ROM-DOS commands that are provided with the PDT 3200 and gives a brief description of each command.

Table 5-1. DOS Commands Used with the PDT 3200

Command	Description
?	CONFIG.SYS command. Directs ROM-DOS to pause for confirmation before processing a command.
@	Suppresses the display of a single batch file command line.
;	Same as REM. Identifies nonexecuting lines.
ATTRIB	Displays or modifies the attributes associated with a file.
BREAK	CONFIG.SYS command. Enables or disables the ability to stop program execution at a non-I/O point.
BUFFERS	CONFIG.SYS command. Sets the number of internal data buffers.
CALL	Batch subcommand. Invokes execution of a secondary batch file.
CHDIR	CHange DIRectory (also CD). Changes the default directory.
CHKDSK	CHecK DiSK. Checks integrity of data on a disk. Displays information.
CLS	CLear Screen. Clears all information from the viewport.
COMMAND	Spawns a second DOS command processor.
COPY	Copies files from one storage location to another.
CTTY	Change TeleTYpe. Changes the default terminal interacting with ROM-DOS.

Table 5-1. DOS Commands Used with the PDT 3200 (Continued)

Command	Description
DATE	Displays the date from the system's internal calendar and allows revision.
DEL	DELete. Deletes specified files.
DEVICE	CONFIG.SYS command. Installs a device driver into ROM-DOS.
DOS	CONFIG.SYS command. Installs ROM-DOS into high-memory area (HMA).
ECHO	Batch subcommand. Turns on the display of batch execution on the viewport.
ERASE	Same as DEL. Erases specified files.
EXIT	Exits “nested” running of ROM-DOS within another program.
FCBS	CONFIG.SYS command. Specifies the number of file control blocks (FCBS) open at one time.
FILES	CONFIG.SYS command. Sets the maximum number of files that can be open at one time on the system.
FIND	Works as a filter to display only lines that contain a specified string.
FOR	Batch subcommand. Performs one DOS command on a set of files.
GOTO	Batch subcommand. Moves control to a specified line in the batch file.
HELP	Lists all available ROM-DOS commands along with brief descriptions.
IF	Batch subcommand. Performs a command based on a specified condition.

Table 5-1. DOS Commands Used with the PDT 3200 (Continued)

Command	Description
INCLUDE	CONFIG.SYS command. Allows instructions in one configuration block to be included with instructions in another configuration block.
INSTALL	Loads terminate-and-stay resident (TSR) programs during CONFIG.SYS processing.
LASTDRIVE	Sets the maximum number of drives.
MENUDEFAULT	Sets the default menu item choice and time-out value for making a selection.
MENUITEM	Specifies an item to be placed in the startup menu displayed during system boot.
MKDIR	MaKe DIRectory (also MD). Creates a new subdirectory.
NEWFILE	Allows continuation of CONFIG.SYS processing from a new file.
PATH	Displays current command search path(s). A new path line can be specified.
PAUSE	Batch subcommand. Causes execution to halt until a key is pressed.
PRINT	Prints a list of files, up to ten files.
PROMPT	Resets the appearance of the system prompt line.
REM	REMark. A batch subcommand for identifying nonexecuting lines.
REN	REName. Renames files.
RMDIR	ReMove DIRectory (also RM). Deletes a specified subdirectory.
SET	Sets environment variables and command processor strings.
SHARE	Installs the capabilities for file sharing and file locking on your hard disk.

Table 5-1. DOS Commands Used with the PDT 3200 (Continued)

Command	Description
SHELL	Allows selections of an alternate boot program other than the default COMMAND.COM command processor.
SHIFT	Batch subcommand. Shifts replaceable parameters one position "to the left."
SORT	Sorts a text file and displays the output to the standard device.
STACKS	Allows for the use of dynamic data stacks to handle interrupts. Note: ROM-DOS does not utilize this command.
SUBMENU	Defines a menu item that represents a secondary menu.
SWITCHES	Allows special CONFIG.SYS file options.
TIME	Displays the current time from the system's internal clock. Allows revision.
TREE	Displays the path of each directory on a specified drive.
TYPE	Displays the contents of a text file on the viewport.
VER	Displays current version of ROM-DOS on the viewport.
VERIFY	Displays the current VERIFY state or sets the VERIFY state to on or off.
VOL	Displays the volume label on a disk.
XCOPY	Copies multiple files and, optionally, subdirectories.
XDEL	Deletes files and subdirectories including empty subdirectories.

ROM-DOS vs. MS-DOS

Table 5-2 identifies differences between ROM-DOS commands and their MS-DOS equivalents. For more information, see the individual command descriptions in the next section.

Table 5-2. Difference Between ROM-DOS and MS-DOS Commands

Command	Features Exclusive to ROM-DOS	Features Exclusive to MS-DOS
ATTRIB	-C: Clears all file attributes	/S: Processes files in all subdirectories
BUFFERS		Allows designation of secondary buffer cache BUFFERS n,m where m is secondary buffer cache
CHKDSK	/C: Corrects errors without user confirmation	
COMMAND	Internal command help only	/Y: Steps through batch file specified by /C or /K switch HELP.HLP in MS-DOS includes all commands (external and internal), installable drivers, etc.
DIR	/A X: Shows attributes	/O:C and /O:-C: Sorts by compression ratio /C[H]: Displays compression ratio of files compressed using Drivespace or Doublespace
FCBS	, [minimum number]	
FIND		/I: Not case-sensitive
HELP	Requires COMMAND.HLP	MS-DOS requires external file, HELP.COM, but provides more help text
NEWFILE	ALL	Not available with MS-DOS

Table 5-2. Difference Between ROM-DOS and MS-DOS Commands (Continued)

Command	Features Exclusive to ROM-DOS	Features Exclusive to MS-DOS
PRINT	/F: Sets maximum number of files	/D:LPT*x* (PDT 3200 has no parallel port.) /U /M /S /Q
SHARE	/U: Unloads SHARE.EXE	/F:space: Allocates space to record
SWITCHES		/W
TREE		/A: Indicates no graphics characters for tree symbols
VER	/R: Shows full version and release number Allows revision of the version number	
XCOPY		/Y /-Y: Turns confirmation prompts on or off
XDEL	All	Not available with MS-DOS Similar to DELTREE

Command Descriptions

This section provides descriptions of the ROM-DOS commands listed in Table 5-2. Each entry includes an explanation of the command's purpose, the command entry syntax, remarks, and examples.

Each command also has a label to designate whether it is an internal or external command. Internal commands are part of the command processor program, COMMAND.COM. These functions are available only while COMMAND.COM is running. External commands are actually stand-alone utility programs. They are independent from COMMAND.COM. Internal commands that are unique to CONFIG.SYS processing are also identified. These commands can be used only in a CONFIG.SYS file.

Command ?

Type Internal

Purpose

The question mark (?) command is placed on a command line in the CONFIG.SYS file following the actual command. It directs ROM-DOS to pause and ask for confirmation before processing the command.

Syntax

[*command*]? = *command_arguments*

Remarks

The *command* can be any of the following standard CONFIG.SYS commands:

BREAK=
DEVICE=
DOS=
FILES=
STACKS=
BUFFERS=
FCBS=
INSTALL=
LASTDRIVE=
SWITCHES=

The *command_argument* can be any of the available options defined for the command. Please refer to the individual command description for complete instructions.

The question mark (?) should be placed just before the equals sign (=) in the command line.

Examples

```
DEVICE?=VDISK.SYS 64 /E
```

The above example causes ROM-DOS to pause and ask for confirmation before installing the VDISK. If Yes (y) is answered, the installation will continue. If No (n) is answered, the device will not be loaded and processing will move on to the external CONFIG.SYS command line.

```
DEVICE?=VDISK.SYS 64 /E
```

The above example causes ROM-DOS to pause and ask for confirmation before installing the VDISK. If Yes (y) is answered, the installation will continue. If No (n) is answered, the device will not be loaded and processing will move on to the External CONFIG.SYS command line.

Command @

Type Internal

Purpose

The AT sign (@) is used to prevent a single command in a batch file from being echoed to the screen as the batch file is being run. The @ sign is placed in front of the command whose display is to be suppressed.

Syntax

@[*batch file command*]

Remarks

The batch file command argument can be any executable line in your batch file.

Examples

In the following example, the COPY instruction will be executed but the instructions will not be echoed to the screen as the batch file runs.

```
@COPY FILE1.BAT FILE1.SAV
```

The ECHO OFF command differs from the @ sign in that it cause all commands following it to *not* be displayed on the screen. To prevent the ECHO OFF command from displaying itself, place the @ sign in front of the command.

```
@ECHO OFF
```

Command ;

Type Internal

Purpose

The semicolon (;) command has two purposes: to allow comments in a batch or CONFIG.SYS file, and to temporarily disable a command without physically deleting the command from the file. See also the REM command.

Syntax

```
;any text here
```

Remarks

The ; command is used to functionally remove a command from the CONFIG.SYS file without actually deleting it from the CONFIG.SYS file.

Examples

```
;C:\BIN\VDISK.SYS 64 /E
```

The above example causes the VDISK command to not be executed until the ; command is removed.

Command ATTRIB

Type External

Purpose

The ATTRIB command either displays or modifies the attribute of a file.

Syntax

ATTRIB [+ ≠ -][*option*][*filespec*]

Remarks

The file attributes define the characteristics of a file. They determine if a file may be deleted or modified, or if it is archived. The ATTRIB command is used to manage these file attributes.

Wildcard characters may be used in the ATTRIB filespec.

The ATTRIB command will modify file attributes if modify commands are given to ATTRIB. The modify commands are:

Option	Description
+/-	Add (+) or remove (-) attribute
A	Archive attribute
C	Clear all attributes
H	Hidden File attribute
R	Read Only attribute
S	System File attribute

If no modify commands are found by ATTRIB, then the files are displayed along with the file names and their current attributes.

Examples

ATTRIB will add the Read Only attribute to the file myfile.dat.

```
ATTRIB +r myfile.dat
```

ATTRIB will remove the Read Only attribute and the Archive attribute of all files with the DAT extension.

```
ATTRIB -a -r *.dat
```

ATTRIB will display the attributes of all files with the DAT extension.

```
ATTRIB *.dat
```

Command BREAK

Type Internal

Purpose

Expands the list of operations that can be stopped by pressing <Ctrl><C> or <Ctrl><Break>. Alternatively, returns to the default setting of a limited number of "break-able" operations.

Syntax

```
BREAK [ON≠OFF]
```

Remarks

In the normal default condition, the BREAK switch is off. In the off mode, the "stop" commands, <Ctrl><C> and <Ctrl><Break>, will affect activities that read from or write to the keypad, the screen, or the printer. ROM-DOS will not look for these stop commands during any other activities.

With the BREAK switch set to ON, ROM-DOS looks for <Ctrl><C> and <Ctrl><Break> during other activities such as disk reads and writes.

Examples

```
BREAK ON
```

Expands the BREAK list.

```
BREAK OFF
```

Returns to limited BREAK list.

```
BREAK
```

Displays the current BREAK setting.

Command BUFFERS

Type CONFIG.SYS

Purpose

ROM-DOS has internal buffers to temporarily hold data read from the disk. Increasing the number of internal buffers will speed system performance.

Syntax

BUFFERS = *number*

Remarks

Each buffer used by ROM-DOS requires 512 bytes of RAM. The BUFFERS command will increase or decrease the amount of RAM used by the operating system.

The minimum number of buffers is 2 and the maximum number is 40. If a number less than 2 is given then the number of BUFFERS is set to 2. If a number larger than 40 is given then BUFFERS is set to 40.

Example

The following example causes ROM-DOS to have 10 buffers. These 10 buffers will use 5120 bytes of RAM.

```
BUFFERS = 10
```

Command CALL

Type

Internal Batch Subcommand

Purpose

CALL invokes execution of a secondary batch file without exiting the primary batch file. When the secondary batch file is done executing, control is returned to the primary batch file.

Syntax

CALL *batchfile* [*batchfile arguments*]

Remarks

Parameters for the secondary batch file may also be included, if appropriate.

Examples

```
CALL BATCH2
```

This command executes the batch file BATCH2.BAT.

```
CALL MYBATCH FILEX FILEZ
```

This command executes the batch file MYBATCH.BAT. The arguments passed to MYBATCH.BAT are:

```
%1 = FILEX

%2 = FILEZ
```

Command CHDIR (CHange DIRectory)

Type Internal

Purpose

Changes the default directory.

Syntax

CHDIR [*d:*][*path*]*subdir*

CD [*d:*][*path*]*subdir*

Remarks

Subdir is the name of the new default subdirectory. Note that CD may be used instead of CHDIR.

The new directory which is to become the default must already exist. See MKDIR for creation of subdirectories.

A series of two periods (..) can be used to indicate a move back to the parent directory.

Specifying only the backslash (\) for the *subdir* argument will move you to the root directory of the current default drive.

Examples

```
CHDIR \TOOLS
```

Entering of this command will move you into the subdirectory called TOOLS, whose parent directory is the root of the current default drive.

```
CD A:
```

Entering of this command will display the current directory on drive A:. Any valid drive letter can be substituted to get the current directory on that drive.

The following examples will all use this directory tree structure:

```
CD D:\TEST\NEW
```

This command will move you into the subdirectory called NEW, located on the D: drive, under the parent directory TEST.

```
CHDIR ..
```

This command will move you back to the parent directory of the current subdirectory. If you were in the directory D:\TEST\NEW (from the previous example), this CHDIR command would move you from NEW back into the TEST directory.

```
CD ..\WORDPROC
```

This command will move you back to the parent directory and then into a subdirectory called WORDPROC. If we started in the TEST directory, we would move back to the ROOT directory and then into the WORDPROC subdirectory.

```
CD \
```

This command will move you back to the root directory, from any starting point in the directory tree.

Command CHKDSK (CHecK DiSK)

Type External

Purpose

The CHKDSK command checks the disk directories and File Allocation Table (FAT) and displays a disk and memory report.

Syntax

CHKDSK [*d:*][*path*][*filespec*][/C] [/F] [/V]

Remarks

CHKDSK examines a disk and determines if the disk has any errors in the File Allocation Table (FAT) and will optionally fix errors.

Options

The /F option causes CHKDSK to fix errors on the disk if any were found. The errors that can be found are directory or FAT errors. If the /F is not specified then CHKDSK acts as if it will fix the disk, but the corrections will not be written out to the disk.

If errors are detected, you will be prompted with a message similar to the following:

15 lost allocation units found in 5 chains.

Convert lost chains to files?

If you answer Y for Yes, each lost chain will be written to a file in the root directory of the current default drive. Each file will have the name FILE*nnnn*.CHK. *nnnn* will be a sequential number. The first chain will be in FILE000.CHK. These files can be verified to see if they contain valuable information, and then deleted if desired.

Answering N for No to the above prompt, CHKDSK will still make the corrections however the lost chains will not be saved to the disk.

The /C option allows CHKDSK to correct errors without user confirmation. This option must be used along with the /F option for corrections to be made.

The /V option causes CHKDSK to display each path and file as it is processed.

If a file specification is specified, then CHKDSK displays all files matching the specification that have non-contiguous data areas on the disk. Files that are stored in non-contiguous areas, especially .EXE files, have slower disk access times. If CHKDSK reports a large number of files with this problem, a utility program that optimizes the files and free space on your disk should be used.

After checking the disk, CHKDSK displays any error messages followed by a report on the state of the disk that was checked. An example of the report is shown below.

Volume ROM-DOS created June 1,1990 1:00a

Volume Serial Number is 190E-4AA2

```
362496 bytes total disk space
     0 bytes in 1 hidden files
  6144 bytes in 2 user files
356352 bytes available on disk
655360 bytes total memory
595360 bytes free
```

CHKDSK does not wait for a disk to be inserted before the checking is initiated nor does it repair any errors.

Examples

CHKDSK will check the integrity of drive A. The report will be printed to the console.

```
CHKDSK a:
```

CHKDSK will check the integrity of RAM disk D. The report will be saved in a file called DRIVE_D.RPT.

```
CHKDSK d: >drive_d.rpt
```

Command CLS (CLear Screen)

Type Internal

Purpose

Clears the viewport to display a blank screen.

Syntax

CLS

Remarks

CLS clears the screen except for the ROM-DOS prompt and the cursor in the upper left-hand corner. There are no additional options for CLS.

Command COMMAND (Command Processor)

Type External

Purpose

Start a new Command Processor.

Syntax

COMMAND[*device*][/E:*number*][/K:*filename*][/P][/C *string*][/*MSG*]

Remarks

This command starts a new copy of the ROM-DOS Command Processor. The Command Processor is the program that has all the internal DOS commands in it.

Starting a new Command Processor will also produce a new environment. The size of the environment is 128 bytes by default, but can be changed using the /E switch.

Command and its arguments can also be used in a SHELL= statement in your CONFIG.SYS file. See the full description of SHELL for more details.

Options

The *device* option specifies that COMMAND.COM should use a different device, such as AUX, for input and output.

The /*E:number* switch sets the environment size. Number represents the size of the environment in bytes. Number must be in the range from 160 to 32768. All other values will be ignored and the default value of 256 will be used. ROM-DOS will round the value entered up to the nearest multiple of 16.

The /*K:filename* option tells the command processor to run the specified filename and then display the ROM-DOS command prompt. It is not recommended that this option be used in a CONFIG.SYS SHELL= statement.

The /*P* switch causes COMMAND not to exit, in other words, remain permanent. The /P switch should only be used when command is used in a CONFIG.SYS SHELL statement.

The /*c* string switch causes COMMAND to execute the command in string and then terminate. The string command can be any internal or external command.

The /*MSG* option indicates that all error messages should be stored in memory. This option is recommended only for diskette based systems. ROM-DOS keeps

many of its error messages in the resident part of COMMAND.COM rather than using valuable memory to store them. If an error message is needed and you have loaded ROM-DOS from a diskette, the message will only be available if the boot disk is still in the drive. By using the /MSG option, the messages will be available in memory at all times. The /P option must be used along with the /MSG option.

Examples

The following command will cause a new copy of COMMAND to be executed. It will perform a DIR command on the C drive and then exit back to the previous Command Processor.

```
COMMAND /C DIR C:
```

The following example shows loading of a permanent copy of command with an environment size of 256 bytes.

```
SHELL=C:\COMMAND.COM /P /E:256
```

Command COPY

Type Internal

Purpose

Copies a file or set of files to a specified destination which could be on another disk, or on another subdirectory on the current disk, or on a completely different drive. COPY may also be used to alter the *filename* within the current directory. In addition, this command can be used to direct communication between files and devices (e.g., file contents to a printer, keypad input to a file).

Syntax

This command has several possible formats. The essential structure of each command is:

COPY *source target option*

The source is the "copy from" filespec or device, and the target is the "copy to" filespec or device. Following are various configurations of the COPY command format.

**COPY [*d:*][*path*]*filename*[*/option*] [*d:*][*path*]*filename*
[*/option*]**

Where the first filespec indicates the source file(s) to be copied and the second filespec indicates the target area on which to copy.

**COPY [*d:*][*path*]*filename*[*/option*] + [*d:*][*path*]*filename*
[*/option*] + ... [*d:*][*path*]*filename*[*/option*]**

As shown, several source filespecs may be listed to be copied into the target filespec which is listed last. The source files will be concatenated one after the other into the target file.

COPY [*d:*][*path*]*filename*[*/option*] *device*

The target device is a console or printer (PRN).

COPY *device* [*d:*][*path*]*filename*[*/option*]

The device is the source such as a keyboard or console, the output of which will be directed to the target filespec.

Options

The /A and /B options stand for ASCII and binary, respectively. They act as switches that allow each of these file types to be copied. When one of them is used, it applies the preceding filename, and remains in effect for any filenames following in the command line until superseded by another /A or /B.

Most of the time the /A and /B options are not necessary. They are only needed when you are combining ASCII and binary files.

/A

Indicates that the file is to be treated as an ASCII file (text file). When used with the source file, everything will be copied up to, but not including, the first <CTRL><Z> end-of-file marker. When /A is used on the target file, a <Ctrl><Z> will be added as the last character in the file.

/B

Indicates that the file is to be treated as a binary file. When /B is used with the source file, the entire file is copied regardless of any <Ctrl><Z> characters. When /B is used with the target filespec, no <Ctrl><Z> end-of-file mark is added.

/V

This is the Verify option. It tells COPY to compare what is written to the target file with the contents of the source file. If discrepancies are discovered, an error message is displayed. This option slows the execution of the COPY command. Since errors in copying are quite rare, this option is recommended as a safeguard only when copying critical data.

/Y

This option indicates that you want to copy the current file(s) over the existing file(s) of the same name(s) without confirmation. Using this option overrides the setting made by the COPYCMD environment variable.

/-Y

This option indicates that you want to confirm the copy of one file over the existing file of the same name. Using this option will override the setting made by the COPYCMD environment variable.

The COPYCMD environment variable can be set using the SET command. This allows you to set confirmation on or off for the COPY command. If you always want to be prompted for confirmation when a file will copy over an existing file, set COPYCMD= /-Y. To automatically overwrite without confirmation during a copy instruction, set COPYCMD= /Y. Refer to the SET command for proper usage.

Remarks

When no filename is specified for the target, the new copy will be given the same name as the source file name.

When no drive or path is specified for the source, the current drive and directory is assumed. When no drive or path is specified for the target, the current drive and directory are assumed.

If a drivename only is specified without a path, the default directory for that drive is assumed.

Both source and target filespecs may include wildcard characters (* and ?) to specify a set of several files.

Examples

```
COPY LETTER.TXT A:
```

This command will copy the file LETTER.TXT (in your current default drive and path) to the default directory on the disk in drive A.

```
COPY *.DOC A:
```

Copies all files in the default directory with an extension of .DOC to the default directory of drive A.

```
COPY DATAORIG.DOC DATABACK.DOC/V
```

This example creates a backup copy, DATABACK.DOC, from the file DATAORIG.DOC, then verifies that the file was correctly copied. The new file will be located in the current default directory.

```
COPY JAN.DAT + FEB.DAT + MAR.DAT QTR1.DAT
```

Copies the files JAN.DAT, FEB.DAT, and MAR.DAT in sequence into the single file, QTR1.DAT.

```
COPY CON NEWFILE.TXT
```

Sets up your console (keypad) to input directly to NEWFILE.TXT. <Ctrl><Z> followed by <Enter> closes the file and returns to normal command line operation.

Command COUNTRY

Type CONFIG.SYS

Purpose

ROM-DOS supports multiple country formats for time, date, and currency, as well as other basic country-specific information. A country is identified by a three-digit international telephone country code.

Syntax

COUNTRY = *countrynumber* [*codepage*]

Remarks

The file COUNTRY.SYS must be present in the same directory as your CONFIG.SYS file.

If you do not specify a code page, ROM-DOS will use the default code page for the chosen country. If a code page is specified, it must be either the default or alternate code page for the chosen country.

The ROM-DOS DATE and TIME commands are affected by this command. Applications that use DOS functions to determine the date, time, or currency format, or request that DOS provide character sort order, or uppercase information, will be affected as well.

The following table shows the currently supported countries:

Country	**Code**	**Code Page**	**Alternative Code Page**
Australia	061	437	850
Belgium	032	850	437
Brazil	055	850	437
Canadian-French	002	863	850
Czech Republic	042	852	850
Denmark	045	850	437
Finland	358	850	437
France	033	850	437
Germany	049	850	437
Hungary	036	852	850
Italy	039	850	437
Japan	081	932	---
Latin America	003	850	437
Netherlands	031	850	437
Norway	047	850	865
Poland	048	852	850
Portugal	351	850	860
Spain	034	850	437
Sweden	046	437	850
Switzerland	041	850	437
United Kingdom	044	437	850
United States	001	437	850
Yugoslavia	038	852	850

For more information on COUNTRY, see the section on Configuring ROM-DOS for International Use. The default country code is 001, for the U.S.A.

Examples

```
COUNTRY= 049

COUNTRY = 049, 437
```

The next time you start ROM-DOS with either of these COUNTRY commands, the date and time will be displayed as follows:

```
DATE

Current date is Wed 20.06.1990

Enter new date (dd.mm.yy):

TIME

Current time is 16:39:54,45

Enter new time:
```

The first COUNTRY command above would use code page 850, by default, for sorting and case conversion. The second COUNTRY command example would use the specified code page 437 instead.

Command CTTY (Change TeleTYpe)

Type

Internal

Purpose

Allows you to direct input and output to a device other than your computer's standard keypad and viewport.

Syntax

CTTY *device*

Remarks

This command would be used in any situation requiring interaction with an alternate console.

Caution

The CTTY command only affects communication with ROM-DOS and with programs that work through ROM-DOS for input and output. For example, if you go into BASIC it will use standard keypad input regardless of previous CTTY command usage.

Examples

```
CTTY COM2
```

Sets the device on COM2 as the input/output device.

```
CTTY CON
```

Returns control to the standard keypad.

Command DATE

Type Internal

Purpose

Displays the current date (month, day, year) as known to ROM-DOS and also allows you opportunity to change it.

Syntax

DATE [*mm-dd-yy*]

Remarks

The date set by this command is used, among other things, for "date stamping" your file-revision dates. This information is displayed when you execute a directory listing of your files.

You may want to include the DATE command in your AUTOEXEC.BAT file, to set the date at boot-up. If your computer has an internal battery-operated clock, you won't need to do so.

The format of the date command is also dependent on the Country specified in CONFIG.SYS. The date is displayed according to local standards for the specified country.

Also see the TIME command.

Examples

If the command is entered without the specification of *mm-dd-yy*, the current date as known to ROM-DOS will be displayed and you will be prompted to enter a new date. Like this:

```
Current date is Sat 6-10-1989

Enter new date (mm-dd-yy):
```

If you do not want to change the date, just press enter. Otherwise, key in the current date and press enter.

Alternatively, you may skip the display and prompting by simply entering the current date on the command line. To enter June 10, 1997, type the DATE command as follows (the century number 19 is assumed):

```
DATE 6-10-97
```

Valid entries for months, days, and years are:

mm **= 1-12** ***dd*** **= 1-31** ***yy*** **= 80-99**

(for years with 19 as the century).

ROM-DOS will figure out the day of the week; do not include it in your entry.

The earliest year that ROM-DOS can accept is 1980. ROM-DOS assumes that the century is 19 unless something different is entered, such as 20, next to the year. For example, February 18, 2014 would be:

```
DATE 02-18-2014
```

ROM-DOS will adopt your entry as the current date.

Command DEL (DELete)

Type Internal

Purpose

Deletes a specified file or set of files.

Syntax

```
DEL [d:][path]filename [/P]
```

Remarks

The DEL command and the ERASE command are functionally identical.

When no drive is specified, the default is assumed. When no path is specified, the default path is assumed.

Global filename characters ? and * (wildcards) can be used in the filespec. This should be done with caution as it is possible to delete multiple files unintentionally.

When the filespec *.* is used to delete all files in the specified subdirectory, a verification message will be displayed:

```
Are you sure (Y/N) ?
```

Type Y only if you want all files in the specified subdirectory to be deleted.

Caution

There is no ROM-DOS command to "undelete" a file. Although utilities exist which can attempt an undelete, certain factors can cause the deleted file to be partially or totally lost. The DEL command should be treated as a permanent delete.

DEL deletes files within a subdirectory, not the subdirectory itself. For subdirectory removal, see the RMDIR command.

Options

/P

The /P option causes DEL to pause and prompt the user before each file is deleted. This option is most useful when deleting files with wildcards. A sample prompt is shown below:

***Filename*, Delete (Y/N) ?**

Examples

This example deletes all files on the A: drive with a .DOC extension. Before each file is deleted the user is prompted to determine if that file should be deleted.

```
DEL A:*.DOC /P
```

Deletes the file MYLETTER.DOC from the current default subdirectory.

```
DEL MYLETTER.DOC
```

Deletes all files in the current subdirectory with a .DOC file extension.

```
DEL *.DOC
```

Command DEVICE

Type CONFIG.SYS

Purpose

A device driver may be installed into ROM-DOS via the DEVICE command.

Syntax

DEVICE = *device_driver arguments*

Remarks

A device driver is used to allow ROM-DOS to access hardware that is not common in all PCs.

The full drive path and file name of the device must be specified. The arguments are different depending on the device driver.

Examples

```
DEVICE=C\BIN\VDISK.SYS 120 /e
```

This example installs the ROM-DOS RAM disk driver, VDISK.SYS, via the DEVICE command. There will be 120K bytes of extended memory dedicated to the RAM disk in this example.

Command DEVICEHIGH

Type

CONFIG.SYS

Purpose

Loads an installable device driver into the upper memory area if available.

Syntax

DEVICEHIGH = *device_driver arguments*

Remarks

A device driver is used to allow ROM-DOS to access hardware that is not common in all PCs. A device can be loaded into the upper memory areas if they are available and there is enough free upper memory to accommodate the drivers needs. To make high memory available, the EMM386.EXE and HIMEM.SYS utilities must be loaded. If these utilities are not loaded or there is not enough upper memory available, the device will load into conventional memory.

The full drive path and file name of the device must be specified. The arguments are different depending on the device driver.

Example

The following example installs a driver called MYDEVICE with its command line arguments as specified. The device will load into upper memory if available.

```
DEVICEHIGH=C:\BIN\MYDEVICE.SYS /20 /M
```

Command DIR (DIRectory)

Type Internal

Purpose

Displays a list of the files that are in a directory.

Syntax

DIR [*d:*][*path*][*filename*][*/option*]

Remarks

The DIR command can be used to list all the files in a directory, or to show the directory entries of specific files. The standard directory display format includes columns for filenames, filename extensions, file sizes, and the dates and times the files were created.

Options

/A *attributes*

The /A option causes the DIR command to display only the files that match the specified filespec and have the given attribute. The table below shows the legal attribute descriptions.

Letter	Description
A	Archive
D	Directories
H	Hidden Files
R	Read Only Files
S	System Files
X	Show Attributes

-

The dash (-) symbol can be used to negate listed attributes. For example, to select all files that do not have the archive bit set, use the /A -A option.

/B

The /B, or bare, option causes the display to be displayed without volume label, date, time, or size information.

/L

The /L option causes the filenames to be displayed in lowercase.

/P

The /P option selects page mode, which makes ROM-DOS pause the display each time the screen is full. Press any key to go on to the next page of entries.

/O

The /O option causes the filenames to be displayed in sorted order. The sort order can contain one or more of the following letters:

Letter	Description
D	By date and time, newest first
E	Alphabetic order by extension
G	Directories grouped before files
N	Alphabetic order by name
S	Size, smallest first

-

The dash (-) symbol can precede the sort option to reverse the sort order. For example, to sort all files in the directory in reverse alphabetic order, use the /O-N option.

/S

The /S option causes the display to include files in subdirectories also.

/W

Displays list in a wide format without date, time, or size.

The DIRCMD environment variable can be used to set the default preferences for the DIR command. The SET command will assign the values to an environment

variable. Refer to the SET command section for proper usage. For example, if you wanted to always have the /P option set for DIR, the statement SET DIRCMD=/P could be used. The default settings in DIRCMD can be overridden by using the minus sign (-) preceding the option. If you wanted to cancel the paging for a single use of the DIR command, you would enter DIR /-P.

Examples

To see the directory entries of all files in the current drive and directory, type:

```
DIR
```

To see all files in the subdirectory MEMOS on drive B, type:

```
DIR B:\MEMOS
```

Display all files sorted by file name order.

```
DIR /ON
```

Display all hidden files.

```
DIR /AH
```

Display all files with a .DOC extension without file sizes, or volume labels.

```
DIR *.DOC /B
```

Command **DOS**

Type **CONFIG.SYS**

Purpose

ROM-DOS can be loaded in special memory, called the high-memory area, or HMA, freeing more of the conventional (lower 640K) DOS memory for use by applications.

Syntax

```
DOS=HIGH
```

Remarks

The DOS=HIGH command frees up more of the standard DOS memory for use by applications.

This command will only work on 286, 386, and 486 CPUs with extended memory and Datalight's HIMEM.SYS high-memory manager, or equivalent, installed. It will not work on standard XT-class PCs. Setting DOS=HIGH is ignored when ROM-DOS is in ROM.

The DOS=HIGH command will also work on Chips and Technology's PC/Chip, if it has extended memory and Chips and Technology's HIDOS.SYS device driver, or another memory driver, is loaded.

See the HIMEM.SYS device driver description in this manual. See also 386MAX by Qualitas for more information.

Example

The following example loads the high-memory-area device driver and then loads ROM-DOS into the HMA for increased conventional memory. The high memory area device driver must be loaded first, before DOS=HIGH.

```
DEVICE=HIMEM.SYS

DOS=HIGH
```

Command ECHO

Type

Internal Batch Subcommand

Purpose

Controls whether ROM-DOS commands and other messages are displayed during batch file execution.

ECHO also allows you to create your own messages for display.

Syntax

```
ECHO [ON≠OFF]

ECHO message
```

Remarks

The ON option is the default ECHO setting. It causes commands in a batch file to be displayed as they are executed by ROM-DOS. Typing ECHO OFF turns off such display, after which the ON option will switch it back on again.

The ECHO command alone, typed without the ON or OFF option, displays the current ECHO setting.

The message option is a string of characters, such as a warning or a reminder, that you want ROM-DOS to display. Although your message will be displayed whether ECHO is on or off, the message display is useful only when ECHO is off.

To create a message, type ECHO followed by your message. If your message is more than one line long, the ECHO command begins each line of the message.

The @ symbol can be used to suppress the echoing of a single command when ECHO is off. Place the @ symbol first on the command line.

Examples

A batch file message of more than one line could be the following:

```
ECHO This batch file moves files
```

ECHO to another directory.

To set ECHO to off , enter the following command:

ECHO OFF

Command ERASE

Type Internal

Purpose

Deletes a specified file or set of files.

Syntax

ERASE [*d:*][*path*]*filename* [/P]

Remarks

The DEL command and the ERASE command are functionally identical.

When no drive is specified, the default drive is assumed. When no path is specified, the default path is assumed.

Global characters (? and *) can be used in the filespec. This should be done with caution as it is possible to delete multiple files unintentionally.

When the filespec *.* is used to delete all files in the specified subdirectory, a verification message will be displayed:

```
Are you sure (Y/N) ?
```

Type Y only if you want all files in the specified subdirectory to be erased (deleted).

Caution

There is not any ROM-DOS command to "undelete" a file. Although utilities exist which can attempt an undelete, certain factors can cause the deleted file to be partially or totally lost. The ERASE command should be treated as a permanent delete.

ERASE deletes files within a subdirectory, not the subdirectory itself. For subdirectory removal, see the RMDIR command.

Options

/P

The /P option causes ERASE to pause and prompt the user before each file is deleted. This option is most useful when deleting files with wildcards. A sample prompt is shown below:

***Filename*, Erase (Y/N) ?**

Examples

Erases the file MYLETTER.DOC from the current default subdirectory.

```
ERASE MYLETTER.DOC
```

Erases all files in the current subdirectory with a .DOC file extension.

```
ERASE *.DOC
```

This example erases all files on the A: drive with a .DOC extension. Before each file is erased the user is prompted to determine if that file should be erased.

```
ERASE A:*.DOC /P
```

Command EXIT

Type Internal

Purpose

Exits a secondary "nested" ROM-DOS operation, and returns control of the system to the primary program.

Syntax

```
EXIT
```

Remarks

The EXIT command has no effect if a secondary COMMAND.COM command processor has not been loaded since the primary COMMAND.COM is always loaded in a "permanent" mode. A secondary COMMAND.COM will be effected if it is loaded without the /P permanent option.

Command FCBS

Type CONFIG.SYS

Purpose

The FCBS command allows you to specify the number of file control blocks (FCBs) open at one time.

Syntax

FCBS = *number* [,*minimum number*]

Remarks

Number specifies the maximum number of FCBs open at any given time. The default for this value is 4. The value for *number* must be in the range from 1 to 255. The minimum number specifies the minimum number of FCBs to be open at all times. The *minimum number* argument has the same default and range value as the number argument.

Example

Set the maximum number of FCBs to 8 and leave at least 4 open at all times.

```
FCBS = 8,4
```

Command FILES

Type CONFIG.SYS

Purpose

The maximum number of files that may be open at one time can be modified using the FILES command.

Syntax

FILES = *number*

Remarks

The number of files includes the standard files stdin, stdout, stderr, stdprn, and stdaux. The minimum this value may be set to is 8, the maximum is 255. All other values are ignored.

Example

```
FILES = 10
```

This example causes the maximum for open files to be ten.

Command FIND

Type External

Purpose

FIND is a filter to display only lines that contain a specified string.

The input to FIND may come from a file, or it may be piped in from another filter or a DOS command.

Syntax

FIND [/*option*] *match-string* [*filename*]

Options

The /C option causes FIND to only display the count of lines found with the specified string.

The /N option causes FIND to display the line number of the line found containing the string.

The /V option causes FIND to display the lines that do not contain the string.

The match-string argument specifies the word or group of characters to search for.

The filename argument specifies the file or group of files to search in. The complete drive and path can be specified. Wildcard characters can be used in the filename.

Examples

The following example shows each line in the file JUNK.C that contains the matchstring "printf".

```
FIND printf junk.c
```

The following example shows each line in a directory listing that contains a DIR. The command first executes a DOS DIR command with the output piped into the FIND command. The FIND command then displays each line that contains the "DIR" string.

```
dir ≠ FIND DIR
```

The following example give a count of the lines in the file MANUAL.TXT that contain the string ROM-DOS.

```
FIND /C ROM-DOS MANUAL.TXT

.....MANUAL.TXT: 105
```

Command FOR

Type Internal Batch Subcommand

Purpose

FOR allows repeated execution of a ROM-DOS command applied to a set of files.

Syntax

FOR %%*variable* IN (*set*) DO *command* %%*variable*

Remarks

In execution, this command attaches the variable as an identifier to each file in the set of files described. It then applies the command to each of these identified files. The set may be an exact list of complete file names, or a global file specification using wildcard characters.

The FOR subcommand can be used directly on the command line as well as within a batch file. To use on the command line, substitute a single percent (%) symbol for the double percent signs (%%).

Examples

```
FOR %%N IN (Q1.TXT Q2.TXT) DOPRINT %%N
```

This command will print only the files Q1.TXT and Q2.TXT.

```
FOR %%N IN (*.TXT) DO PRINT %%N
```

This command will print all files, in the current default directory, with a .TXT extension.

Command GOTO

Type Internal Batch Subcommand

Purpose

The GOTO subcommand transfers control to another line of the batch file.

Syntax

```
GOTO label
```

Remarks

The *label* is another line in the batch file consisting of a string up to eight characters long. The label may be an environment variable.

If the specified label is not found, then the batch file terminates with the error message Label not found.

Examples

```
GOTO MESSAGE
```

This command will move the control of execution within the batch file to a line that says:

```
MESSAGE
```

Note: A batch file label must be preceded by a colon (:).

Command HELP

Type Internal

Purpose

Provides on-line help of each ROM-DOS command.

Syntax

HELP <*command*>

Remarks

HELP serves as a memory aid. For complete information about ROM-DOS commands, always consult this manual.

HELP for each command can also be displayed by entering /? following the command name.

The file COMMAND.HLP must be available to use this command.

Examples

To list the help of the DIR command you can type:

```
HELP DIR
```

or

```
DIR /?
```

All available batch file commands are also listed by HELP.

Command IF

Type Internal Batch Subcommand

Purpose

The IF subcommand allows conditional execution of commands.

Syntax

```
IF [NOT] condition command
```

Remarks

The condition may be any one of the following:

```
ERRORLEVEL number

string1 == string2

EXIST [d:][path]filename
```

If the condition is true then the command is executed, otherwise the command is bypassed and the next command in the batch file is executed. The [NOT] option will test the opposite of any of the conditions.

The ERRORLEVEL *number* is true if the last program to execute had an exit code equal or greater than *number*. Using the [NOT] option with this condition will test if the exit code is less than the *number* argument.

The condition *string1* == *string2* is only true when *string1* and *string2* are identical. The strings must match exactly; uppercase/lowercase mismatches are not allowed. Applying the [NOT] option will create a condition that is true only when the strings are not identical.

The EXIST condition is true if the specified filename is found. Wildcard characters are allowed in the filename. The [NOT] EXIST condition will be true if the filename cannot be found.

Examples

```
IF ERRORLEVEL 15 GOTO :EXIT
```

This command will GOTO the :EXIT label if the ERRORLEVEL was equal to or greater than 15.


```
IF %1 == CONFIG.SYS PRINT %1
```

This command will print the file stored as the %1 parameter only if its exact name is CONFIG.SYS.

```
IF NOT EXIST OLD COPY CONFIG.SYS OLD
```

If there is not a file named OLD, this command will copy CONFIG.SYS to OLD.

Command INCLUDE

Type CONFIG.SYS

Purpose

The INCLUDE command can be used to include the contents of one configuration block into another. The instructions from the originating instruction block, as well as the included block, will be carried out. This command can only be used within a CONFIG.SYS configuration block.

Syntax

INCLUDE = *blockname*

Remarks

This command is useful for sets of instructions common to several system configurations. The commands can be defined once in a single configuration block and then inserted into other configuration blocks via the INSERT command. Please refer to the section on Using Multiple Configurations for more details.

Examples

```
:

:

[MISC]

device=mouse.sys

device=c:\netword\loadnet.sys

[WORDPROC]

files=20

buffers=10

set path=c:\bin;c:\wp;c:\dict
```

```
INCLUDE=MISC
```

If “WORDPROC” was chosen from a CONFIG.SYS menu, the instructions in the configuration block labeled [WORDPROC] would be carried out. The instructions in the INCLUDED block labeled [MISC] would also be implemented as part of the [WORDPROC] block of instructions.

Command INSTALL

Type CONFIG.SYS

Purpose

The INSTALL command loads terminate-and-stay-resident (TSR) programs during CONFIG.SYS processing.

Syntax

INSTALL = [*d*:\][*path*] *TSR_Program TSR_Arguments*

Remarks

The TSR program is loaded much the same as if loaded from AUTOEXEC.BAT, except that an environment is not created. The lack of an environment may cause some programs to execute incorrectly. These programs must be loaded from the AUTOEXEC.BAT file.

Examples

```
INSTALL = C:\BIN\FLASHDSK.EXE H:
```

This example causes the Datalight FLASH Disk File Manager to be loaded from CONFIG.SYS using INSTALL. The program is specified with a full drive and path description, and also has an argument of “H:”.

Command LASTDRIVE

Type CONFIG.SYS

Purpose

Set the maximum number of drives.

Syntax

LASTDRIVE = *letter*

Remarks

Letter may be any character between A and Z. It stands for the last drive letter that ROM-DOS will be able to access. The default value is E.

The minimum number LASTDRIVE may be is the number of drives in your computer. If *letter* is less than the number of drives in your computer, then the LASTDRIVE command is ignored.

LASTDRIVE is often used to cause ROM-DOS to make more space for non-standard drives that are not in your system. These drives may be CD-ROM drives, FLASH Disk drives, or network drives.

Example

```
LASTDRIVE = H
```

The above will cause ROM-DOS to allocate for eight drives. If your computer has five installed then there is room for three additional non-standard drives.

Command LOADHIGH

Type Internal

Purpose

Loads an executable or terminate-and-stay-resident (TSR) program into the upper memory area if available. LOADHIGH can be run as a batch subcommand or from the DOS command line.

Syntax

LOADHIGH = executable [*arguments*]

or

LH = executable [*arguments*]

Remarks

An executable or TSR program can be loaded into the upper memory areas if they are available and there is enough free upper memory to accommodate the program's needs. To make high memory available, the EMM386.EXE and HIMEM.SYS utilities must be loaded. If these utilities are not loaded or there is not enough upper memory available, the program will load into conventional memory.

The full drive path and file name of the device must be specified. The arguments are different depending on the device driver.

Examples

The following example installs an executable called CHECKIT with its command line arguments as specified. The program will load into upper memory if available.

```
LOADHIGH=C:\apps\checkit.exe /p
```

Command MENUCOLOR

Type CONFIG.SYS

Purpose

The MENUCOLOR command allows you to set the text and background colors for the startup menu. This command can only be used in a menu block within your CONFIG.SYS file.

Syntax

MENUCOLOR = *text_color* [,*background_color*]

Remarks

The *text_color* argument selects the display color for the screen text. The color numbers 0 (black) and 7 (white) can be used.

The *background_color* argument is optional. If a value is not entered, the default color 0 (black) will be used. Be sure to specify different colors for background and text and separate the numbers with a comma.

Examples

The following example will display the menu text in white on a black background.

```
MENUCOLOR=7,0
```

Command MENUDEFAULT

Type CONFIG.SYS

Purpose

The MENUDEFAULT command allows you to set the default menu item choice and a time-out value for making a menu selection. This command can only be used with a menu configuration block in the CONFIG.SYS file.

Syntax

MENUDEFAULT = *blockname*[,*timeout*]

Remarks

The *blockname* argument specifies the default menu item. The value for *blockname* must match a configuration block name defined elsewhere in your CONFIG.SYS file.

The optional *timeout* argument represents the number of seconds ROM-DOS will wait for a user input selection before initializing your system with the default configuration. The timeout period can be set to a value between 0 and 90. If you select 0, the default menu item will automatically be implemented without a wait. If you do not enter a timeout value, ROM-DOS will not continue until an Enter key is pressed.

If your system BIOS does not support a video display directly, like Datalight's miniBIOS, please refer to the MENUCOLOR command for special instructions.

Examples

```
[MENU]

menuitem=Word_Proc, Word Processing

menuitem=Network, Network

menuitem=Research, Research and Development

menucolor=15,1
```


```
menudefault=Word_Proc,20
```

The above example makes the Word_Proc configuration block the default menu item. If the user fails to make a selection within 20 seconds, the Word_Proc block will be processed.

Command MENUITEM

Type CONFIG.SYS

Purpose

The MENUITEM command allows you to specify an item on the startup menu. This command can only be used within a menu configuration block in the CONFIG.SYS file.

Syntax

MENUITEM = *blockname* [,*menu_text*]

Remarks

The *blockname* argument is a user defined label given to a configuration block defined elsewhere in the CONFIG.SYS file. If a user selects the menu item, all commands in the selected configuration block will be processed, along with the instructions that are common to all menu choices (denoted by block header [COMMON]). The block name can be up to 70 characters long and may contain most printable characters, including spaces, backslashes (\), forward slashes (/), commas, semicolons (;), equals signs (=). Square brackets ([]) cannot be used in block names.

The *menu_text* argument is a descriptive statement that defines the block name. The menu text will be displayed on the screen as a line item in the startup menu. The *menu_text* argument can be up to 70 characters long and can contain any characters. If this argument is left off, the block name will be used for the startup menu display.

If your system BIOS does not support a video display directly, like Datalight's miniBIOS, please refer to the MENUCOLOR command for special instructions.

Examples

```
[MENU]

menuitem=Word_Proc, Word Processing

menuitem=Network, Network

menuitem=Research, Research and Development
```

```
menudefault=Word_Proc,20
```

The above example defines three menu items, Word_Proc, Network, and Research. Each of these has descriptive text and a set of commands defined later in the CONFIG.SYS file. At boot time, these menu items would be displayed in the startup menu as follows:

```
ROM-DOS 6.22 STARTUP MENU

1. Word Processing

2. Network

3. Research and Development

Enter a choice: 1
```

Command MKDIR (Make DIRectory)

Type Internal

Purpose

Creates a new subdirectory.

Syntax

MKDIR [*d:*][*path*]*subdir*

or

MD [*d:*][*path*]*subdir*

Where *subdir* is the name of the new subdirectory to be created. Note that MD may be used instead of the full MKDIR.

Remarks

If no drive or path is specified, the new subdirectory will be created within (one level below) the current default directory.

If drive and/or path is specified, everything specified must exist or the command will display an error message.

Examples

```
MKDIR TEMPDIR1
```

Creates a new subdirectory called TEMPDIR1 within the current default directory.

```
MKDIR C:\UTIL\TOOLS
```

Assuming the UTIL subdirectory exists, this command creates a new subdirectory called TOOLS within it.

Command NEWFILE

Type CONFIG.SYS

Purpose

The NEWFILE command allows you to continue CONFIG.SYS file processing from a new file. The file can be located on in another directory or even on a different drive.

Syntax

NEWFILE=*filename*

Remarks

The NEWFILE command is especially useful when the CONFIG.SYS file is located on an inaccessible drive or in ROM. Additional device drivers or instructions can be added easily to the new file and will be processed along with the main CONFIG.SYS file upon starting the system.

When the NEWFILE= instruction is processed, control will be passed from the present file (the one containing the NEWFILE instruction) to the file specified in the command. Any commands placed after the NEWFILE instruction in the original file will not be processed. If for some reason the specified file name can not be located, CONFIG.SYS processing will be terminated (even if instructions were to be placed after the NEWFILE command) and the remainder of the startup process will be completed.

NEWFILE commands can be nested. That is, your original CONFIG.SYS can call a second set of instructions via the NEWFILE command. The second file can in turn call a third file by using the NEWFILE command, and so on. Be sure that each filename in the successive steps has a unique name, otherwise, an infinite loop will be created as control is passed back to the same file repeatedly.

Each filename given in a NEWFILE command line will have an environment variable of the same name.

Example

The following example will cause instructions in the file NEWCFG.SYS, located in the C:\BIN directory, to be executed as part of the CONFIG.SYS file. The contents of NEWCFG.SYS may include any of the commands listed in this section.

```
NEWFILE=C:\BIN\NEWCFG.SYS
```

Command NUMLOCK

Type CONFIG.SYS

Purpose

The NUMLOCK command is used to set the Num Lock key on the keyboard to on or off when your computer starts.

Syntax

```
NUMLOCK=[on|off]
```

Remarks

Selecting ON designates that the Num Lock key is set to *on* when DOS boots. Selecting OFF designates that the Num Lock will be *off* when DOS boots. In either case, you will still have the ability to manually turn the Num Lock key on and off after boot up with the NUMLOCK command.

Examples

```
NUMLOCK=on
```

The Num Lock key will be set to on when the system boots.

Command PATH

Type Internal

Purpose

Sets the search path for command files that are not in the current directory.

Syntax

PATH [*d:*][*path*][;[*d:*][*path*]...]

Remarks

Without a specified search path, ROM-DOS will look for an external command file (i.e., one with an extension of BAT, COM, or EXE) only in the current directory. The PATH command tells ROM-DOS which other directories to search after searching the current directory.

Typing only the word

```
PATH
```

by itself displays the current path.

If you want to cancel the command paths you set previously, type:

```
PATH ; or PATH =;
```

Examples

If your applications programs are loaded on a fixed disk, the PATH command enables you to start any of them from any drive or directory. Suppose you want to be able to access utilities, a word processor, and a spreadsheet in subdirectories C:\UTIL, C:\WP, and C:\123. You would set the path command as follows:

```
PATH C:\UTIL;C:\WP;C:\123
```

Command PAUSE

Type Internal Batch Subcommand

Purpose

Suspends the execution of a batch file. Resumes operation when any key is depressed.

Syntax

PAUSE [*message*]

Remarks

A batch job may require that you perform some action such as changing disks, or choose to continue or terminate the operation. When the command processor encounters PAUSE it suspends execution, and displays the message:

Strike a key when ready...

After you perform the appropriate action, or make a decision, striking any key other than the combinations <Ctrl><C> or <Ctrl><Break> resumes the batch job.

If you press <Ctrl><C> or <Ctrl><Break> at this point, ROM-DOS asks:

```
Terminate batch job (Y/N)?
```

Responding "Y" ends the batch job. Strategic placement of the PAUSE command, working with this query, allows you to divide the batch file into sections so you can end it at some intermediate point.

The *message* option allows you to display a reminder on the screen during the pause. Your message will precede the Strike a key message. Note, however, that your message will appear only if ECHO is off.

Examples

```
PAUSE Place blank disk in drive A:
```

The above command will alert the user of the need for a disk and suspend operation until a key has been hit.

Command PRINT

Type External

Purpose

The PRINT utility prints a single file or a list of files.

Syntax

PRINT [*/d*:] [*filename*] [*/options*]

Remarks

PRINT allows you to enter between 1 and 32 files for spooling to the printer. The files are output to the device in a spooled manner (while the user performs other operations).

If PRINT is entered without any parameters, then it displays all the files that are in the queue.

The first time PRINT is used the operator is prompted for the device to perform the operation. The following message is used to prompt the operator for the device.

```
Name of list device [PRN]:
```

The legal devices for printing are LPT1, LPT2, LPT3, LPT4, COM1, COM2, COM3, COM4, AUX, or PRN.

Options

The /B option allows the user to set the buffer size. The default buffer size is 512 bytes. A larger buffer size causes print to operate faster. The maximum buffer size is 32k bytes and the minimum size 256 bytes. This option is only allowed the first time PRINT is run.

The /C option cancels only the file names listed after the /C command.

The /F option allows the user to set the maximum number of files to be queued up at one time. The default number of files is 10. The minimum is 2 and the maximum is 32. Support for more files is often useful when using wild cards in file names.

This option is only allowed the first time PRINT is run (or until the next system reboot).

The /P option causes all files listed after this option to be submitted for printing. This is the default for filenames encountered on the PRINT command line.

The /T option cancels all the files from the print queue (list). Think of this as a terminator.

The /H option will display the help screen.

Examples

PRINT puts three files into the print queue. The first file will start being printed after the command ends.

```
PRINT FILE1.TXT FILE2.TXT FILE3.TXT
```

The file FILE2.TXT will be removed from the print queue. All other files in the queue will print normally.

```
PRINT /C FILE2.TXT
```

All files in the print queue are canceled. Printing may continue for a short time because of the buffer in your printer.

```
PRINT /T
```

Command PROMPT

Type Internal

Purpose

Changes the ROM-DOS command prompt.

Syntax

PROMPT [*text*] [$*character*] [$*character...*]

Remarks

The prompt which ROM-DOS normally displays is the letter of the current drive followed by a right arrow (>) (the "greater-than" symbol). By use of the PROMPT command, this prompt can be changed to include any combination of a message, the current directory, the date, the time, and some other features. The following table shows the codes that can be used with PROMPT.

Code	Corresponding Prompt
$T	Time
$D	Date
$P	Current Path
$V	ROM-DOS version number
$N	Current drive
$G	The > character
$L	The < character
$B	The ≠ character
$Q	The = sign
$H	A backspace (which erases the previous character)
$E	ASCII code for Escape (X'1B')
$_	Start a new line
$$	The $ character

Examples

To show this prompt:

```
Current directory is drive:\path;

Ready for command>
```

you would type in:

```
PROMPT Current directory is $P;$_Ready for command$G
```

To show on separate lines the date, time, and current directory followed by the greater-than character and a space, type:

```
PROMPT $D$_$T$_$P$G<space>
```

here <space> refers to pressing the spacebar once. The resulting prompt will look like:

```
Mon 6-26-1989

10:17:45.99

A:\> _
```

The last character on the last line indicates the position of the cursor.

Command REM (REMark)

Type Internal

Purpose

The REM command has two purposes: to allow comments in a batch or CONFIG.SYS file, and to temporarily disable a command without physically deleting the command from the file. See also the ; command.

Syntax

REM [*message*]

Remarks

The REM command provides information for the user but has no effect on the execution of the batch file.

The comment may be made up of any set of characters. A blank line can also be created by omitting the message portion of the line.

REM can also be used to temporarily disable a command in a batch file or CONFIG.SYS, without having to delete the line from the file. For CONFIG.SYS files, the semicolon (;) can also be used in place of the REM command.

Examples

```
REM This batch file created by

REM T.J. Sherrill
```

These lines could be added at any point in a batch file as user information only.

```
DEVICE=HIMEM.SYS

DOS=HIGH

REM DEVICE=TESTDEV.SYS /P
```

The DEVICE=TESTDEV.SYS statement has been temporarily removed from these CONFIG.SYS instructions. This statement will not be processed again until REM is removed.

Command REN (REName)

Type Internal

Purpose

Changes the name of a file.

Syntax

REN [*d:*][*path*]*filename1 filename2*

Remarks

REN renames files within a directory; it will not move a file to a different drive or directory as part of the command.

The wildcard characters * and ? can be used to rename more than one file at a time.

ROM-DOS will not allow you to give a file a name that matches the name of an existing file in the same directory.

Examples

To rename the file NOTES.DOC in drive B to REPORT.DOC, type:

```
REN B:NOTES.DOC REPORT.DOC
```

To assign the extension TXT to all files with the current extension DOC, type:

```
REN *.DOC *.TXT
```

Command RMDIR (ReMove DIRectory)

Type Internal

Purpose

Removes (deletes) a specified subdirectory.

Syntax

`RMDIR [d:][path]subdir`

or

`RD [d:][path]subdir`

Where *subdir* is the name of the subdirectory being deleted. Note that RD may be used instead of the full RMDIR.

Remarks

If no drive or path is specified, RMDIR will look for the specified subdirectory one level below the current directory.

If a drive or path is specified, everything specified must exist or ROM-DOS will display an error message.

RMDIR will not remove a subdirectory unless it is empty. An error message will be displayed if you attempt to remove a subdirectory that still has files or other subdirectories within it.

Examples

```
RD TOOLS
```

Removes the TOOLS subdirectory from the current directory (assuming TOOLS is an empty directory).

Command SET

Type Internal

Purpose

The SET command is used to set, display, or remove environment variables.

Syntax

SET [*variable* = [*string*]]

Remarks

Environment variables can be used to control the behavior of programs and batch files and also the behavior of ROM-DOS. This command can be used in the AUTOEXEC.BAT and CONFIG.SYS files and on the DOS command line. The environment variables that can be defined with the set command include, but are not limited to, PATH, COMSPEC, PROMPT, and user defined variables.

Using SET *variable* = with no argument string will clear the current environment string for the named variable.

Examples

```
SET PROMPT = $p$g
```

This sets the prompt, although the prompt can also be set with the PROMPT command.

```
SET PROMPT =
```

This clears any previously set prompt settings and returns the prompt to its default state.

Command SHARE

Type External

Purpose

SHARE installs the capabilities for file-sharing and file-locking on your hard disk.

Syntax

SHARE [/*options*]

Or from CONFIG.SYS:

INSTALL=[*d:*][*path*]SHARE.EXE [/*options*]

Remarks

The SHARE utility is most commonly used in a network or multitasking environment where file sharing is necessary. When SHARE is loaded, DOS will utilize the SHARE utility to validate read and write requests from application programs and users.

The /L:# option specifies the maximum number of files that can be locked at one time. The default number of files is 20.

The /U option unloads the share utility and frees the memory. SHARE will not unload if other TSRs have been loaded on top of it. The other TSRs must be unloaded first before trying to unload SHARE.

Examples

The following example loads the SHARE program from the command line:

```
SHARE
```

The next example installs SHARE from the CONFIG.SYS file and changes the maximum number of locked files to 30:

```
INSTALL=C:\UTILS\SHARE.EXE /l:30
```

The final example unloads SHARE and frees the used memory.

```
SHARE /U
```

Command SHELL

Type CONFIG.SYS

Purpose

The SHELL command allows the user to specify a different boot program other than the default COMMAND.COM. ROM-DOS will boot this new program, with arguments, instead of the one specified internally.

Syntax

SHELL = *boot_program arguments*

Remarks

The SHELL command is most often used to start the initial copy of COMMAND with special parameters. One parameter is used for providing a larger environment than the default 128 bytes.

The *boot_program* can be any executable program. The full path, including drive letter, should be specified if the program is not in the root directory of the default drive.

Arguments are optional and program specific. They will vary depending on the *boot_program* being executed by the SHELL command.

Examples

This SHELL command boots the standard Command Processor but sets the environment space to 512 bytes (up from the default 128). The /P parameter tells COMMAND that it is permanent (cannot terminate).

```
SHELL=C:\COMMAND.COM /E:512 /P
```

This example of the SHELL command boots a program called MYPROG.EXE, located in the directory TEMP, instead of the standard Command Processor.

```
SHELL = C:\TEMP\MYPROG.EXE
```

Command SHIFT

Type Internal Batch Subcommand

Purpose

SHIFT moves each replaceable parameter for a batch file one position "to the left." Execution of the SHIFT command allows use of more replaceable parameters in a batch file--beyond the standard set of %0 through %9.

Syntax

SHIFT

Remarks

This command moves the string or value stored for each replaceable parameter one position to the left. Upon execution of SHIFT, the %0 argument assumes the value of the %1 argument, the %1 argument then assumes the value of the %2 argument, and so on.

Examples

The following example batch file will read in a list of files (provided as arguments on the command line), and display each one to the screen. After displaying each one, the SHIFT command copies the next file in the argument list into the %1 slot, verifies the existence of the file and continues.

Command line argument:

```
TYPEIT autoexec.bat config.sys net.bat

TYPEIT.BAT batch file:
:repeat
if EXIST %1 goto doit
goto end

:doit
type %1
pause
shift
goto repeat

:end
@echo All Done
```

Command SORT

Type External

Purpose

The SORT filter sorts a text file and displays the output to the standard device.

Syntax

SORT [*/options*] [*filename*]

Remarks

SORT normally starts its comparisons at the first character in a line.

The input to *SORT* may come from a file or it may be piped in from another filter or a DOS command.

Options

The /+*n* option causes SORT to begin its alphabetical sorting starting at the *n*th position in the string.

The /r option causes SORT to sort in reverse alphabetical order.

Examples

The following example sorts the file NAMES.LST and displays the output to the screen.

```
SORT NAMES.LST
```

The following example produces a directory and then sorts the directory by file size (the file size in a directory display starts on the 14th position each line or string). The output display is then shown one screen at a time. This is done by directing the output from SORT into the display utility MORE.

```
DIR ≠ SORT /+14 ≠ MORE
```

Command STACKS

Type CONFIG.SYS

Purpose

The STACKS command allows for the use of dynamic data stacks to handle hardware interrupts. ROM-DOS does not utilize this command, although it can be added to a CONFIG.SYS file without error. Using the STACKS command will have no effect on the number or size of stacks available.

Syntax

STACKS = *number,size*

Command SUBMENU

Type CONFIG.SYS

Purpose

The SUBMENU command defines a menu item that represents a secondary menu when selected. This command can only be used within a menu configuration block in the CONFIG.SYS file.

Syntax

SUBMENU=*blockname*[,*menu_text*]

Remarks

The *blockname* argument defines the name of the secondary menu block of commands. The block menu must be defined elsewhere in the CONFIG.SYS file, otherwise, ROM-DOS will leave this item off of the startup menu. The label can be up to 70 characters long and can contain most printable characters, including spaces, backslashes (\), forward slashes (/), commas, semicolons (;), and equals signs (=). Square brackets ([]) cannot be used in block names.

The optional *menu_text* argument specifies the text that ROM-DOS will display for this menu item on the startup menu. If this argument is left out, ROM-DOS will display the block name as the text. The menu text can be up to 70 characters long and can contain any character.

The submenu can be defined with any user provided descriptive label. It need not have the [MENU] label.

Examples

```
[MENU]
menuitem=Word_Proc, Word Processing
menuitem=Network, Network
submenu=Research, Research and Development
menucolor=15,1
menudefault=Word_Proc,20
[WORD PROC]
files=10
buffers=10
lastdrive=m
device=c:\network\loadnet.sys
[NETWORK]
```

```
include=Word_Proc
numlock=off
[RESEARCH]
menuitem=proj1, Project 1
menuitem=proj2, Project 2
menudefault=proj1
[PROJ1]
files=50
buffers=25
numlock=on
[PROJ2]
files=10
buffers=20
device=vdisk.sys 64 /e
numlock=off
```

In the above example, a submenu is defined as one of the startup menu choices. If you were to select Research and Development from the first menu, a secondary menu would be displayed offering the choices of Project 1 and Project 2. The actual commands for Project 1 and Project 2 are defined in the configuration blocks labeled [PROJ1] and [PROJ2].

Command SWITCHES

Type CONFIG.SYS

Purpose

The SWITCHES command allows special CONFIG.SYS file options.

Syntax

```
SWITCHES=[/k][/n][/f]
```

Remarks

The /k argument makes an enhanced keyboard behave like a conventional style keyboard.

The /n argument prevents the use of the F5 and F8 function keys to bypass the startup commands.

The /f argument instructs ROM-DOS to skip the delay after displaying the Starting ROM-DOS... message at boot time. The delay allows the user time to use the F5 and F8 options to alter the processing of the startup files.

Examples

The following example prevents the user from using the F5 and F8 keys at boot time.

```
switches = /n
```

Command TIME

Type Internal

Purpose

Displays the current time as shown on the system's internal clock. Allows resetting of the clock.

Syntax

TIME [*hh*:*mm*:*ss*] [pm≠am]

Remarks

The time set by this command is used, among other things, for "time stamping" your file revision dates. This information is displayed when you execute a directory listing of your files.

You may want to include the TIME command in your AUTOEXEC.BAT file, to set the date at bootup. If your computer has an internal battery-operated clock, you won't need to do so.

The format of the time command is also dependent on the country specified in CONFIG.SYS. The time is displayed according to local standards for the specified country.

Also see the DATE command.

If you just want to check the time maintained by ROM-DOS, type the TIME command by itself. ROM-DOS will display something like:

```
Current time is 3:00:02.48p

Enter new time:
```

after which you merely press <Enter> to return to an empty command line.

If you want to change the time you can include the desired time on the prompt line after the word TIME. Or you may type the command with no option (as you do to check the time) and enter the new time before pressing <Enter>.

ROM-DOS displays the time according to the 24-hour clock with the a or p indicator to show AM or PM. The AM/PM indicator can be typed as a or p or as

am or pm. The time may be entered in a 24-hour format or a 12-hour format with the AM or PM designator.

The allowed options for hours and minutes are:

***hh* = 0–24 *mm* = 0–59 *indicator* = a, p, am, or pm**

ROM-DOS displays time to hundredths of seconds. When entering time, however, you needn't enter seconds or hundredths; ROM-DOS will assume a value of zero if they are not specified.

You may skip the display and prompting by simply typing the current time after the word TIME on the command line:

```
TIME 23:24
```

ROM-DOS will accept your entry as the current time.

Examples

To enter the time 11:15 pm, you can type:

```
TIME 23:15
```

or

```
TIME 11:15 p
```

To enter the time as 9:26 am, you can type:

```
TIME 9:26
```

or

```
TIME 9:26 am
```

Command TREE

Type External

Purpose

The TREE command displays each subdirectory and optionally the files within them for a specified drive.

Syntax

TREE [*d:*] [/F]

Remarks

The TREE command displays the full path of each subdirectory on a specified disk.

The *d:* specifies the drive that TREE will display the subdirectories from. This argument must be specified.

Options

The /F switch causes TREE to display the files in each subdirectory.

Examples

This command will display all subdirectories on drive C.

```
C:\DATA> TREE C:
```

This command will display all subdirectories on drive A: along with the files within each sub directory.

```
C:\DATA> TREE A: /F
```

Command TYPE

Type Internal

Purpose

Displays the contents of a text file on the screen.

Syntax

TYPE [*d:*][*path*]*filename*

Remarks

If a file containing formatting codes or other non-alphanumeric characters is displayed with TYPE, you will see unintelligible characters and possibly hear beeps. This will not harm the system.

Examples

To display the AUTOEXEC.BAT file on drive A, enter:

```
TYPE A:AUTOEXEC.BAT
```

Command VER

Type Internal

Purpose

Displays the version number of ROM-DOS in use. Allows revision of this version number.

Syntax

VER [*n.nn*] [/R]

Remarks

If a new version number is specified, two digits after the decimal are required. Note that this command revises only the record of the DOS version number; it does not change the actual operating system loaded in the computer.

The version command shows both the version of the VER command itself and the version of DOS in operation.

Options

The /R option shows the full version and release number of ROM-DOS.

Example

The following example changes the record of current DOS version in use to DOS 5.0. Any programs that are executed, following this command, will recognize that DOS 5.0 is running.

```
VER 5.0
```

Command VERIFY

Type Internal

Purpose

Display or modify the verify state.

Syntax

```
VERIFY [ON ≠ OFF]
```

Remarks

The VERIFY command will set the state to on or off. When VERIFY is on then each following disk write operation will be verified. This is usually used when critical data is being written to disks.

If VERIFY is on, then each disk write operation will take longer. Due to this extra time for disk operations, most users leave VERIFY off during normal operations.

Examples

```
VERIFY
```

The above example will display the verify state.

```
VERIFY ON
```

This example will set the verify state to on, causing ROM-DOS to verify each following write to a disk.

Command VOL

Type Internal

Purpose

Display the volume label on a specified disk.

Syntax

VOL [*d:*]

Remarks

If you do not specify a drive, then the default drive is assumed. VOL does not allow the setting of volume labels. Refer to the LABEL command for instructions on setting the volume labels.

Examples

```
A>VOL
```

The above will cause ROM-DOS to display the volume label on the default drive which is the A: drive.

```
A>VOL C:
```

The above will cause ROM-DOS to display the volume label on the C: drive.

Command XCOPY

Type External

Purpose

The XCOPY command copies multiple files and optionally subdirectories from one disk to another.

Syntax

XCOPY [*source*] [*target*] [*/options*]

Remarks

The XCOPY command is used for copying multiple files and subdirectories, if they exist.

The *source* and the *target* parameter are complete drive path and file specification descriptions. If you do not specify a path, XCOPY assumes the default path. If a file name is not specified then *.* is assumed.

The ATTRIB command may be used to modify the archive bit for the various XCOPY options that check the archive status of files. Refer to the ATTRIB command description for instructions.

Options

The /A copies only source files that have the archive bit set in them. The archive is not reset.

The /D<*mm-dd-yy*> option causes XCOPY to copy only those files with a date later than the date specified in the /D option.

The /E option causes XCOPY to create subdirectories on the target even if they are empty.

The /M option causes XCOPY to copy only those source files that have the archive bit set. Once the source file is copied the archive bit is reset.

The /P option causes XCOPY to prompt before each file is copied. The prompt appears as follows:

```
C:\COMMAND.COM (Y/N)?
```

If a Y is entered then the file is copied, otherwise, the file is not copied.

The /S option causes XCOPY to copy files in subdirectories of the source directory.

The /V option causes XCOPY to verify each write to the disk.

The /W option causes XCOPY to wait before starting to copy files. The following prompt is displayed.

```
Press any key to begin copying file(s)
```

Example

XCOPY to the A: drive all files in the BIN subdirectory that have an EXE extension and that have the archive bit set.

```
XCOPY \bin\*.exe a: /a
```

Command XDEL

Type External

Purpose

The XDEL command deletes files and subdirectories including empty subdirectories.

Syntax

XDEL *filespec* [/*options*]

Remarks

The XDEL command allows the deletion of files and subdirectories in the same step.

The *filespec* argument is the starting point for the deletion. The *filespec* argument can contain the drive and path for reaching the starting point and can also contain wild card characters to designate a group of file or directory names.

Options

The /D option deletes empty subdirectories. Deletion will not occur if there are any files in the subdirectory, unless the /S option is specified along with the /D option.

The /P option gives you a confirmation prompt before deleting each file.

The /R option allows deletion of read-only files without having to change the file attributes prior to the delete.

The /S option deletes files in subdirectories located below the specified starting directory. When used along with the /D option, it will delete the files within a subdirectory along with the subdirectory entry itself.

Examples

The following examples will all use the diagram below as a reference:

The following XDEL command would delete all of the files in the directories DIR1, SUB1, and SUB2 but not the directory headings themselves.

```
XDEL DIR1 /s
```

To delete the empty subdirectory heading for SUB3, use XDEL as follows:


```
XDEL DIR1 /d
```

To delete all of the files in the three directories and the directory headings at the same time, use the following command:

```
XDEL DIR1 /s /d
```

Index

Symbols

A

B

C

Index

S

T

U

V

X

Y

Tell Us What You Think...

We'd like to know what you think about this manual. Please take a moment to fill out this questionnaire and fax this form to: (516) 738-3318, or mail to:

Symbol Technologies, Inc.
One Symbol Plaza M/S B-4
Holtsville, NY 11742-1300
Attn: Technical Publications Manager

IMPORTANT: If you need product support, please call the appropriate customer support number provided. Unfortunately, we cannot provide customer support at the fax number above.

User's Manual Title:_____

_____(please include revision level)

How familiar were you with this product before using this manual?

®Very familiar ® Slightly familiar ® Not at all familiar

Did this manual meet your needs? If not, please explain. _____

_____

_____

What topics need to be added to the index, if applicable?_____

_____

What topics do you feel need to be better discussed? Please be specific. _____

_____

_____

What can we do to further improve our manuals? _____

_____

_____

Thank you for your input—We value your comments.