



Commands Reference

Operating System/2™
Extended Edition
Version 1.1

Programming Family

First Edition (November 1988)

The following paragraph does not apply to the United Kingdom or any country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

It is possible that this publication may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Requests for copies of this publication and for technical information about IBM products should be made to your IBM Authorized Dealer or your IBM Marketing Representative.

© Copyright International Business Machines Corporation 1988. All rights reserved.

Trademarks

The following trademarks apply to this book:

- IBM is a registered trademark of International Business Machines Corporation.
- Microsoft is a trademark of the Microsoft Corporation.
- Operating System/2 and OS/2 are trademarks of International Business Machines Corporation.
- PC Mouse is a trademark of Metagraphics/Mouse Systems.
- Personal Computer AT and AT are registered trademarks of International Business Machines Corporation.
- Personal Computer XT and PC XT are trademarks of International Business Machines Corporation.
- Personal Computer XT Model 286 is a trademark of International Business Machines Corporation.
- Personal System/2 and PS/2 are registered trademarks of International Business Machines Corporation.
- Proprinter is a trademark of International Business Machines Corporation.
- Quietwriter is a registered trademark of International Business Machines Corporation.
- VisiOn is a trademark of VisiCorp.

About This Book

This book contains user information for IBM Operating System/2 Extended Edition (referred to as the OS/2 program in the remainder of this book). This book provides information about the use of OS/2 commands and shows how to use them.

This book also lists the Communications Manager commands, and the commands and procedure language statements for the Query Manager portion of Database Manager, and describes how and when to use these commands and statements.

Who Should Read This Book

This book is a reference for users who work with the base operating system, Communications Manager, or Query Manager commands.

How This Book Is Organized

This book is organized into five chapters and two appendixes.

- Chapter 1, which describes the OS/2 and DOS operating modes, OS/2 commands, batch file commands and CONFIG.SYS statements.
- Chapter 2, which lists and describes the OS/2 commands.
- Chapter 3, which lists and describes the Communications Manager commands.
- Chapter 4, which lists and describes the Query Manager commands.
- Chapter 5, which lists and describes the Query Manager procedure language statements.
- Appendix A, which describes how to read the syntax diagrams for the commands.
- Appendix B, which provides a quick reference to the OS/2 commands by task.

- A glossary containing a list of the terms and definitions used throughout this book.
- An index is located at the back of this book.

List of Prerequisite Publications

IBM Operating System/2 Extended Edition Version 1.1 Getting Started (referred to as *Getting Started* in the remainder of this book) is the only prerequisite IBM publication for operations in this book. You must complete the tasks in *Getting Started* before using this book.

Getting Started contains:

- An introduction to the OS/2 program
- The steps and basic information needed to install the base operating system, Communications Manager, LAN Requester, and Database Manager, as well as how to use *Introducing OS/2*. It is also recommended that you view *Introducing OS/2* before continuing in this book to familiarize yourself with the OS/2 program and the way you interact with it.

List of Related Publications

The following are related publications that are included in the OS/2 Extended Edition library:

- *IBM Operating System/2 Extended Edition Version 1.1 User's Guide*
- *IBM Operating System/2 Extended Edition Version 1.1 System Editor Guide*
- *IBM Operating System/2 Extended Edition Version 1.1 Keyboard Layouts*
- *IBM Operating System/2 Extended Edition Version 1.1 Structured Query Language (SQL) Concepts.*

The following is a related publication that are included in the Standard Edition library that you can purchase separately:

- *IBM Operating System/2 Standard Edition Version 1.1 Programmer's Toolkit*, (6280211).

The following are related publications for the Communications Manager component of the OS/2 Extended Edition library that you can purchase separately:

- *IBM Operating System/2 Extended Edition Version 1.1 System Administrator's Guide for Communications*, (90X7908)
- *IBM Operating System/2 Extended Edition Version 1.1 EHLLAPI Programming Reference*, (01F0252)
- *IBM Operating System/2 Extended Edition Version 1.1 Programming Services and Advanced Problem Determination for Communications*, (90X7906)
- *IBM Operating System/2 Extended Edition Version 1.1 ACDI Programming Reference*, (90X7911)
- *IBM Operating System/2 Extended Edition Version 1.1 APPC Programming Reference*, (90X7910)
- *IBM Operating System/2 Extended Edition Version 1.1 ECF Server-Requester (SRPI) Programming Reference*, (90X7909).

The following are related publications for the Database Manager component of the OS/2 Extended Edition library that you can purchase separately:

- *IBM Operating System/2 Extended Edition Version 1.1 Database Manager Programming Guide and Reference*, (90X7905)
- *IBM Operating System/2 Extended Edition Version 1.1 SQL Reference*, (90X7907).

The following is a related publication for the OS/2 program that you can purchase separately:

- *IBM Operating System/2 Extended Edition Version 1.1 Problem Determination Guide for the Service Coordinator, (90X7904).*

The following is a related publication for the Local Area Network (LAN) that you can purchase separately:

- *IBM Local Area Network Technical Reference, (25F7688).*

Contents

Chapter 1. Information on Commands	1-1
Types of Commands	1-1
Understanding Icon Symbols	1-2
Chapter 2. OS/2 Commands	2-1
ANSI (Use Extended Keyboard/Display Features)	2-2
APPEND (Set Search Path)	2-3
ASSIGN (Assign Drives)	2-5
ATTRIB (Set File Attribute)	2-6
BACKUP (Back Up Files)	2-7
BREAK (Check for Ctrl+Break)	2-9
BUFFERS (Determine Disk Buffers)	2-11
CALL (Nest Batch Files)	2-12
CHCP (Change Code Page)	2-13
CHDIR or CD (Change Current Directory)	2-14
CHKDSK (Check Disk)	2-15
CLS (Clear Screen)	2-16
CMD (Start OS/2 Command Processor)	2-17
CODEPAGE (Prepare Code Pages)	2-18
COMMAND (Start DOS Command Processor)	2-19
COMP (Compare Files)	2-21
COPY (Copy or Combine Files)	2-22
COUNTRY (Identify Country Information)	2-25
CREATEDD (Create Dump Diskette)	2-27
DATE (Set System Date)	2-28
DDINSTAL (Install Automated Device Drivers)	2-29
DETACH (Process Noninteractively)	2-30
DEVICE (Install Device Drivers)	2-31
ANSI.SYS	2-32
ASYNCDx.SYS Device Driver	2-33
COM0x.SYS	2-35
DFTDD.SYS Device Driver	2-38
EGA.SYS	2-39
EXTDSKDD.SYS	2-40
LANDD.SYS	2-44
MOUSExxx.SYS	2-45
NETBDD.SYS	2-49
PCNETDD.SYS	2-51
PMDD.SYS	2-53

POINTDD.SYS	2-54
SDLCDD.SYS Device Driver	2-55
TRNETDD.SYS	2-58
VDISK.SYS	2-60
DEVINFO (Prepare Code Page Devices)	2-63
DIR (Display Files in Directory)	2-66
DISKCACHE (Allocate Storage Blocks)	2-67
DISKCOMP (Compare Diskettes)	2-68
DISKCOPY (Copy Diskettes)	2-69
DPATH (Set Search Path)	2-70
ECHO (Display Commands as They Process)	2-71
ENDLOCAL (Restore Environment Variables)	2-72
ERASE or DEL (Delete Files)	2-73
EXIT (End a Command Processor)	2-74
EXTPROC (Define External Batch Processor)	2-75
FCBS (Use File Control Blocks)	2-76
FDISK (Manage Partitions)	2-77
FIND (Search Files for Strings of Text)	2-78
FOR (Process Commands Repetitively)	2-79
FORMAT (Prepare a Disk for Use)	2-80
GOTO (Transfer Processing to a Specified Label)	2-82
GRAFTABL (Display Extended Character Sets)	2-83
HELP (Provide System Help)	2-84
IF (Allow Conditional Processing of Commands)	2-85
IOPL (Allow Input/Output Privilege)	2-86
JOIN (Join Drives)	2-87
KEYB (Select a Keyboard Layout)	2-89
LABEL (Create or Change Volume Label)	2-92
LIBPATH (Identify Dynamic Link Libraries)	2-93
MAXWAIT (Set Maximum Wait)	2-94
MEMMAN (Select Storage Allocation)	2-95
MKDIR or MD (Make Directory)	2-96
MODE (Set Communication Modes)	2-97
Set Communication Modes	2-98
Set Display Modes	2-102
Set Parallel Printer Mode	2-103
Set Diskette I/O Write Verification	2-104
MORE (Display Output One Screen at a Time)	2-105
PATCH (Apply Software Repairs)	2-106
PATH (Set Search Path)	2-107
PAUSE (Suspend Batch Operation)	2-108
PAUSEONERROR (Pause CONFIG.SYS Messages)	2-109
PRINT (Print Information)	2-110

PRIORITY (Schedule Threads)	2-112
PROMPT (Change Command Prompt)	2-113
PROTECTONLY (Select Operating Mode)	2-115
PROTSHELL (Load a User Interface Program)	2-116
RECOVER (Recover Files)	2-117
REM (Display Remarks)	2-118
RENAME or REN (Rename Files)	2-119
REPLACE (Replace Files)	2-120
RESTORE (Restore Backed-up Files)	2-121
RMDIR or RD (Remove Directory)	2-123
RMSIZE (Specify DOS Mode Size)	2-124
RUN (Start System Programs)	2-125
ACSPCSYS.EXE System Process	2-126
ACSTRSYS.EXE System Process	2-127
SET (Set Environment Variables)	2-128
SETCOM40 (Set COM Port Address)	2-130
SETLOCAL (Define Local Variables)	2-131
SHELL (Install a Command Processor)	2-132
SHIFT (Use More Than 10 Replaceable Parameters)	2-133
SORT (Sort Information by Letter or Number)	2-134
SPOOL (Specify Print Queue)	2-135
START (Start Programs Automatically)	2-137
SUBST (Substitute Drives)	2-141
SWAPPATH (Specify Swap File)	2-143
SYSLEVEL	2-144
THREADS (Determine Independent Actions)	2-145
TIME (Set System Time)	2-146
TIMESLICE (Allocate Time)	2-147
TRACE (Select or Set System Trace)	2-148
TRACEBUF (Set Size of Trace Buffer)	2-150
TRACEFMT (Display Formatted Trace Records)	2-151
TREE (Display Directory Structure)	2-152
TYPE (Display File Contents)	2-153
UNPACK (Decompress a File)	2-154
VER (Display OS/2 Version Number)	2-155
VERIFY (Set Data Verification)	2-156
VOL (Display Disk Volume Label)	2-157
XCOPY (Copy Subdirectories)	2-158

Chapter 3. Communications Manager Commands	3-1
CM (Start Communications Manager)	3-2
STARTCM	3-3
SEND for VM Host	3-5

RECEIVE for VM Host	3-12
SEND for TSO Host	3-18
RECEIVE for TSO Host	3-25
SEND for CICS Host	3-31
RECEIVE for CICS Host	3-38
Chapter 4. Query Manager Commands	4-1
BEGIN WORK	4-2
CANCEL WORK	4-4
END WORK	4-5
ERASE	4-7
EXPORT	4-9
GET variable	4-11
IMPORT	4-14
PRINT	4-17
RUN	4-21
SAVE DATA AS	4-24
SET variable	4-26
SQLQMFM	4-28
SQLQMFV	4-30
Chapter 5. Query Manager Procedure Language Statements	5-1
Assignment Statement	5-6
DO END	5-7
DO TO BY	5-8
DO UNTIL	5-10
DO WHILE	5-12
EXIT	5-13
IF THEN ELSE	5-14
LEAVE	5-16
SAY	5-17
TRACE	5-18
Appendix A. How to Read the Syntax Diagrams	A-1
Appendix B. Quick Reference Tables for Commands	B-1
Commands for One Mode Only	B-1
Using Configuration Statements	B-3
Using Device Statements	B-5
Obtaining or Changing System Operations	B-7
Managing Your Files	B-9
Creating and Maintaining Your Directories	B-10
Preparing and Maintaining Your Disks and Diskettes	B-11

Piping and Filtering Input and Output B-12
Using Batch File Commands B-13
Using Memory Statements B-14
Managing Your Country Information B-15
Problem Determination and System Software Repair B-16

Glossary X-1

Index X-23

Chapter 1. Information on Commands

This chapter describes the OS/2 and DOS operating modes provided by the base operating system. OS/2 commands, batch file commands, and CONFIG.SYS statements are presented in alphabetic order beginning on page 2-2. For examples and details of the commands and their functions, see the “Using the Base Operating System” section of the *IBM Operating System/2 Extended Edition Version 1.1 User's Guide* (referred to as the *User's Guide* in the remainder of this book).

Types of Commands

The base operating system has two types of commands:

- Internal commands
- External commands.

Internal commands run without any delay associated with loading the program from disk. These commands reside in the OS/2 command processor.

External commands are on the disk as program files. When an external command is entered, the base operating system searches for it in the current directory of the default or specified drive. If not found, the base operating system continues searching for it in the directories listed in the path statement.

Note: System installation creates two special files, each of which contains the necessary paths to OS/2 external commands and data files. When you use Presentation Manager to start an OS/2 command processor, the paths are automatically set by the CONFIG.SYS file. The first time you select the DOS command prompt to start a DOS command processor (or screen group), the paths are automatically set by the AUTOEXEC.BAT batch file. See “Using the Base Operating System” section of the *User's Guide* for more information.

Global file name characters and reserved device names are not allowed in a command name. You can use global file name characters only when specifying file names and file name extensions as command parameters. There are some commands where global file name characters are not permitted in file name parameters. For example, the **FIND** command does not allow them. For more information on global file name characters and reserved names, see the appendix for “File and Directory Concepts” in the *User’s Guide*.

For commands that display a large amount of output, if you are using an IBM Enhanced Keyboard, you can press the Pause key to pause the display output. If you are using an AT keyboard, press the Ctrl and Num Lock keys to pause the display of the output. In both cases, press any character key to continue the display. You can also pipe the command with the **MORE** command by typing the command name followed by a broken vertical bar and **MORE** at the OS/2 command prompt, for example, type:

```
DIR | MORE
```

Understanding Icon Symbols

Two *icons* are used throughout Chapter 2 to describe the modes of operation in which a command works. These symbols can be found on page headings and in the text. When no symbol is shown on the page heading, the command works in both DOS and OS/2 modes. If the symbol is located in the text or next to diagrams (such as in the list of parameters or the syntax diagrams), the parameter or syntax only works in the mode that is shown by the symbol, as follows:

OS/2 Mode Only



DOS Mode Only



Chapter 2. OS/2 Commands

This chapter describes OS/2 commands. The commands appear in alphabetic order and the descriptions include the purpose, syntax and a list of parameters for each command. Examples for each command and how to use each command can be found in “Using the Base Operating System” section of the *User's Guide*.

In addition, a description of how to read the syntax diagram is found in Appendix A.

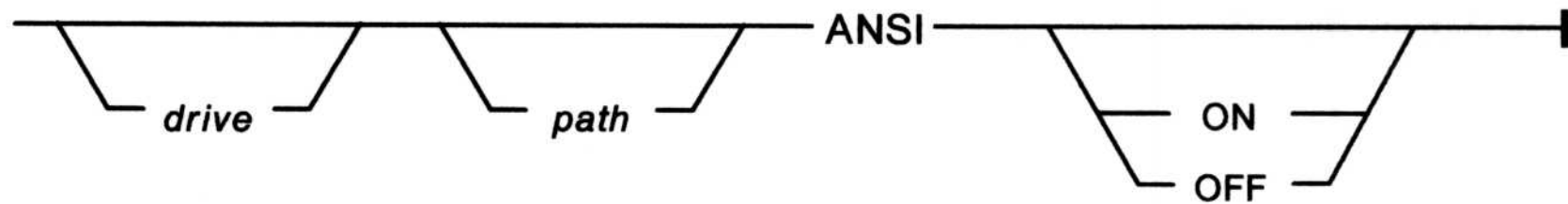
ANSI (Use Extended Keyboard/Display Features)



Purpose

This external command allows extended display and keyboard support in OS/2 mode.

Syntax



Parameters

ON Allows extended display and keyboard support. This is the default value.

OFF Prevents extended display and keyboard support.

Note: Typing this command without a parameter displays its current status of on or off.



APPEND (Set Search Path)

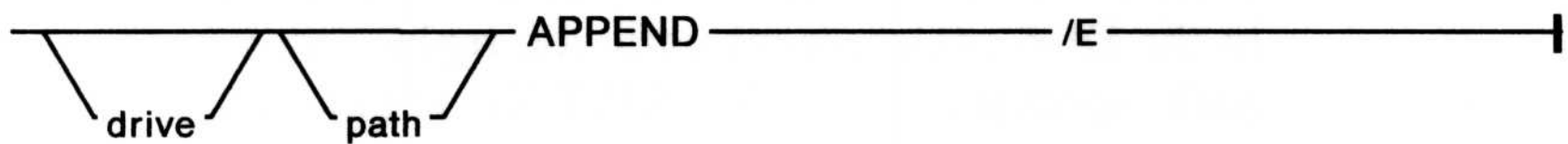
Purpose

This external command sets a search path for data files outside the current directory.

Syntax

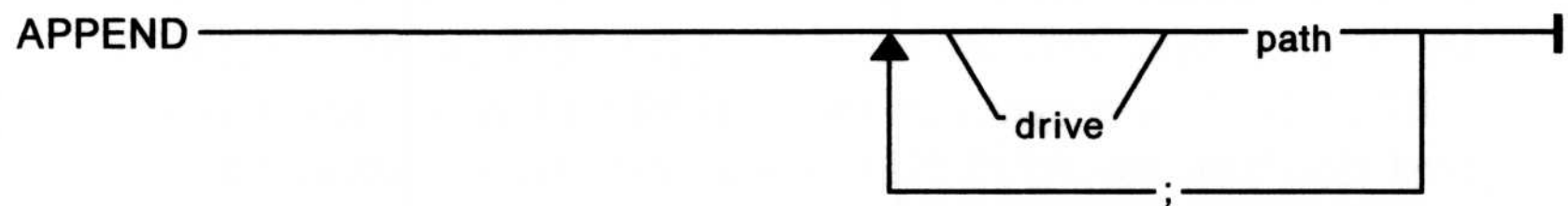
The first time APPEND is loaded

To keep paths specified with APPEND in the DOS environment:



At any time during the DOS session

To specify directories to be searched:



To delete usage of the APPEND command:



APPEND

(Set Search Path)



To display the list of appended directories:

APPEND _____

Parameters

- ;** Cancels an APPEND command.
- /E** Keeps the paths specified in the PATH statement in the DOS environment until other paths are specified by another APPEND command. If you specify the /E parameter the first time you use APPEND, the paths you specify with a subsequent APPEND command are kept in the DOS environment until you replace them with paths specified by another APPEND command.

Notes:

1. The APPEND command can be set by the system installation in the AUTOEXEC.BAT file. Although you may use any of the preceding syntaxes at any time during the DOS session, specifying either of the last two the first time you use APPEND would be of no value to you, because no paths have been specified yet with APPEND. These syntaxes show APPEND as already loaded. The first time you use APPEND, it is an external command; however, once loaded, it functions as an internal command. If, in the preceding syntaxes, you are using APPEND for the first time, you may need a drive and path to locate APPEND. For an explanation of the DOS environment, see the SET command on page 2-128.
2. Typing this command without a parameter displays the current setting of the APPEND command.

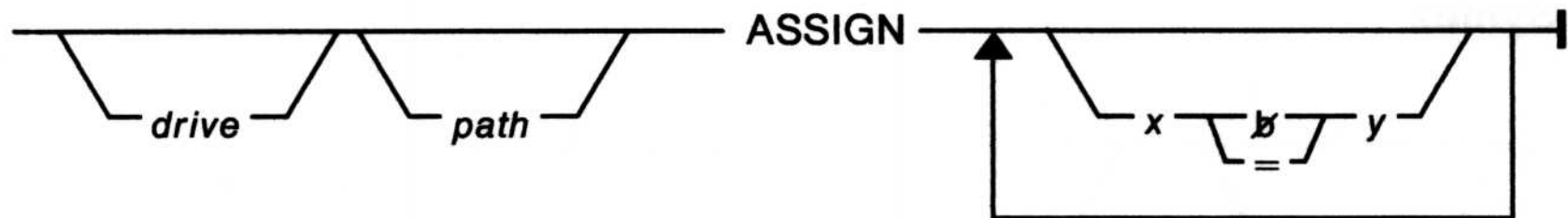


ASSIGN (Assign Drives)

Purpose

This external command assigns a drive letter to a different drive.

Syntax



Parameters

- x* Specifies the drive to which reads and writes are currently sent. Do not type a colon after the drive letter.
- y* Specifies the drive to which you want reads and writes sent. Do not type a colon after the drive letter.

Notes:

1. Typing this command without a parameter resets all drives back to their original assignments.
2. Family API programs cannot be loaded from cross-assigned drives.
3. ASSIGN hides the true device type from commands that require actual drive information. The following commands do not work in DOS mode on drives involved with ASSIGN:

BACKUP
CHKDSK
DISKCOMP
DISKCOPY
FORMAT
LABEL
RECOVER
RESTORE.

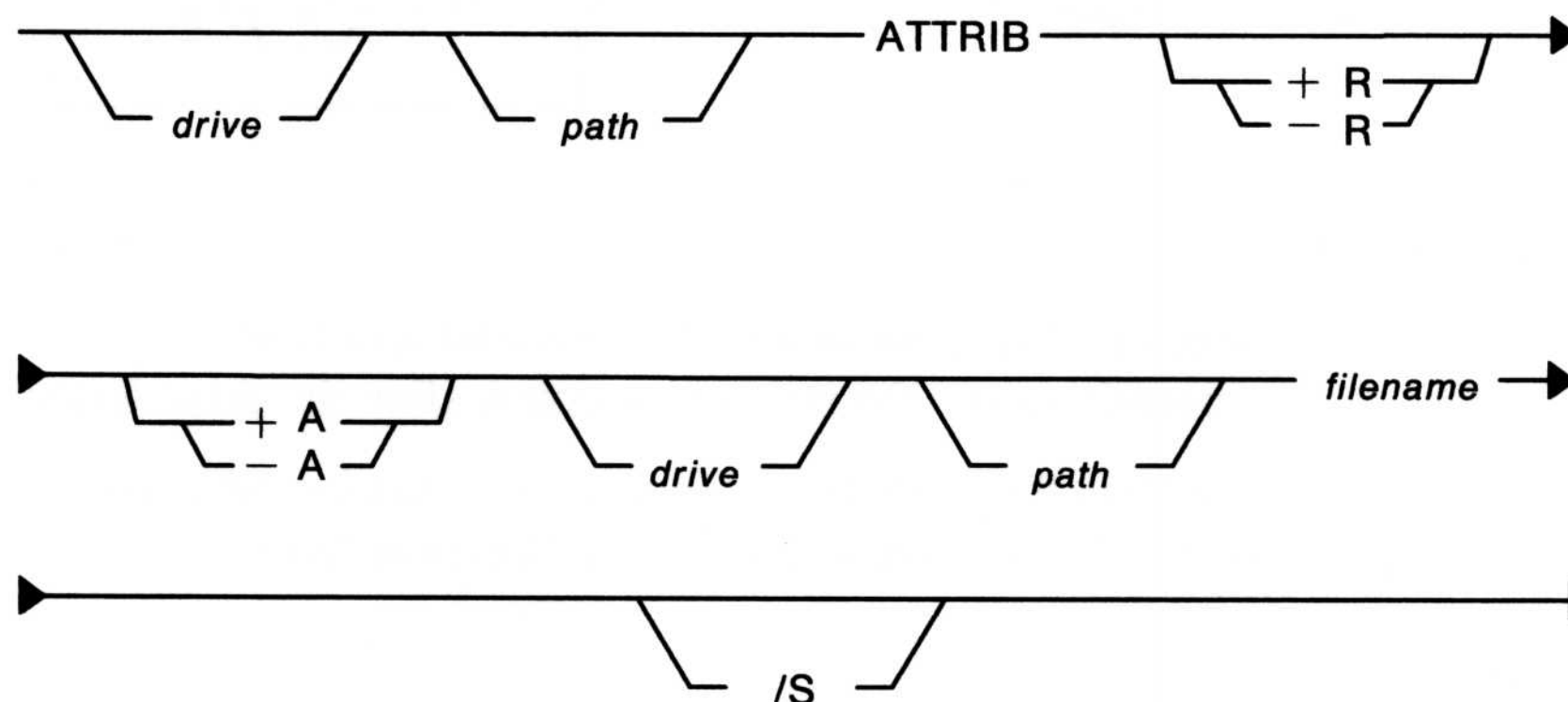
ATTRIB

(Set File Attribute)

Purpose

This external command displays the current file state. It also turns on or off the read-only attribute and the archive bit of a file for selected files in a directory or for all files in a directory level.

Syntax



Parameters

- +R** Turns on the read-only attribute of a file.
- R** Turns off the read-only attribute of a file.
- +A** Turns on the archive bit of a file.
- A** Turns off the archive bit of a file.
- /S** Processes all subdirectories.

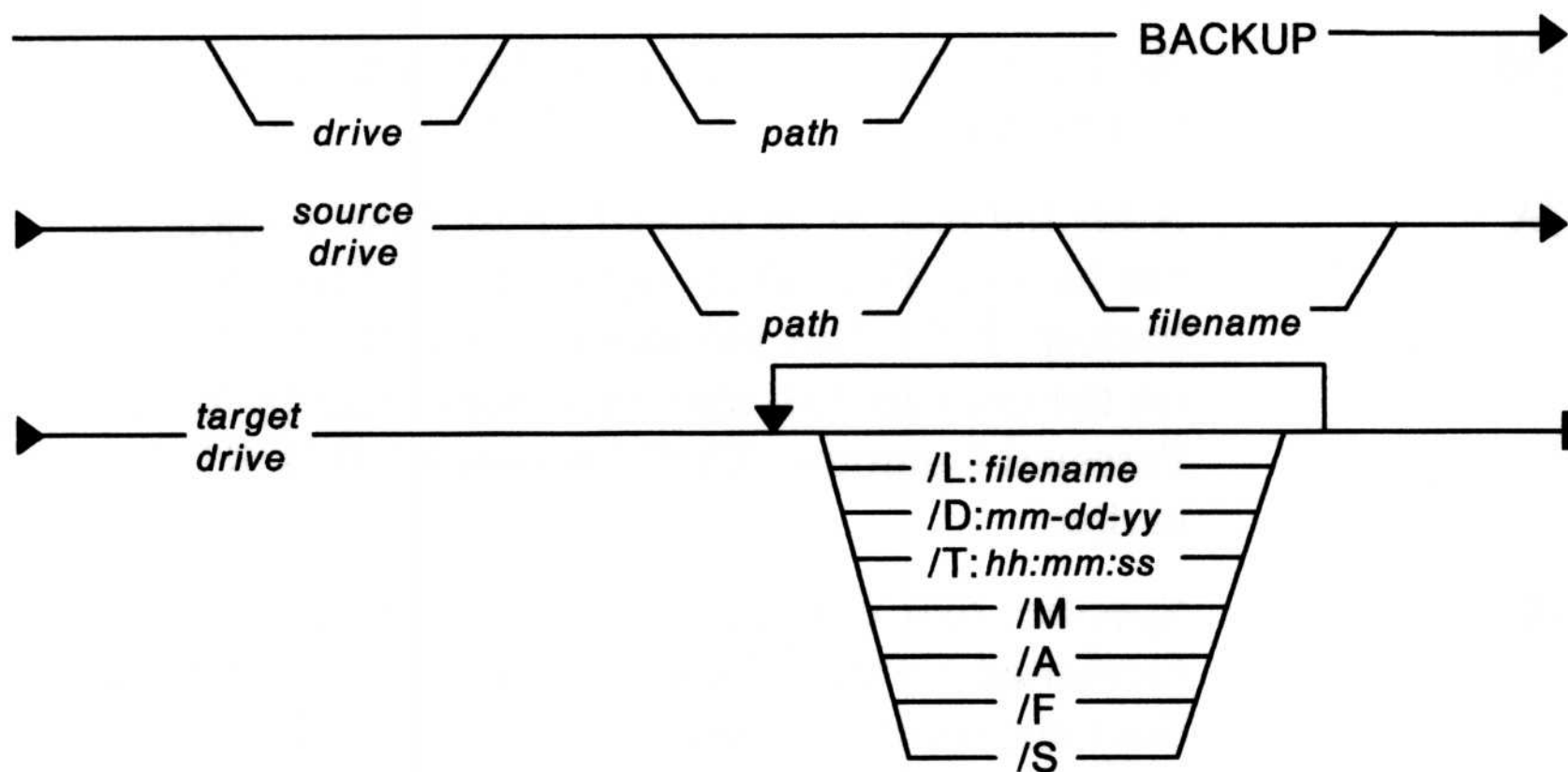
Note: You can specify the optional +R, -R, +A, or -A parameters either before or after the *filename*.

BACKUP (Back Up Files)

Purpose

This external command backs up one or more files from one disk to another.

Syntax



Parameters

/L:filename

Makes a backup log entry in the file specified. If no file name is given, the file **BACKUP.LOG** is placed in the root directory of the source drive. The first line of the entry in the file contains [date time], where date and time are the backup dates and times. Each subsequent line in the entry corresponds to one of the files that was backed up.

These lines consist of the file name and the number of the disk that contains the file. This information can be used when you need to restore a particular file from a diskette. You must specify which disk to restore so that **RESTORE** does not have to search for files. If the backup log file already exists, the current entry is appended to the file.

BACKUP

(Back Up Files)

- /D:mm-dd-yy** Backs up only those files that were last modified on or after the specified date. This parameter expects the date to be entered in the correct format for the current country specified with the COUNTRY command.
- /T:hh:mm:ss** Backs up only those files that were last modified at or after a certain time. It is recommended that /T be used with /D.
- /M** Backs up only those files that have changed since the last backup.
- /A** Adds the files to be backed up to those already on the backup disk. Does not erase old files on the backup disk. This parameter is not valid if files exist on the backup disk that were backed up by IBM Personal Computer DOS Version 3.2 BACKUP or earlier.
- /F** Specifies that unformatted target diskettes are formatted before BACKUP starts. This parameter does not format a fixed disk or previously formatted diskette. If an unformatted diskette is encountered, it is formatted as needed.
- Warning:** When using the /F parameter, the target diskette capacity and the target drive capacity must be identical. For example, do not use a 360KB diskette in a 1.2MB diskette drive.
- /S** Backs up subdirectories.

Note: BACKUP does not work in DOS mode on drives involved with ASSIGN, JOIN, or SUBST.



BREAK

(Check for Ctrl+Break)

Purpose

Allows you to instruct DOS to check if you have pressed the Ctrl and Break keys when a program requests the base operating system to perform any functions.

Syntax

To use **BREAK** at the command prompt:

```
BREAK _____|  
                \  ON  /  
                 \____/
```

To use **BREAK** in your CONFIG.SYS file:

```
BREAK = _____|  
              \____/
```

BREAK

(Check for Ctrl+Break)



Parameters

ON Checks for when you press the Ctrl and Break keys when you request it. It allows you to break out of a program that produces few or no standard device operations (such as a compiler).

OFF Checks to see if you pressed the Ctrl and Break keys only during:

- Standard output operations
- Standard input operations
- Standard print operations
- Standard auxiliary operations.

Note: If you want to display the current status of **BREAK**, type **BREAK** at the DOS prompt.

BUFFERS

(Determine Disk Buffers)

Purpose

Sets the number of disk buffers that the system uses.

Syntax

BUFFERS = _____ x _____

Parameters

x A number from 1 through 100. Values greater than 100 are ignored.

CALL

(Nest Batch Files)

Purpose

Allows a batch file to be called or nested from within another batch file without ending the first batch file. This allows other batch files to be used as commands from a batch file being used as a master file to organize or operate them.

Syntax

```
CALL  _____ batchfile _____|
                \_____/
                arguments
```

Parameters

batchfile The file name of the batch file to begin running.

arguments Specifies information passed to the called batch file.

Warning: In OS/2 mode, a batch file should not CALL itself. If it does, it may run out of system resources.

CHCP (Change Code Page)

Purpose

Allows you to switch back and forth between two code page character sets that are defined in your CONFIG.SYS file.

Syntax

CHCP _____|
 └── nnn ─┘

Parameters

nnn Indicates the code page number you want to use.

Note: Typing this command without a parameter displays the current code page being used or tells you if none are specified.

For more information, see “Using the Base Operating System” section of the *User's Guide*.

CHDIR or CD (Change Current Directory)

Purpose

Changes the current directory or displays its name.

Syntax

CHDIR _____
CD _____ *drive* _____ *path* _____

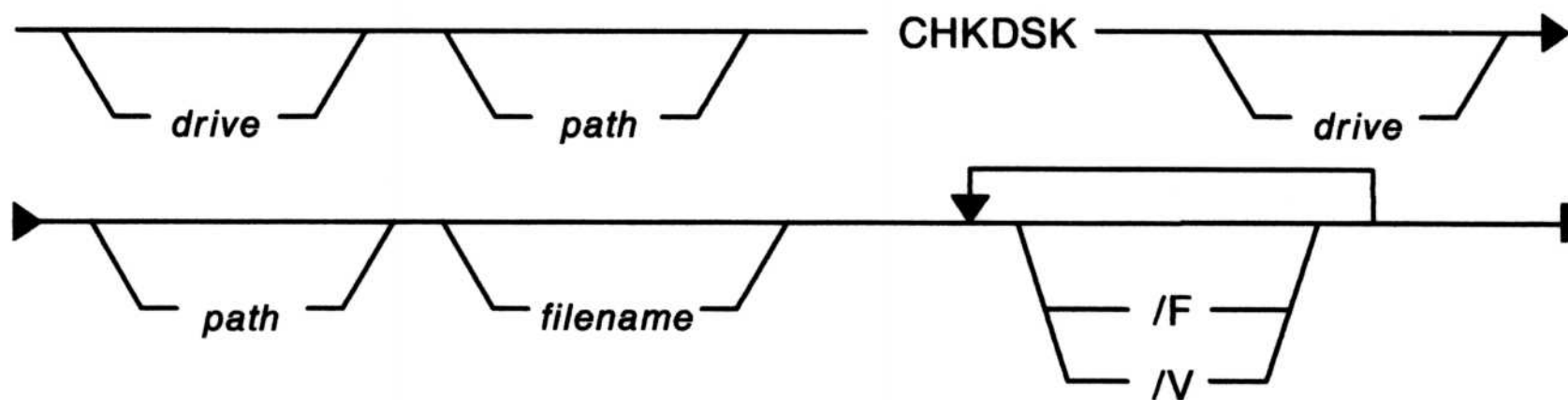
Note: Typing this command without a parameter displays the current directory.

CHKDSK (Check Disk)

Purpose

This external command analyzes the directories, files, and the file allocation table (FAT) on the specified or default drive and produces a disk status report.

Syntax



Parameters

- /F** Fixes errors found in your directory.
- /V** Displays all files and their paths on the default or specified drive.
- filename* Displays the number of noncontiguous areas occupied by the file or files specified.

Notes:

1. Typing this command without a parameter produces a disk storage report in DOS mode.
2. CHKDSK does not work in DOS mode on drives involved with ASSIGN, JOIN, or SUBST.

CLS

(Clear Screen)

Purpose

Clears the window or entire display screen of any information.

Syntax

CLS

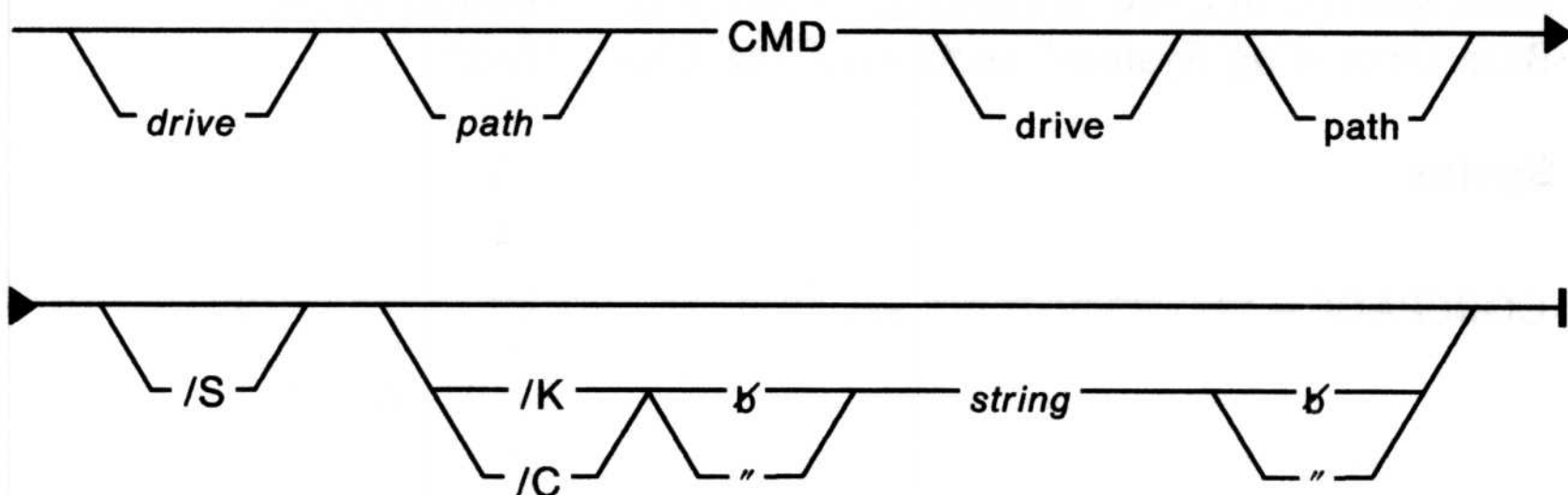


(Start OS/2 Command Processor)

Purpose

This external command starts another command processor in OS/2 mode.

Syntax



Parameters

- drive* Tells the CMD.EXE which drive to set the value of the COMSPEC environment variable to.
- path* Tells the CMD.EXE which path to set the value of the COMSPEC environment variable to.
- /S* Tells the CMD.EXE being started not to install a signal handler (such as ^C) or a request to terminate the program.
- /K string* Allows you to pass a command to CMD.EXE but does not automatically return to the previous command processor after the command is completed. The *string* is the command you want to pass.
- /C string* Allows you to pass a command to CMD.EXE and automatically return to the previous command processor after the command is completed. The *string* is the command you want to pass.

Note: Typing this command without a parameter causes another command processor to be started.

CODEPAGE

(Prepare Code Pages)

Purpose

Selects the system code pages (defined character sets) to be prepared by the base operating system for code page switching.

Note: This statement is one of the interrelated CONFIG.SYS statements required for successful code page switching. For a list of these statements, see Managing Country Information in the “Using the Base Operating System” section of the *User's Guide*.

Syntax

```
CODEPAGE = _____ xxx _____  
                                           /      \  
                                           ,yyy
```

Parameters

- xxx* Specifies the primary code page.
- yyy* Specifies a secondary code page.

Remarks

Code pages supported by the OS/2 program are:

- 437** U.S.
- 850** Multilingual
- 860** Portuguese
- 863** Canada (French-speaking)
- 865** Nordic



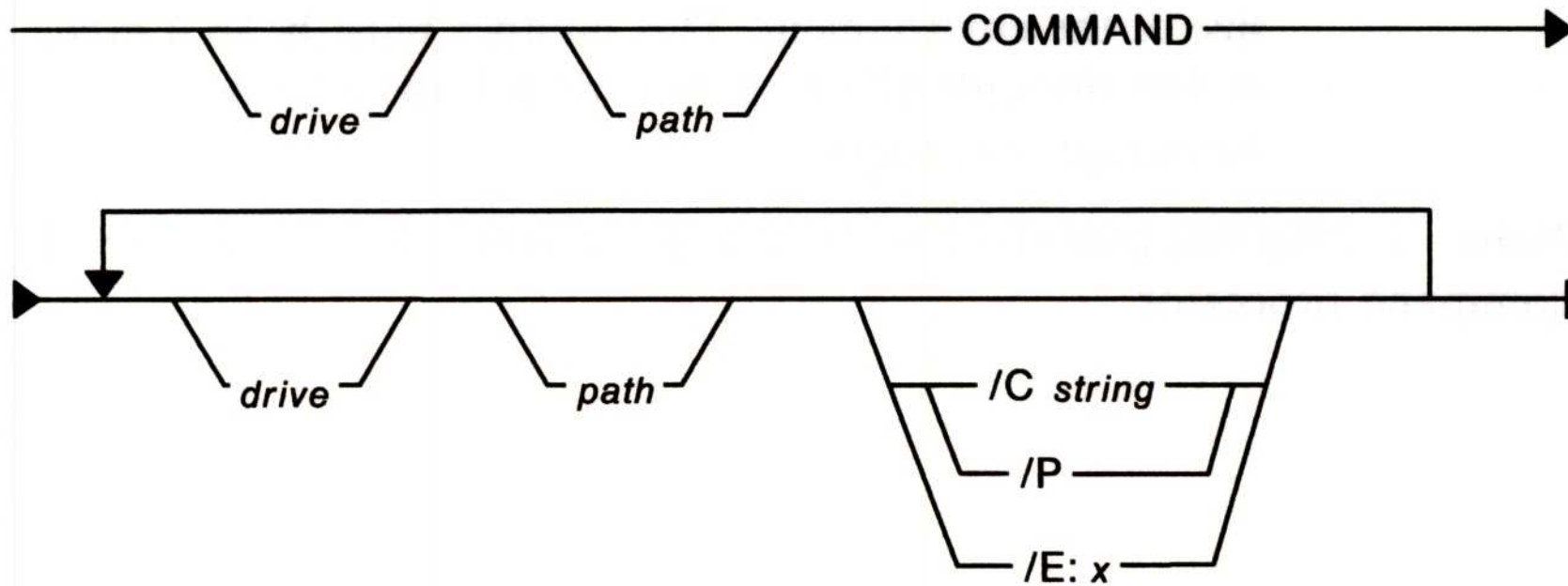
COMMAND

(Start DOS Command Processor)

Purpose

This external command starts another DOS command processor.

Syntax



Parameters

- drive* Tells COMMAND which drive to set the value of the COMSPEC environment variable to.
- path* Tells COMMAND which path to set the value of the COMSPEC environment variable to.
- /C string* Allows you to pass a command to a secondary command processor and return to the primary command processor after the command is completed. The *string* is a command you want to pass to the command processor. The command is interpreted and acted on as if you had entered it at the DOS command prompt. For example, if you type the following and press the Enter key:

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

a secondary DOS command processor is loaded, runs the command DIR B:, and exits back to the primary DOS command processor.

COMMAND

(Start DOS Command Processor)



/P Causes the DOS command processor you start to become permanent in storage. If you specify /P, the new command processor does not return to the previous command processor. You must restart the base operating system to remove the second DOS command processor.

/E:x Specifies a base-10 integer that you can select to set the size of the environment. This number must be in a range of 160 through 32768. It is rounded up to the nearest paragraph boundary.

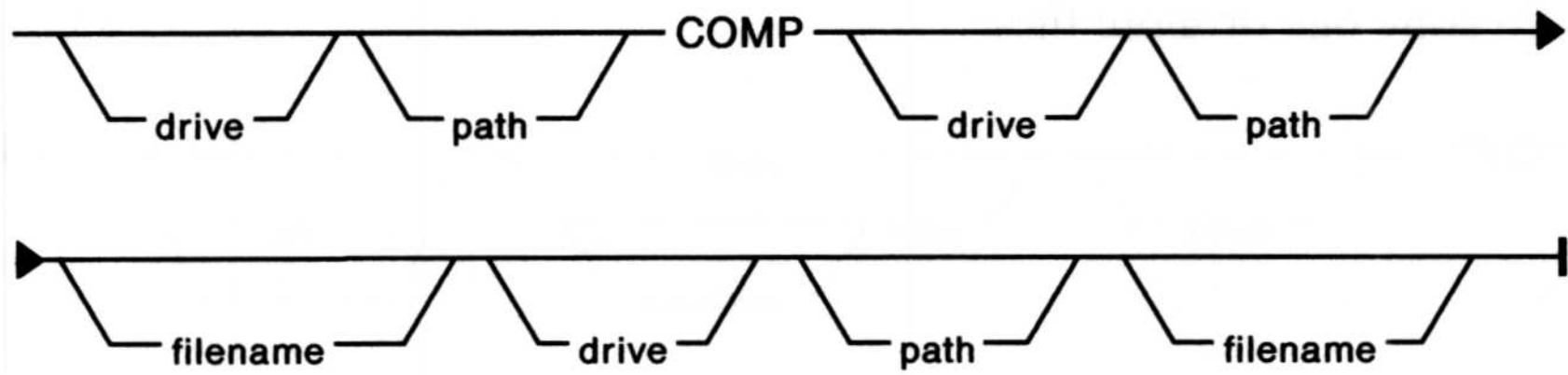
Note: Typing this command without a parameter starts another DOS command processor.

COMP (Compare Files)

Purpose

This external command compares the contents of two files.

Syntax



Note: Typing this command without a parameter starts a step-by-step menu to compare files.

COPY

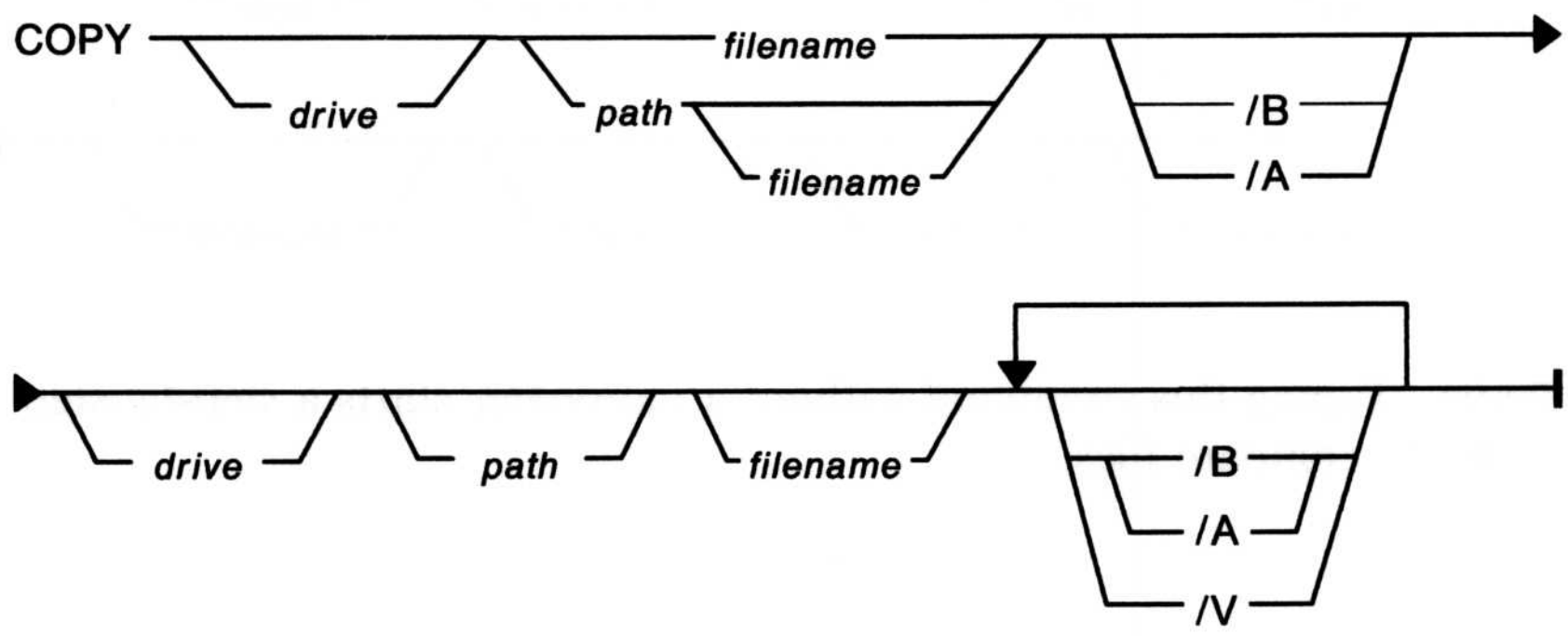
(Copy or Combine Files)

Purpose

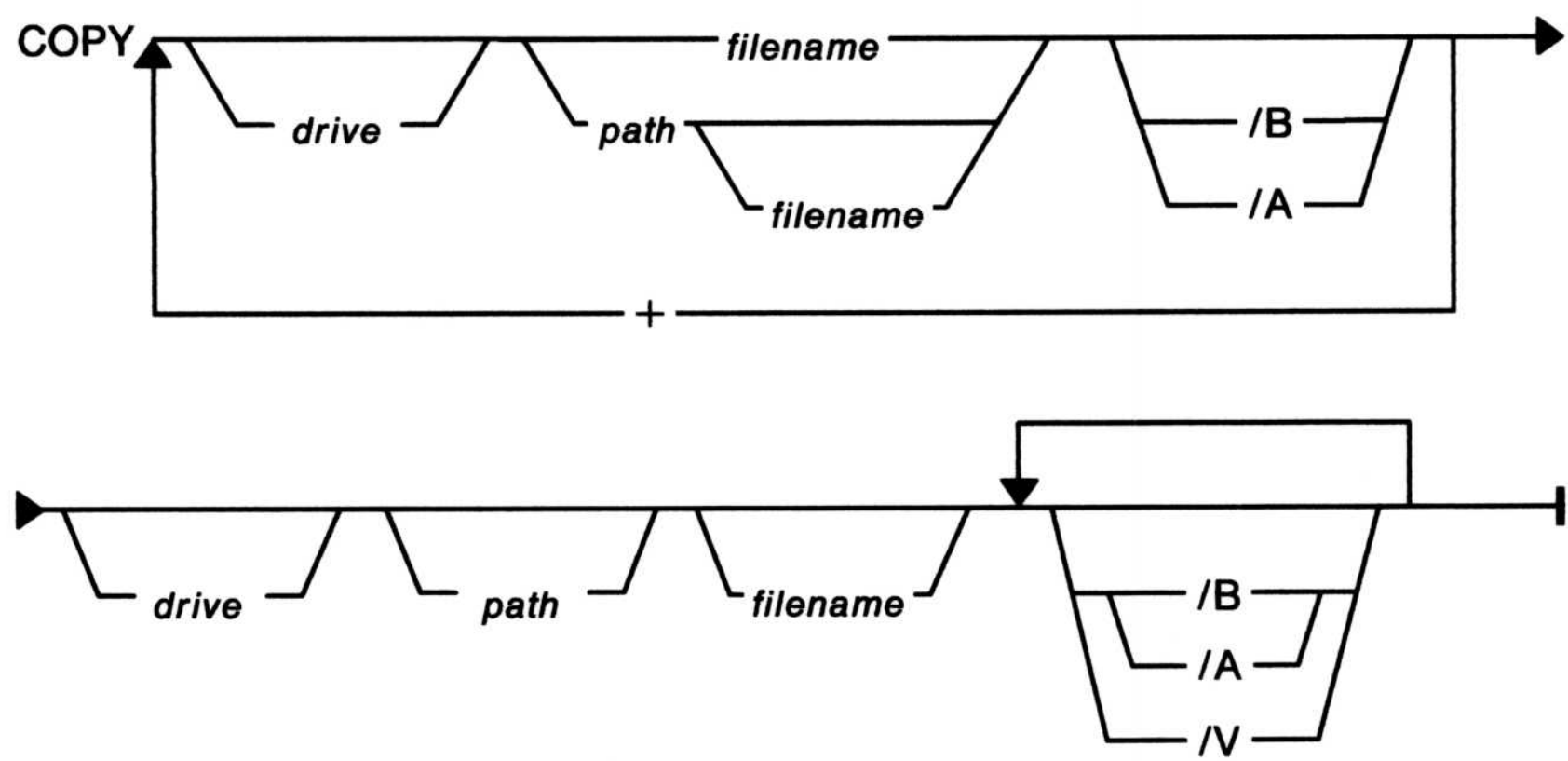
Copies one or more files. This command also combines files.

Syntax

To copy one or more files:



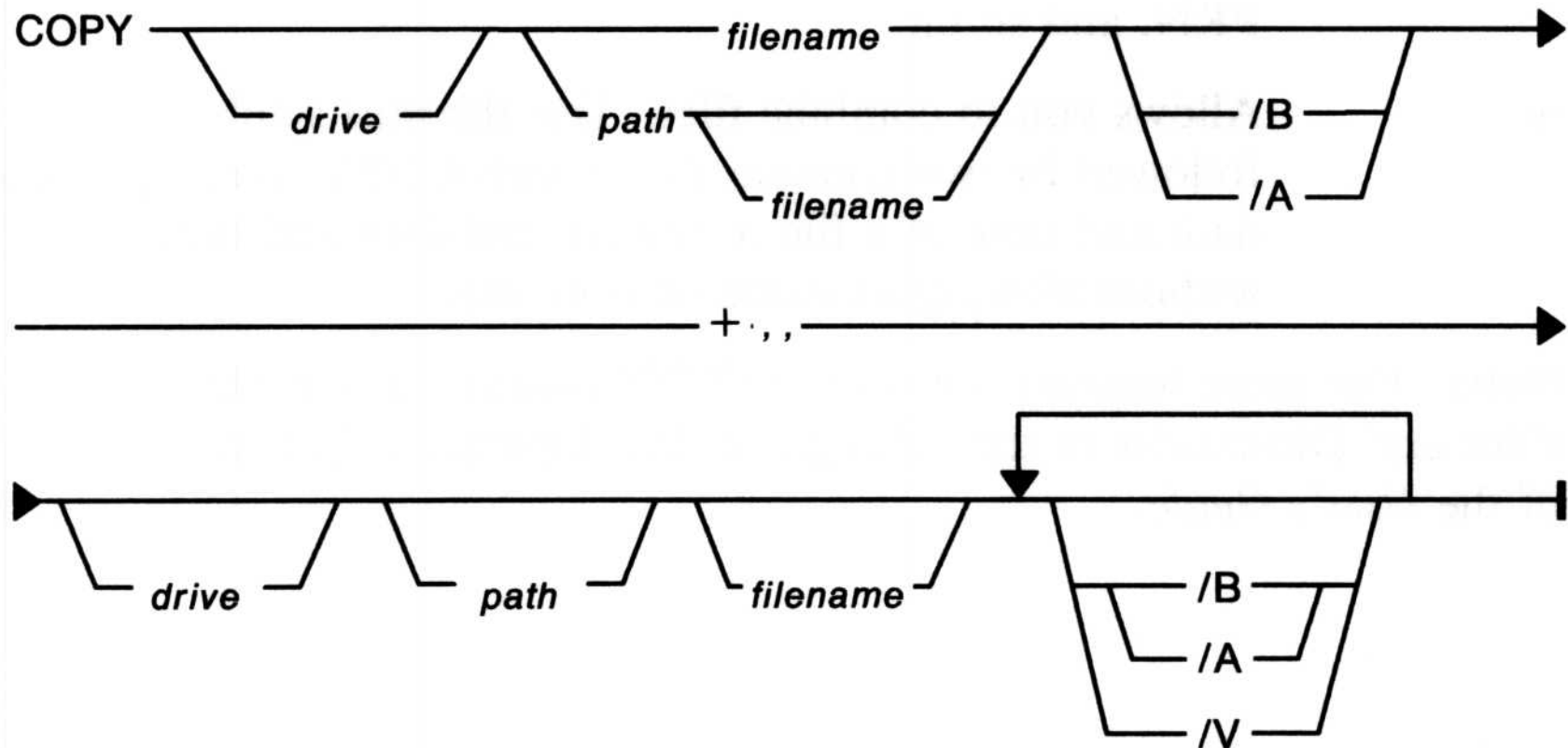
To combine files:



COPY

(Copy or Combine Files)

To change the date and time of a file to the current date and time without altering its name or contents.



Parameters

/B When used with a *source* file name, causes the entire file to be copied, including any end-of-file mark.

When used with a *target* file name, causes no end-of-file character to be added.

/A When used with a *source* file name, causes the file to be treated as an ASCII (text) file. Data in the file is copied up to but not including the first end-of-file mark (Ctrl and Z).

When used with a *target* file name, causes an end-of-file character to be added as the last character of the file.

/V Causes the base operating system to verify that the sectors written on the target disk are recorded properly. Although there are rarely recording errors when you run COPY, this parameter lets you verify that critical data has been correctly recorded. This parameter causes the COPY command to run more slowly because the base operating system must check each entry recorded on the disk. The base operating system displays an error message if a write is not verified.

COPY

(Copy or Combine Files)

filename Specifies the name of a file in any directory or subdirectory specified by PATH. In addition, this can refer to a character device name such as COM1, LPT2, PRN, and so on.

+

Allows you to combine files. Use the plus symbol followed by two commas (+ ,,) with COPY to change the date and time of a file to the current date and time without altering its name or contents.

Note: For more information on the COPY command, see Managing Files and Directories in the “Using the Base Operating System” section of the *User's Guide*.

COUNTRY

(Identify Country Information)

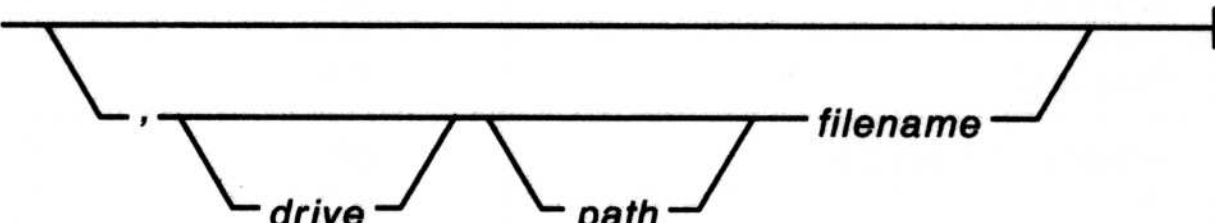
Purpose

Identifies the country for which country-dependent information is selected:

- Date and time format
- Decimal separator
- Character case map table
- Collating sequence table used by SORT
- DBCS environment vector for double-byte characters.

Note: This statement is one of the interrelated CONFIG.SYS statements required for successful code page switching. For a list of these statements, see Managing Country Information in the “Using the Base Operating System” section of the *User's Guide*.

Syntax

COUNTRY = *nnn* 

Parameters

nnn Specifies a three-digit value. For most countries, this value is the three-digit international country code for the telephone system.

The following figure lists the country code and supported code pages for each country. The first code page listed for each country is the system default when a CODEPAGE statement is not specified in the CONFIG.SYS file. The CONFIG.SYS file does not change if the country panel in Presentation Manager is changed.

COUNTRY

(Identify Country Information)

Country	Country Code	Code Pages Supported
Arabic-speaking	785	864 (Note 1), 850
Australia	061	437, 850
Belgium	032	850, 437
Canada (French-speaking)	002	863, 850
Denmark	045	850, 865
Finland	358	850, 437
France	033	437, 850
Germany	049	437, 850
Hebrew-speaking	972	862 (Note 1), 850
Italy	039	437, 850
Japan	081	932 (Note 2), 437
Korea	082	934 (Note 2), 437
Latin America	003	437, 850
Netherlands	031	437, 850
Norway	047	850, 865
Portugal	351	850, 860
Simplified Chinese	086	936 (Note 2), 437
Spain	034	437, 850
Sweden	046	437, 850
Switzerland	041	850, 437
Traditional Chinese	088	938 (Note 2), 437
United Kingdom	044	437, 850
United States	001	437, 850

Notes:

1. This code page is supported only with a country supplement.
2. This code page is supported only with the Asian version of the base operating system on Asian hardware.
3. If your country is not listed, select the most similar country supported.



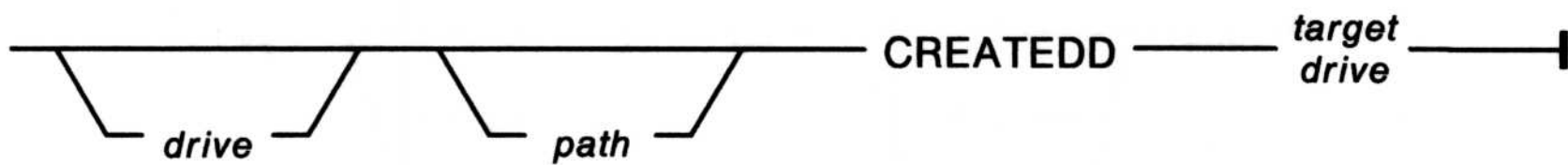
CREATEDD (Create Dump Diskette)

Purpose

Creates a dump diskette.

Note: This command is intended for use with aid from an IBM Service Representative. For more information on CREATEDD, see the "Using the Base Operating System" section of the *User's Guide*.

Syntax



Parameters

target drive Specifies the diskette drive that contains the diskette to be formatted as a dump diskette.

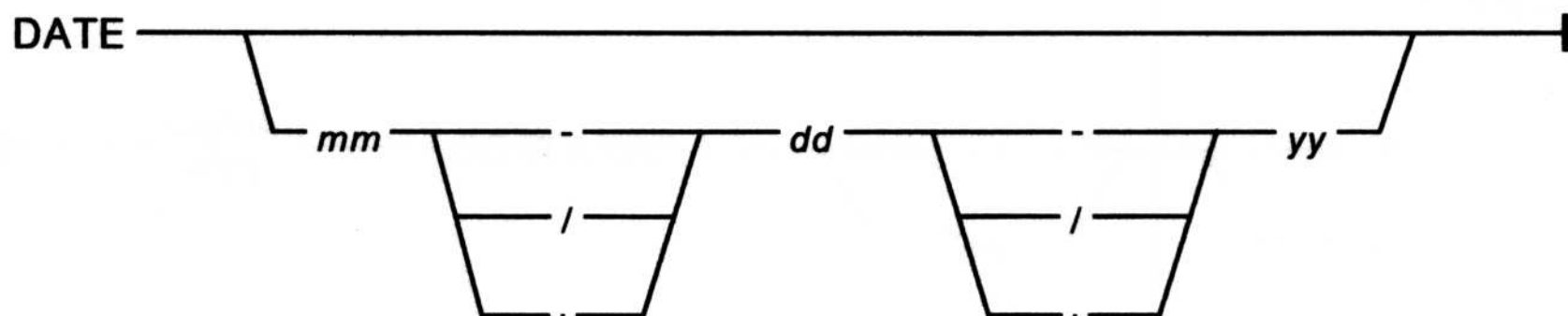
DATE

(Set System Date)

Purpose

Displays or changes the date known to the system and resets the date on your computer clock. This date is recorded in the directory when you create or change a file.

Syntax



Parameters

- mm* Specifies the month that must be entered using only the numbers 1 through 12.
- dd* Specifies the day that must be entered using only the numbers 1 through 31.
- yy* Specifies the year that must be entered using only the numbers 1980 through 2079, or the numbers 00 through 99. The 00 – 79 range represents the years 2000 through 2079, and the 80 – 99 range represents the years 1980 through 1999.

Note: Typing this command without a parameter displays the system date and asks if you want to change it.



DDINSTAL (Install Automated Device Drivers)

Purpose

Provides an automated way to install new device drivers.

Syntax

DDINSTAL _____

Note: Typing this command without a parameter starts a step-by-step menu to install device driver files.

DETACH

(Process Noninteractively)



Purpose

Starts and simultaneously detaches an OS/2 program from its command processor.

Syntax

DETACH _____ *command* _____|

Parameters

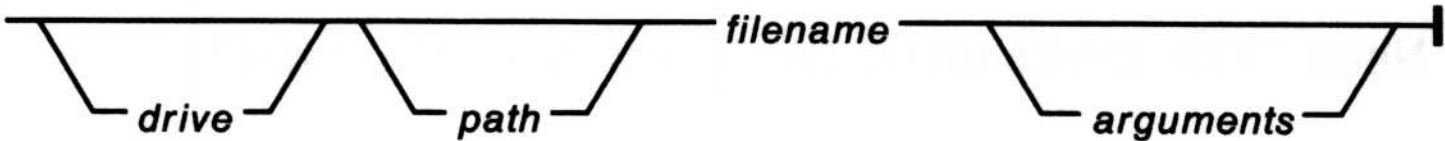
command Specifies any program or OS/2 command and its options that do not require any input from the keyboard or mouse, or output to the display.

DEVICE (Install Device Drivers)

Purpose

Specifies the path and file name of a device driver to be installed in the CONFIG.SYS file.

Syntax

DEVICE = 

Parameters

- filename* The complete name of the file, including its extension, that contains the device driver.
- arguments* Specifies the parameters of the identified device driver.

DEVICE (Install Device Drivers)

ANSI.SYS



Purpose

Allows you to use extended keyboard and display support in DOS mode.

Note: This device driver has no effect on OS/2 mode.

Syntax

DEVICE = _____ ANSI.SYS _____
 drive path

If this **DEVICE** statement is in your **CONFIG.SYS** file when you start the base operating system, the standard input and standard output support is replaced with the extended functions in DOS mode. These functions allow you to redefine keys, manipulate the cursor, and change display color attributes. If you no longer want the extended functions for DOS, you can remove the **DEVICE = C:\OS2\ANSI.SYS** from your **CONFIG.SYS** file and restart the system.

ASYNCCDDx.SYS Device Driver



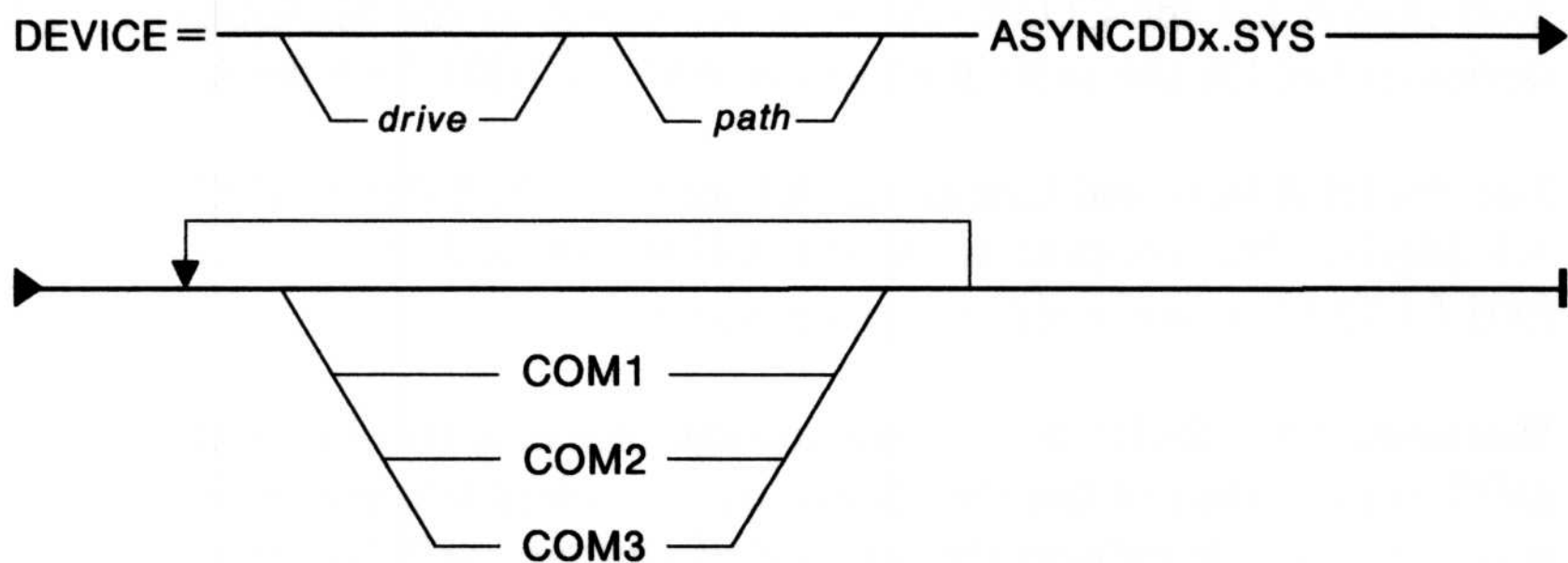
Purpose

Allows OS/2 application programs to use serial devices COM ports through the ACDI. ASCII terminal emulation is an example of an application program that interfaces to serial devices through ACDI.

Warning: If a device driver that supports any one of the serial ports (COM1, COM2, or COM3) is installed by a DEVICE statement that precedes a DEVICE = ASYNCCDDx.SYS statement, the port will be unavailable to ASYNCCDDx.SYS.

The DEVICE = ASYNCCDDx.SYS statement must precede the DEVICE = COM0x.SYS statement in the CONFIG.SYS file or the COM port will be unavailable for use by the device driver.

Syntax



Parameters

COMx For the IBM Personal Computer AT and the IBM Personal Computer XT Model 286 (ASYNCCDDA.SYS), COM1 and COM2 can be specified as parameters. For the supported IBM Personal System/2 models, (ASYNCCDDB.SYS), COM1, COM2, or COM3 can be specified.

DEVICE

(Install Device Drivers)

All devices specified will be loaded for ACDI support. Devices specified will not be available for use by the COM0x.SYS or the MOUSExxx.SYS device drivers.

Note: ASYNCDDA.SYS supports COM1 and COM2 only. ASYNCDDDB.SYS supports COM1, COM2, and COM3.

Examples

To load the ACDI for COM1 on the IBM Personal Computer AT or the IBM Personal Computer XT Model 286:

```
DEVICE = C:\CMLIB\ASYNCDDA.SYS COM1.
```

To load the ACDI support for COM2 and COM3 on an IBM Personal System/2:

```
DEVICE = C:\CMLIB\ASYNCDDDB.SYS COM2 COM3
```

Remarks

You cannot use the COM0x.SYS device driver or the MOUSExxx.SYS device driver for the same port as the ASYNCDDx.SYS device driver.

For the IBM Personal Computer AT and the IBM Personal Computer XT Model 286, you cannot use the ASYNCDDA.SYS and SDLCDD.SYS device drivers at the same time.

Warning: This device driver uses storage that can also be used for DOS mode. Do not use this device driver unless you require its function, since it reduces the amount of storage available to run a OS/2 application.

Related Commands

DEVICE = COM0x.SYS and DEVICE = MOUSExxx.SYS statements in the CONFIG.SYS file.

DEVICE (Install Device Drivers)

COM0x.SYS

Purpose

Allows OS/2 application programs or system programs, such as SPOOL, to use serial devices.

Warning:

- If a device driver that supports any one of serial ports COM1 through COM3 is installed by a DEVICE statement that precedes a DEVICE = C:\OS2\COM0x.SYS statement, the port is unavailable to COM0x.SYS.
- This device driver uses storage that can also be used for DOS mode. Do not use this device driver unless you require its function because it reduces the amount of storage available to run a DOS application.

Syntax

DEVICE =  `COM0x.SYS`

Parameters

- x* Specifies the system file. For the IBM Personal Computer AT and the IBM Personal Computer XT Model 286, this value must be 1. For the supported IBM Personal System/2 (PS/2) models, this value must be 2.

DEVICE

(Install Device Drivers)

Remarks

The COM01.SYS system file is the device driver to use for the IBM Personal Computer AT and IBM Personal Computer XT Model 286. This device driver supports ports COM1 and COM2. The COM02.SYS system file is the device driver to use for the supported IBM Personal System/2 models. This device driver supports ports COM1, COM2, and COM3.

This device driver supports OS/2 applications with the following RS-232-C (serial device) interface functions:

- Duplex communication
- Automatic flow control (XON/XOFF) for both transmit and receive
- Various modem line *handshaking* modes
- Standard and non-standard baud rates
- Receive data error character replacement.

It also supports these system features:

- Multiple active COM ports
- Read and write request queueing
- Interrupt-driven I/O processing
- Internal transmit and receive data buffers
- Several device status query functions.

COM0x.SYS does not provide any support for devices that are attached to the COM port. Instead, it provides enabling support for the asynchronous communications interface itself. OS/2 application programs and system programs provide the support needed to use devices attached to the COM port.

DEVICE (Install Device Drivers)

This device driver allows other device drivers to be installed that support specific COM ports, provided that they are listed *before* COM0x.SYS in the CONFIG.SYS file. This allows support for:

- A special device for use by an application program that is not adequately supported by this device driver
- Additional COM ports beside COM1 through COM3
- Enhanced device driver function for a COM port (may be required for certain communication support).

Device Installation Warning Messages: If a device driver fails to install properly, warning messages are generated during system initialization.

If you are going to use ASCII terminal emulation, refer to page 2-33.

DEVICE (Install Device Drivers)

DFTDD.SYS Device Driver

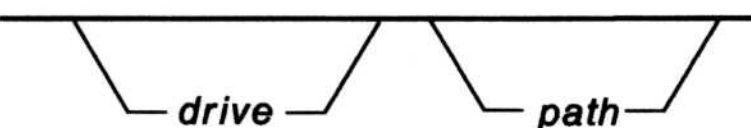



Purpose

Allows Communications Manager 3270 terminal emulation (DFT) to access 3270 Emulation adapter cards for use in communicating with 3274/3174 control units in DFT mode, as well as the System/9370. For the IBM Personal Computer AT and the IBM Personal Computer XT Model 286 use the 3278/79 Emulation Adapter and the Advanced 3278/79 Emulation Adapter. For the supported IBM Personal System/2 models use the IBM 3270 Connection Adapter.

To use this device driver, you must place the following syntax in your CONFIG.SYS file.

Syntax

DEVICE =  DFTDD.SYS 

Examples

DEVICE=C:\CMLIB\DFTDD.SYS

DEVICE (Install Device Drivers)

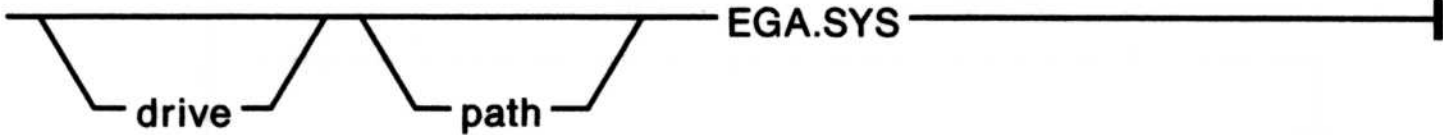
EGA.SYS



Purpose

Supports the Enhanced Graphics Adapter (EGA) register interface.

Syntax

DEVICE =  EGA.SYS

Remarks

The EGA.SYS device driver provides support for the EGA register interface in DOS mode. EGA.SYS must be installed for those application programs that use the EGA register interface.

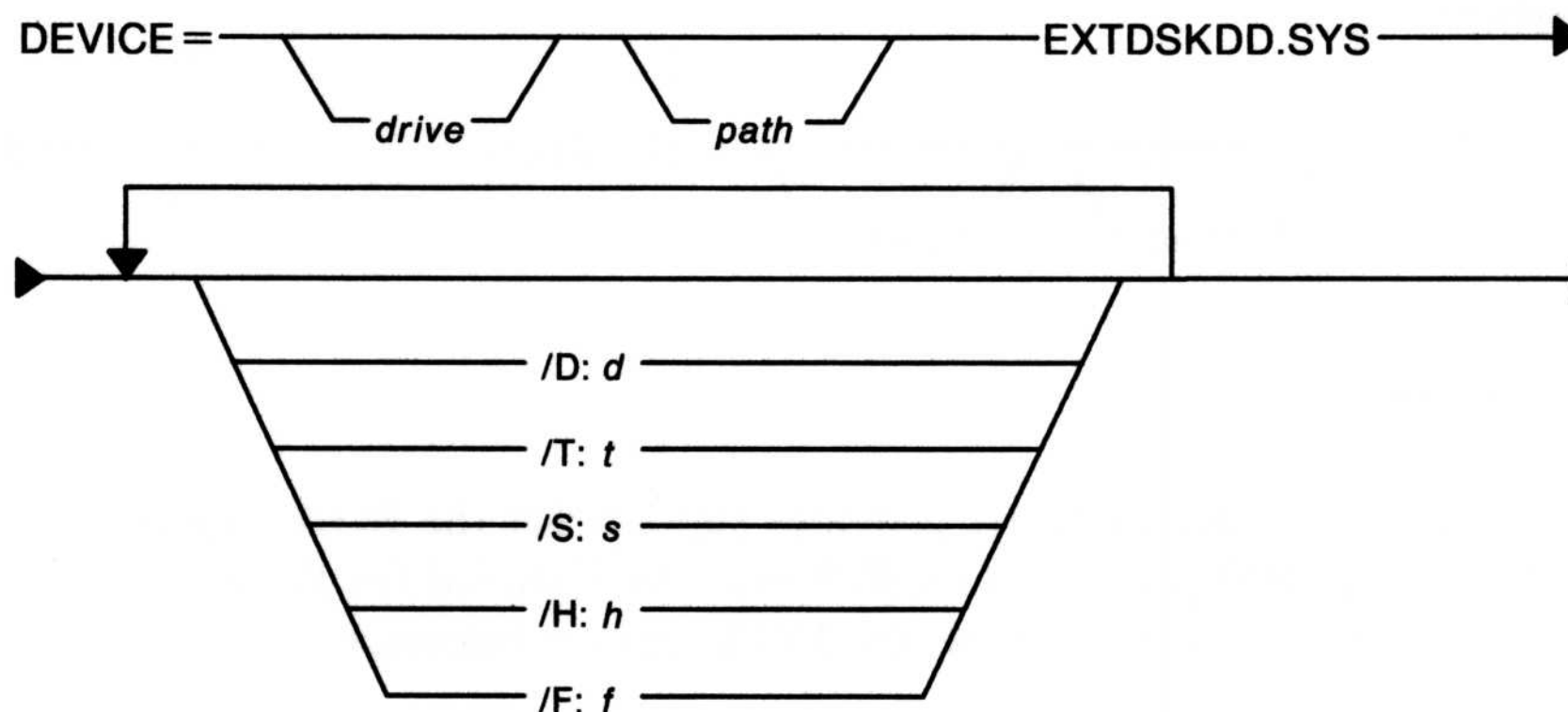
DEVICE (Install Device Drivers)

EXTDSKDD.SYS

Purpose

Allows you to access a disk using a logical drive letter.

Syntax



Parameters

- /D:*d*** Specifies the physical drive number, which can have a value of 0 through 255.
- To specify the first physical diskette drive accessed as A: from the command prompt, specify the value 0. The value 1 specifies the second physical diskette drive, and the value 2 specifies the third physical diskette drive, which must be external.
- /T:*t*** Specifies the number of tracks per side. The valid range is from 1 through 999. The default is 80.
- /S:*s*** Specifies the number of sectors per track. The valid range is from 1 through 99. The default is 9.
- /H:*h*** Specifies the maximum number of heads. The valid range is 1 through 99. The default is 2.

DEVICE (Install Device Drivers)

/F:f Specifies the device type (form factor). Choose from the following list. The default is 2.

Value	Device
0	160/180KB
0	320/360KB
1	1.2MB
2	720KB

Examples

The following examples apply to the IBM Personal Computer AT:

To set up logical drive D for a 720KB external diskette drive when you have one internal diskette drive and one fixed disk, type the following command in the CONFIG.SYS file:

```
DEVICE=C:\OS2\EXTDSKDD.SYS /D:2
```

If, in the preceding example, you want to copy from the external diskette drive to the same drive, type the preceding command in the CONFIG.SYS file twice:

```
DEVICE=C:\OS2\EXTDSKDD.SYS /D:2  
DEVICE=C:-OS2-EXTDSKDD.SYS /D:2
```

The logical drive letters D and E are assigned to the external diskette drive.

Remarks

More than one external device driver can be installed at the same time. The maximum number is the total number of physical diskette drives installed in your system. Installed virtual disks do not affect the drive letter assignments described here if `DEVICE = C:\OS2\VDISK.SYS` statements are listed after `DEVICE = C:\OS2\EXTDSKDD.SYS` statements in the CONFIG.SYS file.

DEVICE (Install Device Drivers)

You can also use EXTDSKDD.SYS to copy from an internal diskette drive to the same drive. Assume that you have a 1.2MB drive as the first diskette drive, a 320/360KB drive as the second diskette drive, and a fixed disk. Type:

```
DEVICE=C:\OS2\EXTDSKDD.SYS /D:0 /T:80 /S:15 /H:2 /C /F:1
```

This assigns the logical drive letter D to the first diskette drive, which can now be referenced as A or D. If, from the default (A) command line, you type:

```
COPY FILE1 D:
```

you are prompted to insert the diskette for the appropriate logical drive, and FILE1 is copied from one diskette to another using only the 1.2MB drive.

Rules for Drive Letters Assigned to Device Drivers: The first physical internal diskette drive is assigned A. The second internal diskette drive is assigned B. Letters from C on are assigned in the order devices or device drivers are encountered. The existence of internal diskette drives and fixed disks is checked first; then the CONFIG.SYS file is checked for device drivers. For the base operating system to recognize an external physical device, the CONFIG.SYS file must have the correct device driver information.

The drive letter B is automatically used, even if there is only one physical drive; thus, a machine with only one diskette drive has two logical diskette drives; A and B. In this case, a specified parameter of /D:1 is an error. The first fixed disk or the first block device driver cannot assign a drive letter lower than C.

For machines with an external drive, if the external device driver is loaded twice, where D:d is the same, this creates two logical drives for the one physical drive, giving you the ability to transfer data from one diskette to another using the same drive.

DEVICE (Install Device Drivers)

This same concept can be applied to internal drives. The base operating system automatically loads a diskette drive device driver for the drive at setup time. If you include a `DEVICE=C:\OS2\EXTDSKDD.SYS` in the CONFIG.SYS file for the same drive, two drive letters are associated with the same drive. The command:

```
DEVICE=C:\OS2\EXTDSKDD.SYS /D:0
```

in the CONFIG.SYS file at startup time causes the base operating system to load another device driver for the first diskette drive. The drive letter depends on the number of diskette drives and fixed disks in the system. For a system with two diskette drives and a fixed disk, the logical drive letter for the first diskette drive is D. With this setup, you can copy files from the first physical diskette drive to the first logical diskette drive by referring to it as A or D.

DEVICE

(Install Device Drivers)

LANDD.SYS



Purpose

Allows OS/2 application programs to use local area network (LAN) communications.

Syntax

DEVICE = _____ LANDD.SYS →
 └─── *drive* ───┬─── *path* ───┘

Parameters

- drive:* The drive where the LANDD.SYS file resides. Specify this drive only if it is not in the default drive.
- path* Path designator that specifies where the file is. Use this if the file is not in the default path.

Remarks

This device driver must be loaded before either the IBM Token-Ring device driver (TRNETDD.SYS) or the IBM PC Network device driver (PCNETDD.SYS) is loaded.

DEVICE (Install Device Drivers)

MOUSExxx.SYS

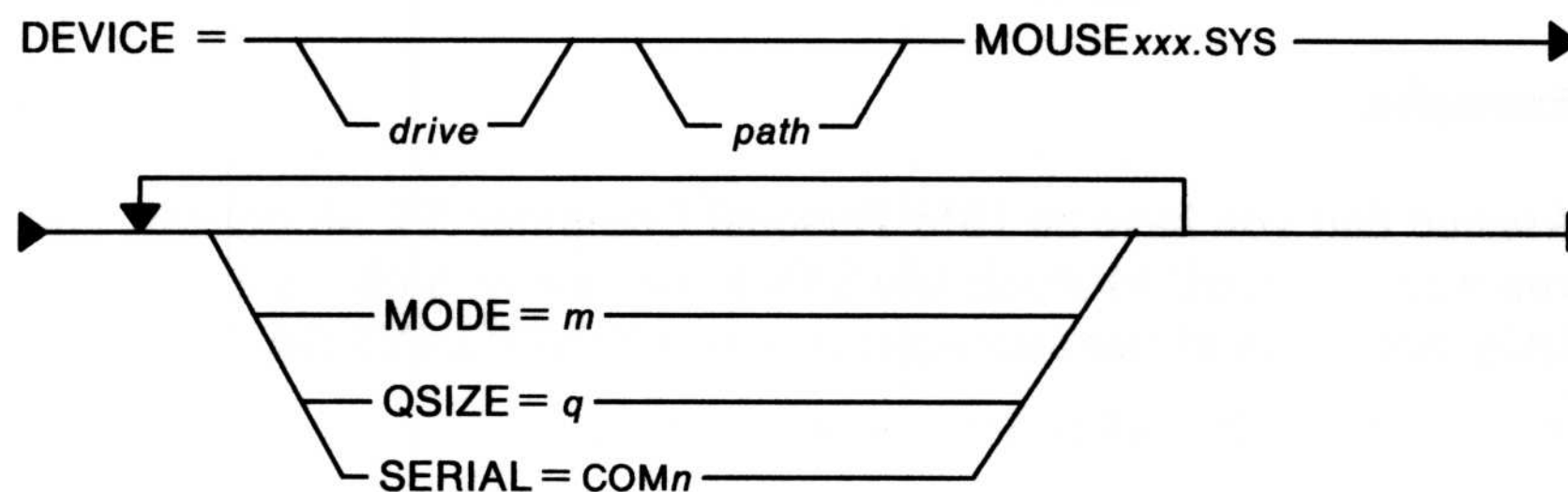
Purpose

Implements support for pointing devices.

Note: To use a mouse, you must also load the POINTDD.SYS device driver.

Warning: DEVICE = C:\OS2\MOUSExxx.SYS statements for serial mouse device drivers must precede any COM0x.SYS statements in the CONFIG.SYS file so that both device drivers are installed correctly.

Syntax



Parameters

MODE = *m*

Informs the mouse device driver if mouse support is for OS/2 mode, DOS mode, or both. The default value is for both. The values indicating the type of support are:

- P = OS/2 mode only
- R = DOS mode only
- B = Both operating modes.

DEVICE (Install Device Drivers)

QSIZE = *q* Specifies the length of the queue for events using all OS/2 mode tasks. The valid range is 1 through 100. The default value is 10 event records.

SERIAL = COM n Specifies the serial communications port that the serial mouse is connected to. The values are COM1 and COM2 for the IBM Personal Computer AT, and COM1 through COM8 for the IBM Personal System/2. The default value is COM1 for a serial mouse.

This parameter is not valid for a parallel mouse (Microsoft Bus Mouse, for example). A COM port connected to a mouse cannot be supported by the COM0x.SYS or the ASYNCDDx.SYS device driver.

Examples

Assume that you have an IBM Personal Computer XT Model 286, and you want to install MOUSEA00.SYS to operate in both OS/2 and DOS modes. Add this statement to your CONFIG.SYS file:

```
DEVICE=C:\OS2\MOUSEA00.SYS MODE=B
```

Assume that you have an IBM Personal Computer AT, and you want to install MOUSEA01.SYS to operate in OS/2 mode only and to use serial port 1. Add this statement:

```
DEVICE=C:\OS2\MOUSEA01.SYS MODE=P SERIAL=COM1
```

Remarks

- Each of the supported devices is supplied with a device driver containing function for both DOS and OS/2 modes.

The mouse device drivers for the IBM Personal System/2, for the IBM Personal Computer AT, and the IBM Personal Computer XT Model 286 are:

MOUSEA05.SYS	IBM Personal System/2 Mouse
MOUSEA00.SYS	PC Mouse by Mouse Systems Serial Mouse (part number 900120-214)

DEVICE (Install Device Drivers)

- MOUSEA01.SYS** Visi-On Mouse Serial Mouse
(part number 69910-1011)
- MOUSEA02.SYS** Microsoft Mouse for IBM Personal
Computers
Serial Mouse (part numbers 039-099, 039-199)
- MOUSEA03.SYS** Microsoft Mouse for IBM Personal
Computers
Parallel Mouse (part numbers 037-099,
037-199)
- MOUSEA04.SYS** Microsoft Mouse for IBM Personal
Computers
In-Port Mouse (part number 037-299)

The mouse device drivers for the supported IBM Personal System/2 models are:

- MOUSEB00.SYS** PC Mouse by Mouse Systems
Serial Mouse (part number 900120-214)
- MOUSEB01.SYS** Visi-On Mouse
Serial Mouse (part number 69910-1011)
- MOUSEB02.SYS** Microsoft Mouse for IBM Personal
Computers
Serial Mouse (part numbers 039-099, 039-199)
- MOUSEB05.SYS** IBM Personal System/2 Mouse

Note: If the jumper on the Microsoft Mouse is set to level 4 and the mouse device driver is loaded, the serial support for COM1 cannot be loaded, because COM1 uses interrupt level 4. Similarly, using interrupt level 3 conflicts with COM2.

DEVICE

(Install Device Drivers)

Assume that you have an IBM Personal System/2 and you want to install MOUSEB02.SYS to work in both modes using COM port 2. You want to use ASCII terminal emulation on COM port 1. You also want to install the COM0x.SYS device driver. Add these statements in the order shown in your CONFIG.SYS file:

```
DEVICE=C:\OS2\POINTDD.SYS  
DEVICE=C:\OS2\MOUSEB02.SYS MODE=B SERIAL=COM2  
DEVICE=C:\CMLIB\ASYNCDDB.SYS COM1  
DEVICE=C:\OS2\COM02.SYS
```

- COM02.SYS must be loaded after MOUSExxx.SYS so that the mouse can gain access to the COM2 port and after ASYNCDDB.SYS so that ACDI can gain access to COM port 1. The preceding procedure dedicates the COM2 port to the mouse. When your system is started, the mouse device driver is loaded and takes the COM2 port, leaving the COM1 and COM3 ports available. ASYNCDDB.SYS then takes COM port 1, leaving COM port 3 available for COM0X.SYS. You then get a message, as the COM02.SYS device driver installs, that tells you the COM2 port did not install. This message is followed by an additional message:

```
The device adapter could not be located.
```

```
The device adapter could not be located.
```

The device adapters are not available because the ports have been claimed by the mouse and the ASYNCDDB.SYS device drivers.

- The POINTDD.SYS device driver is required by the mouse support in Presentation Manager. If this device driver is not installed in the CONFIG.SYS file, an error may occur when a mouse is used in a Presentation Manager window.

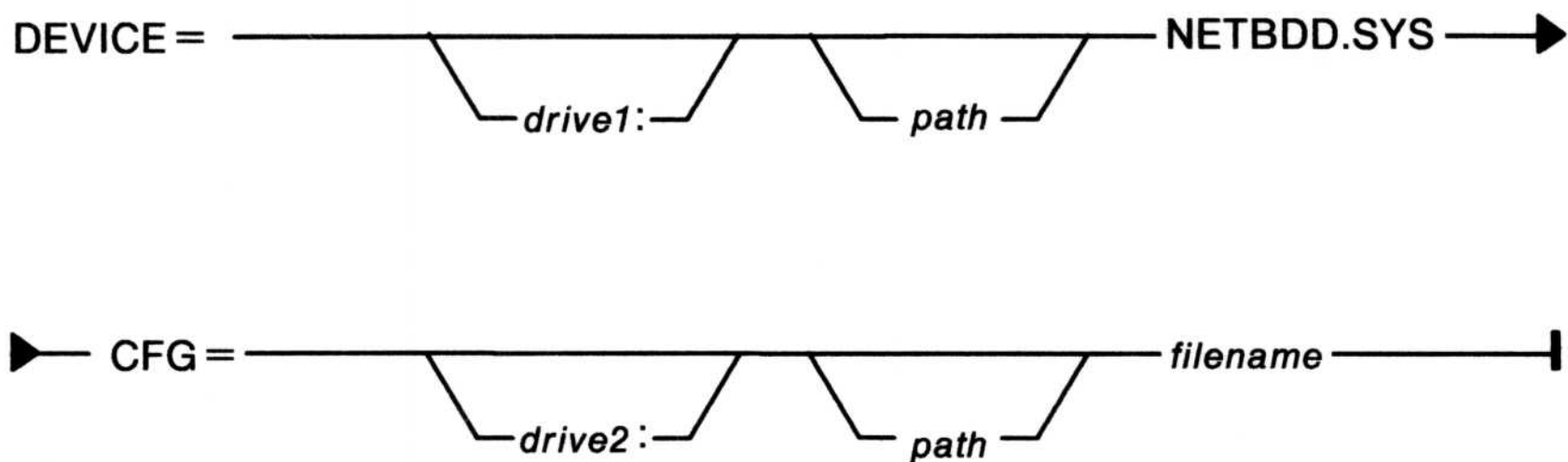
NETBDD.SYS



Purpose

Allows OS/2 application programs to use the NETBIOS application programming interface (API). This command results in the NETBIOS device driver being loaded.

Syntax



Parameters

- drive1:* The drive where the NETBDD.SYS file resides. Specify this drive only if it is not in the default drive.
- drive2:* The drive where the configuration files resides. Specify this drive only if it is not in the default drive.
- path* Path designator that specifies where the file is. Use this only if the file is not in the default path.
- filename* The name of the Communications Manager configuration file that contains the NETBIOS configuration, the IBM Token-Ring adapter and the IBM PC Network adapter interface. If no configuration file name is specified, the adapter initialization continues and the device driver defaults are used. This is not recommended.

DEVICE

(Install Device Drivers)

Remarks

The NETBDD.SYS device driver needs to be loaded after any of the following device drivers:

- LANDD.SYS
- TRNETDD.SYS
- PCNETDD.SYS.

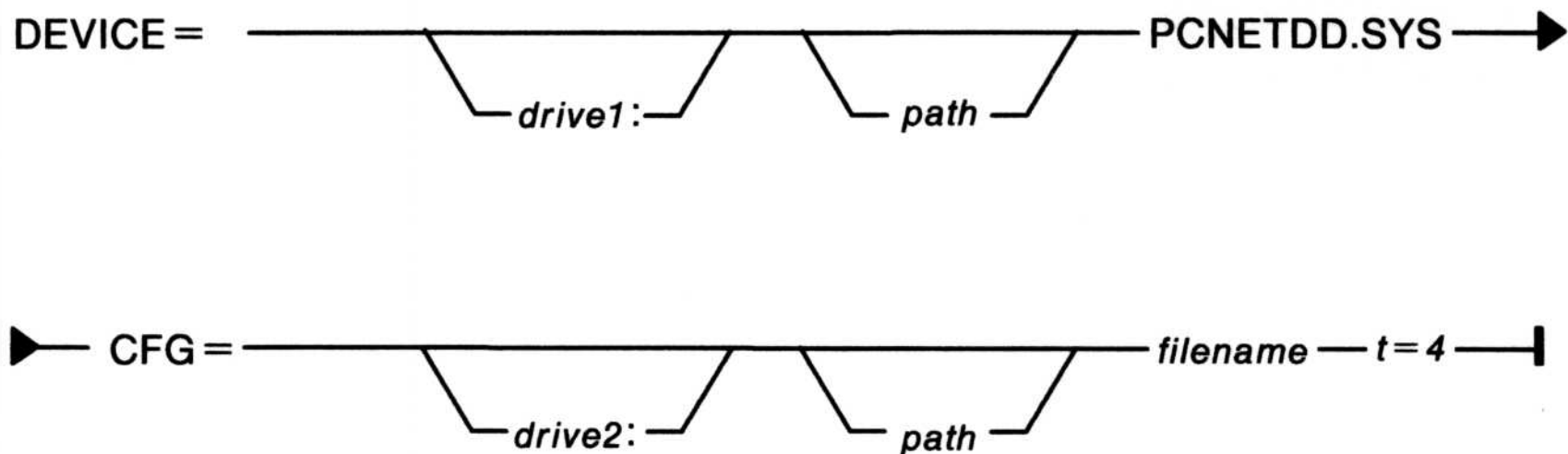
PCNETDD.SYS



Purpose

Allows users to use the IBM PC Network. This command results in the IBM PC Network device driver being loaded whenever the system is started.

Syntax



Parameters

- drive1:* The drive where the PCNETDD.SYS file resides. Specify this only if it is not in the default drive.
- drive2:* The drive where the configuration files resides. Specify this only if it is not in the default drive.
- path* Path designator that specifies where the file is. Use this only if the file is not in the default path.
- filename* The name of the Communications Manager configuration file that contains the IBM PC Network configuration information. If no configuration file name is specified, the adapter initialization continues and the device driver defaults are used. This is not recommended.

DEVICE (Install Device Drivers)

t=4 If specified, indicates that trace is enabled on the configured adapters. For more information about tracing refer to *IBM Local Area Network Technical Reference*.

Remarks

This device driver must be loaded before the NETBDD.SYS device driver is loaded and after the LANDD.SYS device driver.

Related Commands

- ACSPCSYS.EXE
- LANDD.SYS
- NETDD.SYS.

DEVICE (Install Device Drivers)

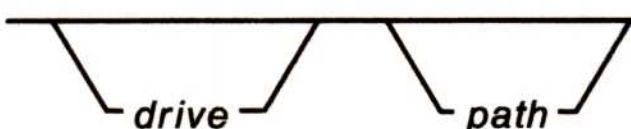
PMDD.SYS



Purpose

Provides pointer draw device driver support for the Presentation Manager.

Syntax

DEVICE =  PMDD.SYS

Examples

The following example shows how the PMDD.SYS device statement looks when it is installed by the Installation Program.

```
DEVICE=C:\OS2\PMDD.SYS
```

Remarks

The `DEVICE = PMDD.SYS` statement is added to your `CONFIG.SYS` file when you install the base operating system using the Installation Program. When the base operating system is started, the `PMDD.SYS` device driver is used to provide pointer draw support for the Presentation Manager.

Note: If the `PMDD.SYS` device statement is removed from your `CONFIG.SYS` file, your system will not restart. To restart, insert the base operating system Installation Program diskette. Press the Escape (Esc) key at the IBM logo screen. Use the instructions in the “Using the Base Operating System” section of the *User's Guide* to add the `PMDD.SYS` statement to your `CONFIG.SYS` file.

DEVICE

(Install Device Drivers)

POINTDD.SYS

Purpose

Provides mouse pointer draw support.

Note: To use a mouse, you must also load this device driver, in addition to loading MOUSExxx.SYS.

Syntax

DEVICE =  POINTDD.SYS

Remarks

In OS/2 mode, text modes 0, 1, 2, 3, and 7 (and the + and * variations of these modes) are the only modes supported. In DOS mode, graphic modes 4, 5, 6, D, E, F, and 10 are also supported. For graphic modes D, E, F, and 10, the EGA.SYS device driver must also be installed, and the DOS application program to be used must support the EGA register interface.

POINTDD.SYS provides draw support in all text modes for both DOS and OS/2 modes; it draws in graphic modes for CGA, EGA, and VGA displays for DOS mode. POINTDD.SYS tracks only (provides mode information) for all advanced function modes on the 8514A for both DOS and OS/2 modes; it tracks only for CGA, EGA, and VGA graphic modes for OS/2 mode.

Note: The POINTDD.SYS device driver is required by the mouse support in Presentation Manager. If this device driver is not installed in the CONFIG.SYS file, an error may occur when a mouse is used in a Presentation Manager window.

SDLCDD.SYS Device Driver



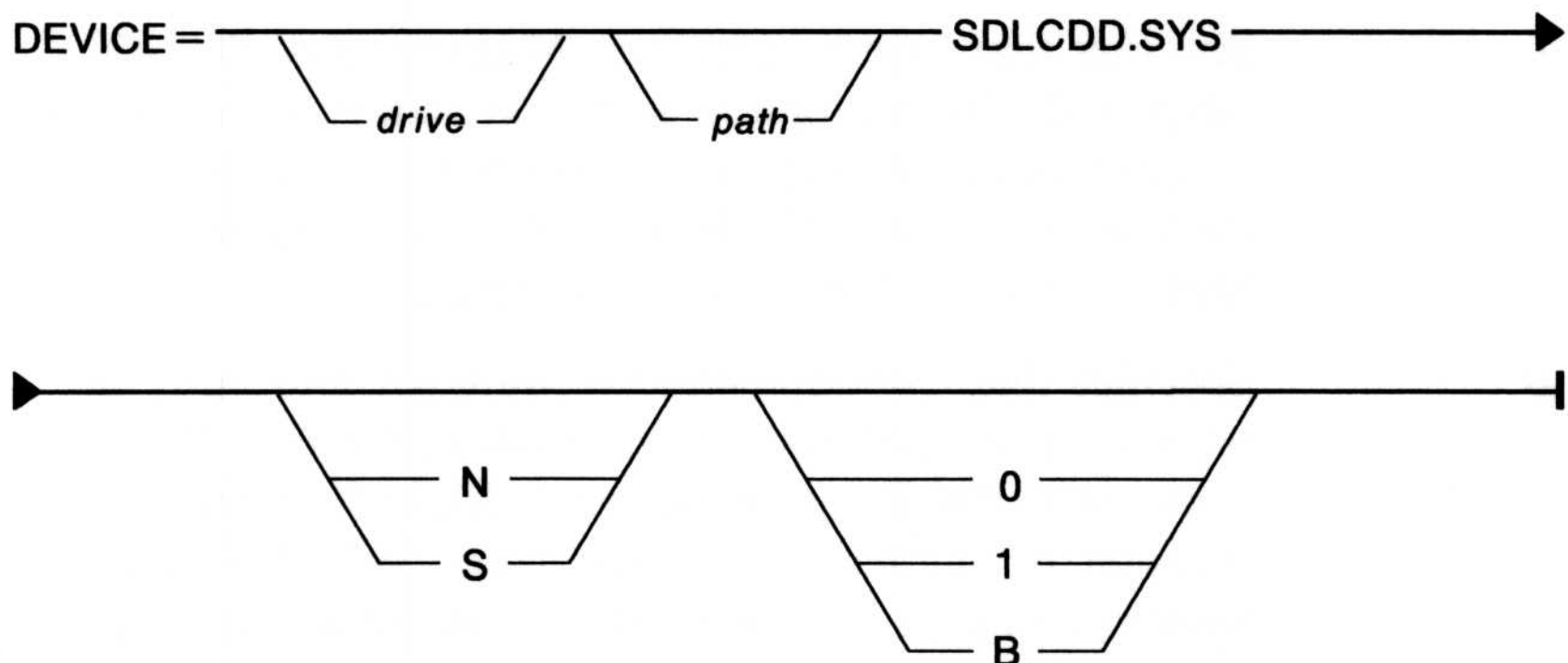
Purpose

Allows OS/2 application programs or system programs (such as Communications Manager 3270 terminal emulation) to use SDLC (Synchronous Data Link Control) devices for communications.

Warning: For the IBM Personal Computer AT and the IBM Personal Computer XT Model 286, you cannot use the SDLCDD.SYS device driver at the same time as either the serial MOUSExxx.SYS device driver or the ASYNCDDA.SYS device driver.

If you need to use either of the above (for example, 3270 terminal emulation (SDLC) and ASCII terminal emulation), you must maintain separate copies of the CONFIG.SYS file; one for each device driver. The system must be restarted (press the Ctrl and Alt and Del keys) after the appropriate CONFIG.SYS file is placed in the root directory.

Syntax



DEVICE

(Install Device Drivers)

Parameters

- N* Specifies that the interrupt levels for SDLC are non-shareable. This value must be used for the IBM Personal Computer AT and the IBM Personal Computer XT Model 286, and is recommended for the supported IBM Personal System/2 models if the `DEVICE = ASYNCDDx.SYS` statement is not coexisting in the same `CONFIG.SYS` file. If no value is specified, *N* is the default.
- S* Specifies that the interrupt levels for SDLC are shared. For the supported IBM Personal System/2 models, this value is necessary in order to use SDLC and ASCII terminal emulation simultaneously. Specifying this value can affect SDLC performance.
- 0* Specifies that Communications Manager uses the SDLC adapter that is configured as the primary adapter on an IBM Personal Computer AT or an IBM Personal Computer XT or is configured as SDLC 1 on an IBM Personal System/2. This value is the only valid value for the IBM Personal Computer AT and the IBM Personal Computer XT Model 286. If no value is specified, 0 is the default.
- 1* Specifies that Communications Manager uses the SDLC adapter that is configured as the secondary (or alternate) adapter on an IBM Personal Computer AT or an IBM Personal Computer XT Model 286 or is configured as SDLC 2 on an IBM Personal System/2.
- B* Specifies that Communications Manager uses both the primary and secondary (or alternate) adapters. When using two IBM Personal System/2 Multi-Protocol Adapter/A adapter cards in the supported IBM Personal System/2 models, you must specify *B*. One card must be set up as SDLC 1 (primary adapter) and the other as SDLC 2 (secondary adapter.)

DEVICE (Install Device Drivers)

Examples

The following statement shows how to install the SDLC device driver on an IBM Personal Computer AT or an IBM Personal Computer XT Model 286.

```
DEVICE=C:\CMLIB\SDLCDD.SYS
```

Note: The default values of *N* and *O* are assumed.

The following statement shows how to install the SDLC device driver for a primary adapter, non-shareable interrupts on the supported IBM Personal System/2 models.

```
DEVICE=C:\CMLIB\SDLCDD.SYS N 0
```

DEVICE

(Install Device Drivers)

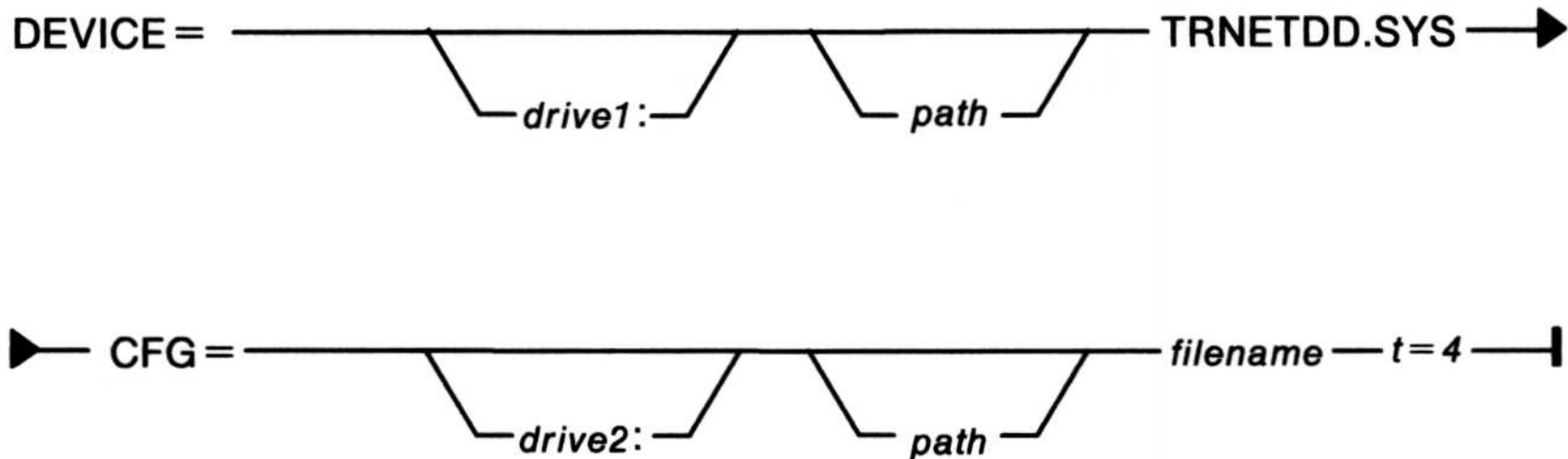
TRNETDD.SYS



Purpose

Allows users to use the IBM PC Network. This command results in the IBM PC Network device driver being loaded whenever the system is started.

Syntax



Parameters

- drive1:* The drive where the TRNETDD.SYS file resides. Specify this only if it is not in the default drive.
- drive2:* The drive where the configuration files resides. Specify this only if it is not in the default drive.
- path* Path designator that specifies where the file is. Use this only if the file is not in the default path.
- filename* The name of the Communications Manager configuration file that contains the configuration for the IBM Token-Ring configuration information. If no configuration file name is specified, the adapter initialization continues and the device driver defaults are used. This is not recommended.

DEVICE (Install Device Drivers)

t=4 If specified, indicates that trace is enabled on the configured adapters. For more information about tracing, see *IBM Local Area Network Technical Reference*.

Remarks

This device driver must be loaded before the NETBDD.SYS device driver is loaded and after the LANDD.SYS device driver.

Related Commands

- ACSTRSYS.EXE
- LANDD.SYS
- NETDD.SYS

DEVICE (Install Device Drivers)

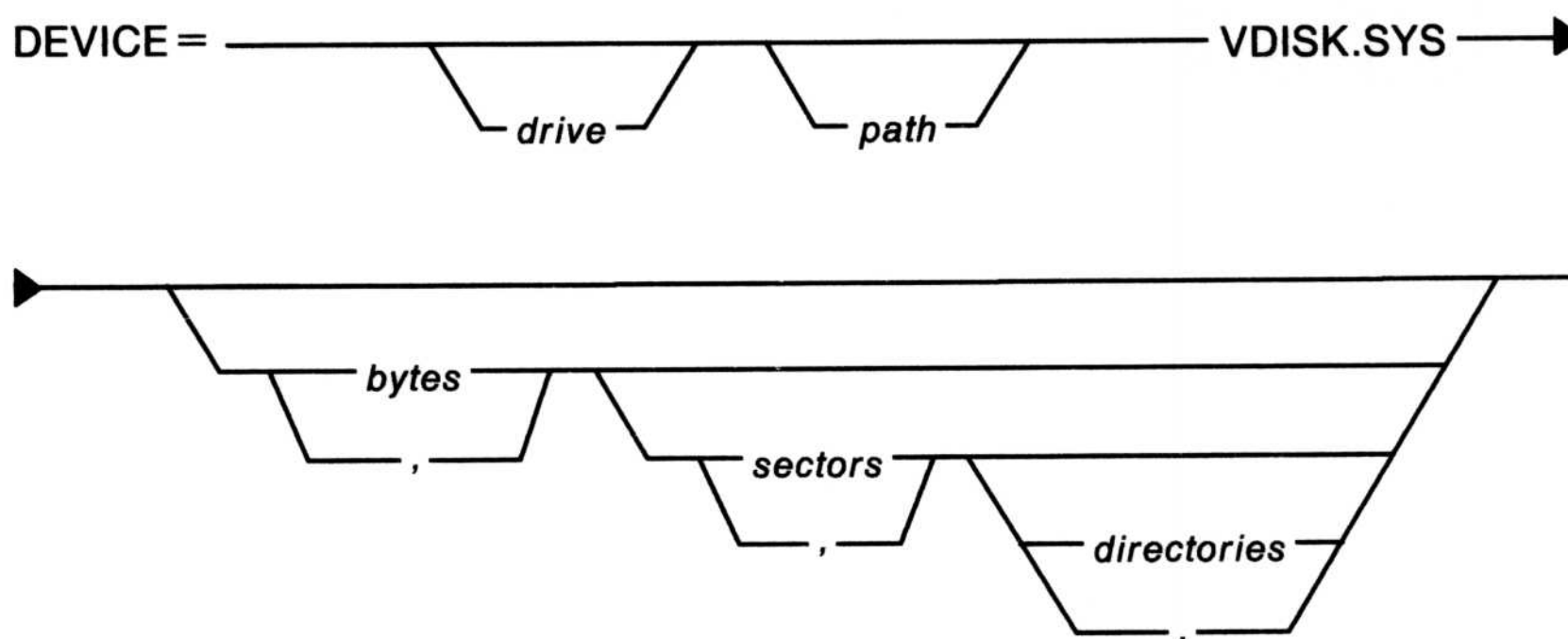
VDISK.SYS

Purpose

Installs a virtual disk.

Warning: `DEVICE = C:\OS2\VDISK.SYS` statements must be listed *after* any `DEVICE = C:\OS2\EXTDSKDD.SYS` statements in the `CONFIG.SYS` file to avoid affecting logical drive letter assignments.

Syntax



Parameters

- bytes* Specifies the size of VDISK in kilobytes. The valid range is 16 through 4096. The default value is 64.
- sectors* Specifies the sector size in bytes. The values are 128, 256, 512, and 1024. The default value is 128.

DEVICE (Install Device Drivers)

directories Specifies the number of directory entries. The valid range is 2 through 1024. The default value is 64.

Note: The FAT file system cannot support a root directory containing more than 255 sectors. For example, a 64KB disk with 128-byte sectors and 1024 directory entries requires 256 sectors. For 128-byte sectors, the maximum number of root directory entries you should specify is 1020.

, (*comma*) Specifies that you do not want to alter the default value of the parameter it supplements. This tells the system to skip past the value it is looking for at this position, install the default value, and proceed to the next value.

Note: The parameters *bytes*, *sectors* and *directories* are positional parameters. This means that if a parameter is omitted but the one that follows it is specified, the omitted parameter's position must be designated by a comma. However, if all the parameters are specified, blank spaces can be used as separators.

Examples

The following example shows how to install a 160KB virtual disk with 128-byte sectors and 64 directory entries. Note that all parameters are specified.

```
DEVICE=C:\OS2\VDISK.SYS 160 128 64
```

If you decide to use the default value of any one of the parameters, such as not specifying the sector size, you can do this by not entering its value and substituting a comma where the value would be. The preceding command can be typed as follows:

```
DEVICE=C:\OS2\VDISK.SYS 160,,64
```

VDISK displays an informative message to tell you that it is installed, and gives you the drive letter that is assigned to the virtual disk.

Assume that you want to specify a value for *directories* and accept the defaults for *bytes* and *sectors*, the value for *directories* should be preceded by two commas, as in this example:

```
DEVICE=C:\OS2\VDISK.SYS,,32
```

DEVICE

(Install Device Drivers)

Remarks

The value of *directories* is rounded up to the nearest sector size boundary. For example, if you give a value of 25, and your sector size is 512 bytes, 25 is rounded up to 32, which is the next multiple of 16. (There are sixteen 32-byte directory entries in 512 bytes.)

If the virtual disk size specified is too large to fit in storage, VDISK tries to make a 16KB virtual disk. This may result in a virtual disk with a different number of directory entries than was specified by *directories*.

DEVINFO (Prepare Code Page Devices)

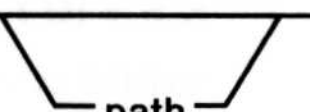
Purpose

Prepares a device for system code page switching.

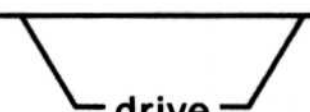
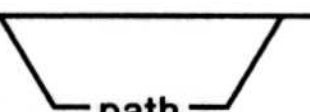
Note: This statement is one of the interrelated CONFIG.SYS statements required for successful code page switching. For a list of these statements, see Managing Country Information in the "Using the Base Operating System" section of the *User's Guide*.

Syntax

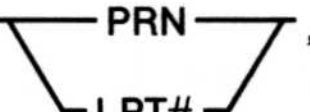
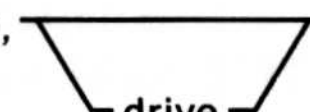
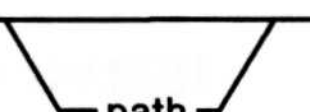
Keyboards:

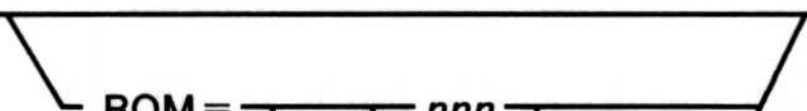

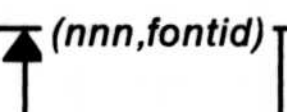
DEVINFO = KBD ,layout,   filename

Displays:

DEVINFO = SCR ,device,   filename

Parallel Printers:

DEVINFO =  PRN ,device,   filename

 ,ROM =  nnn  (nnn,fontid)

DEVINFO

(Prepare Code Page Devices)

Parameters

Keyboards:

layout Specifies the keyboard layout. See the **KEYB** command on page 2-89 for keyboard layout values and additional subcountry information. If you use an IBM enhanced keyboard from France, Italy, or the United Kingdom, see Managing Country Information in the "Using the Base Operating System" section of the *User's Guide* for more information and examples.

filename Specifies the complete name of the file, including its extension, that contains the keyboard translation tables. The name of the system file is **KEYBOARD.DCP** and can be found in the **C:\OS2** subdirectory by typing the path statement **C:\OS2\KEYBOARD.DCP**. There is no default.

Displays:

device Specifies the reserved device name. The allowable device names are:

IBMCGA.DLL IBM Color Graphics Adapter

IBMEGA.DLL IBM Enhanced Graphics Adapter

IBMVGA.DLL IBM Personal System/2 Video
Graphics Adapter

IBMVGA.DLL IBM Personal System/2 Display
Adapter

IBMVGA.DLL IBM Personal System/2 8514A

IBMBGA.DLL IBM Personal System/2 8514/A with
memory expansion

filename Specifies the complete name of the file, which includes its extension, that contains the system code pages for the display. The name of the system file is **VIOTBL.DCP** and can be found in the **C:\OS2** subdirectory by typing the path statement **C:\OS2\VIOTBL.DCP**. There is no default.

DEVINFO

(Prepare Code Page Devices)

Parallel Printers:

LPT#	<i>The #</i> specifies that the value is printer number 1, 2, or 3.
<i>device</i>	Specifies the physical device number. The allowable device numbers are: 4201 IBM Proprinter 5202 IBM Quietwriter III
<i>filename</i>	Specifies the complete name of the file, which includes its extension, that contains the system code pages for the printer. The name of the system file is the device number of the printer with DCP as its extension. For example, the file name to specify for the IBM Proprinter is 4201.DCP and can be found in the C:\OS2 subdirectory by entering the path statement C:\OS2\4201.DCP. There are no defaults.
ROM	Specifies that system code pages are available in device read-only storage or cartridge.
<i>nnn</i>	Represents a value for a system code page that is available in the printer device read-only storage. A printer device may support more than one system code page.
<i>fontid</i>	Specifies the font identification number to be associated with the system code page. A printer may support more than one system code page. One system code page may have more than one font (a set of typographic characters all of one size and style) associated with it.

Notes:

1. For the IBM Quietwriter III, the *fontid* is a three-digit number on the label of the cartridge. The *fontids* for the imbedded code page 437 font are 011, 085, 159, and 254.
2. The IBMCGA.DLL support file is shipped on the OS/2 Device Support Supplement diskette.

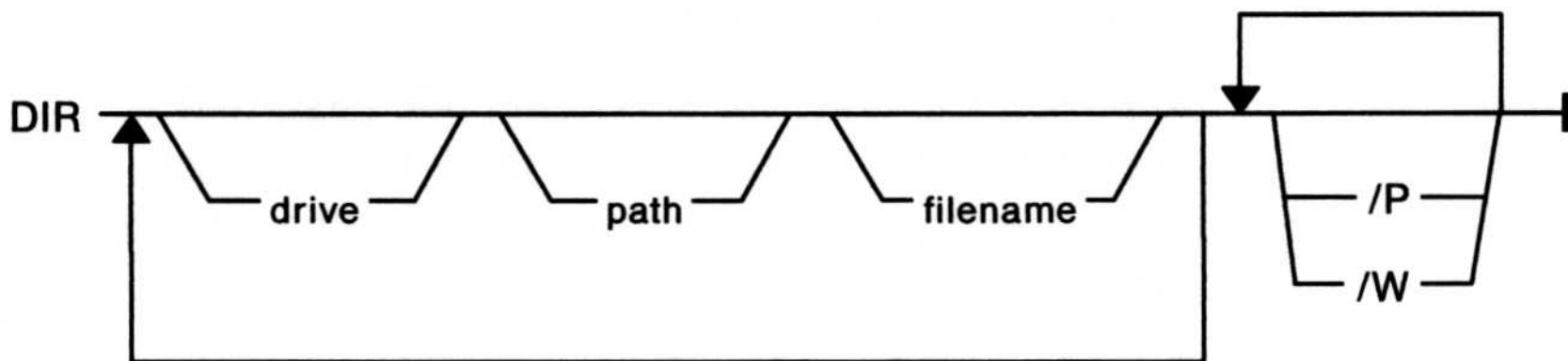
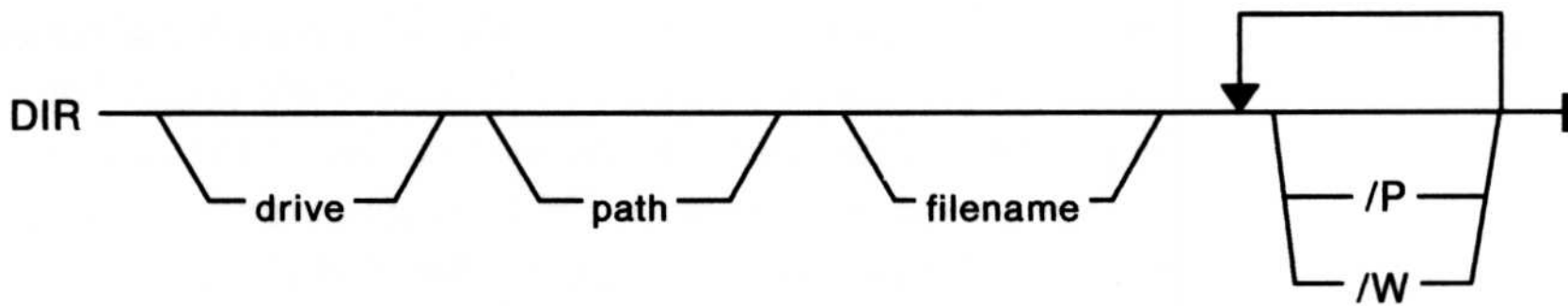
DIR

(Display Files in Directory)

Purpose

Lists the files in a directory.

Syntax



Parameters

/P Pauses the display of directory entries after the screen is filled. To continue the display of output, press any key.

/W Selects wide display. In DOS mode, five file names per line are shown without other file information. In OS/2 mode, five names are shown for 80-character width display mode and two names are shown for 40-character width display mode.

Note: Typing this command without a parameter lists the files in the current directory.

DISKCACHE

(Allocate Storage Blocks)

Purpose

Specifies the number of blocks of storage to allocate for control information and for use by the disk cache.

Syntax

DISKCACHE = _____ *n* _____|

Parameters

n Specifies a number from 64 through 7200 that indicates the number of 1024-byte blocks of storage to use for control information and for use by the disk cache.

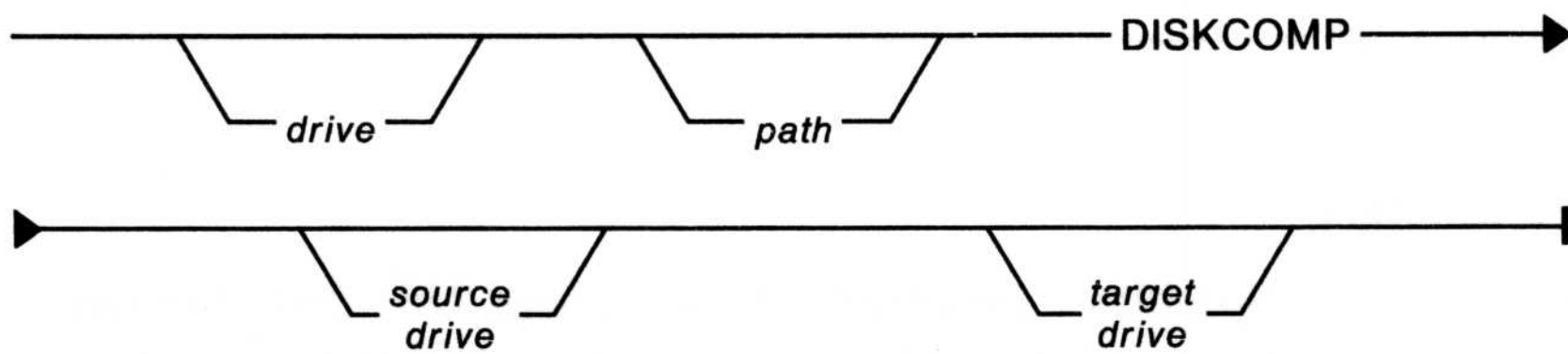
DISKCOMP

(Compare Diskettes)

Purpose

This external command compares the contents of the diskette in the source drive to the contents of the diskette in the target drive.

Syntax



Notes:

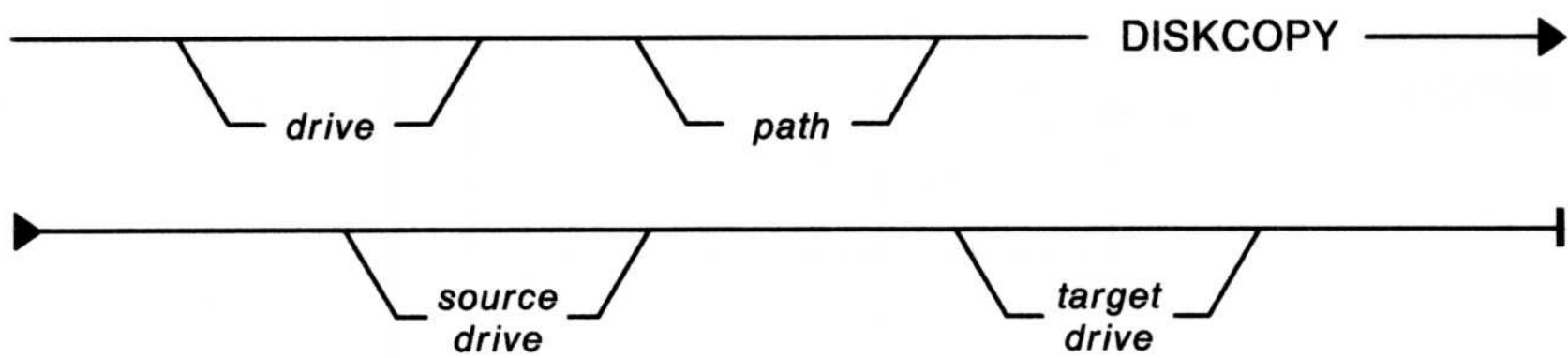
1. DISKCOMP does not work in DOS mode on drives involved with ASSIGN, JOIN, or SUBST.
2. Typing this command without a parameter starts a step-by-step menu to compare diskettes in a diskette drive.

DISKCOPY (Copy Diskettes)

Purpose

This external command copies the contents of the diskette in the source drive to the diskette in the target drive. If necessary, the target diskette is formatted during the copy.

Syntax



Notes:

1. DISKCOPY does not work in DOS mode on drives involved with ASSIGN, JOIN, or SUBST.
2. Typing this command without a parameter starts a step-by-step menu to copy diskettes in a diskette drive.

DPATH

(Set Search Path)



Purpose

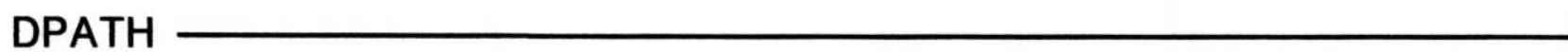
Gives application programs the search path to data files outside the current directory.

Syntax

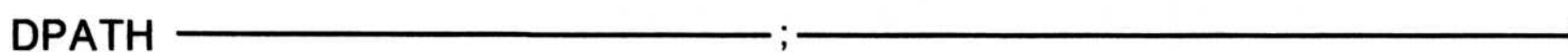
To specify paths to data files outside the current directory:



To display paths to data files currently in effect:



To delete usage of the DPATH command:



Note: In OS/2 mode, DPATH can be specified with the SET command. For more information on using the SET command, see page 2-128.

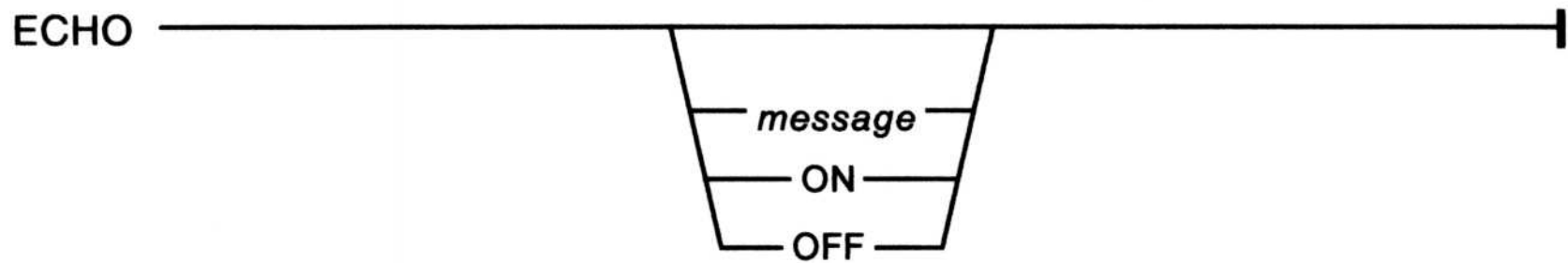
ECHO

(Display Commands as They Process)

Purpose

Allows or prevents the screen display of OS/2 commands run from a batch file. It does not interfere with messages produced while commands are running.

Syntax



Parameters

message Displays *message* regardless of the current ON or OFF state. In this way, you can display specific messages even when ECHO has been turned off.

ON Displays all commands as they are run. This is the default value.

OFF Stops the display of commands on the screen (including the REM command).

Note: Typing this command without a parameter displays its current status of on or off.

ENDLOCAL **(Restore Environment Variables)**



Purpose

Restores the drive, directory, and environment variables that were in effect before the SETLOCAL command ran.

Syntax

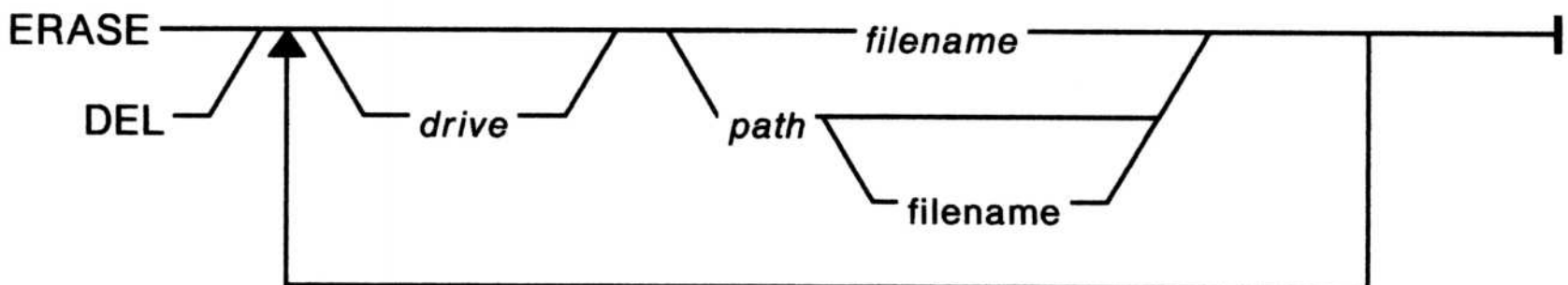
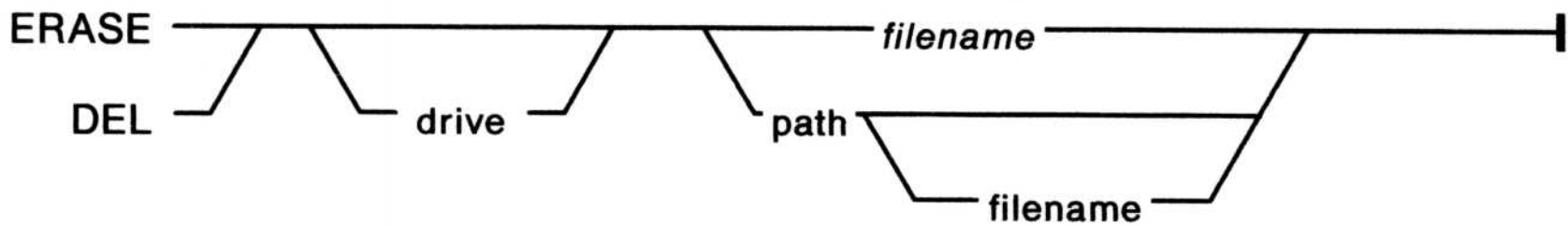
ENDLOCAL

ERASE or DEL (Delete Files)

Purpose

Deletes one or more files. Use ERASE or DEL interchangeably.

Syntax



Warning: You cannot use ERASE or DEL to delete a subdirectory; use the remove directory (RD/RMDIR) command. In addition, read-only and hidden files, such as the operating system files of IBMBIO.COM and IBMDOS.COM, cannot be deleted.

EXIT (End a Command Processor)

Purpose

Ends the current command processor (CMD or COMMAND) and returns to the previous one, if one exists.

Syntax

EXIT

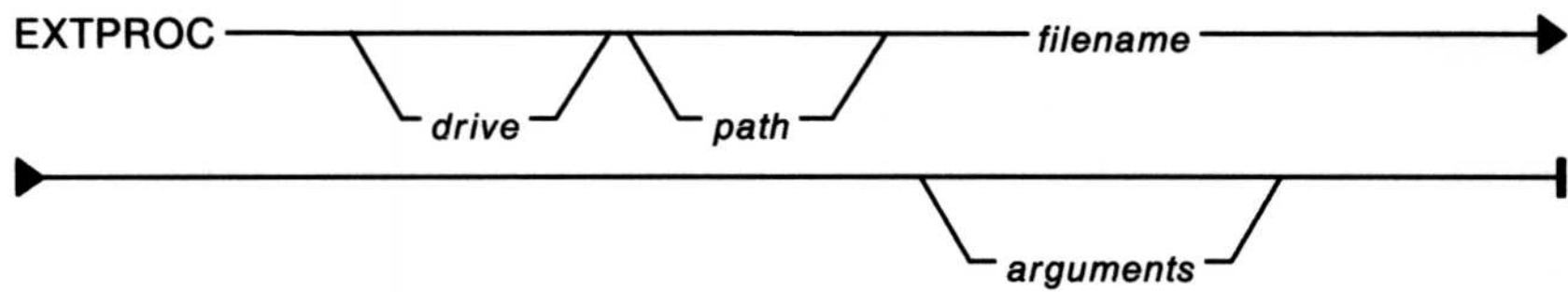


EXTPROC (Define External Batch Processor)

Purpose

Defines an external batch processor for a batch file.

Syntax



Parameters

filename

The complete name of the file, including its extension, that contains the external batch processor.

arguments

Information passed to the external batch processor.

FCBS (Use File Control Blocks)



Purpose

Determines file control blocks (FCBs) management information for DOS mode.

Note: This statement has no effect on OS/2 mode.

Syntax

FCBS = _____ *m,n* _____

Parameters

- m* Specifies a number from 1 through 255 that indicates the total number of file control blocks (FCBs) that can be open at the same time.
- n* Specifies the number of files opened by FCBs that cannot be closed automatically by the system when a program tries to have more than *m* files opened by FCBs at one time. The first *n* files opened by FCBs are protected from being closed. The range of values for *n* is from 0 through 255. The value for *n* must be less than or equal to *m*.



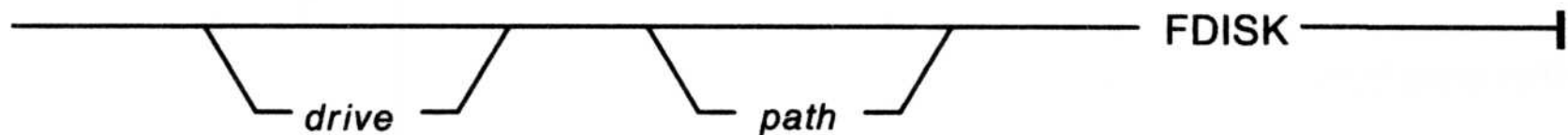
FDISK (Manage Partitions)

Purpose

This external command allows you to:

- Create a primary partition
- Create an extended partition
- Create a logical drive in the extended partition
- Change the active partition
- Delete a primary partition
- Delete an extended partition
- Delete a logical drive in the extended partition
- Display partition data
- Display logical drive information
- Select the next fixed-disk drive for partitioning (for systems with more than one fixed disk).

Syntax



Warning: Any primary partitions or logical drives you create with FDISK must be formatted before you can use them.

Note: Typing this command without a parameter starts a step-by-step menu to run the FDISK utility.

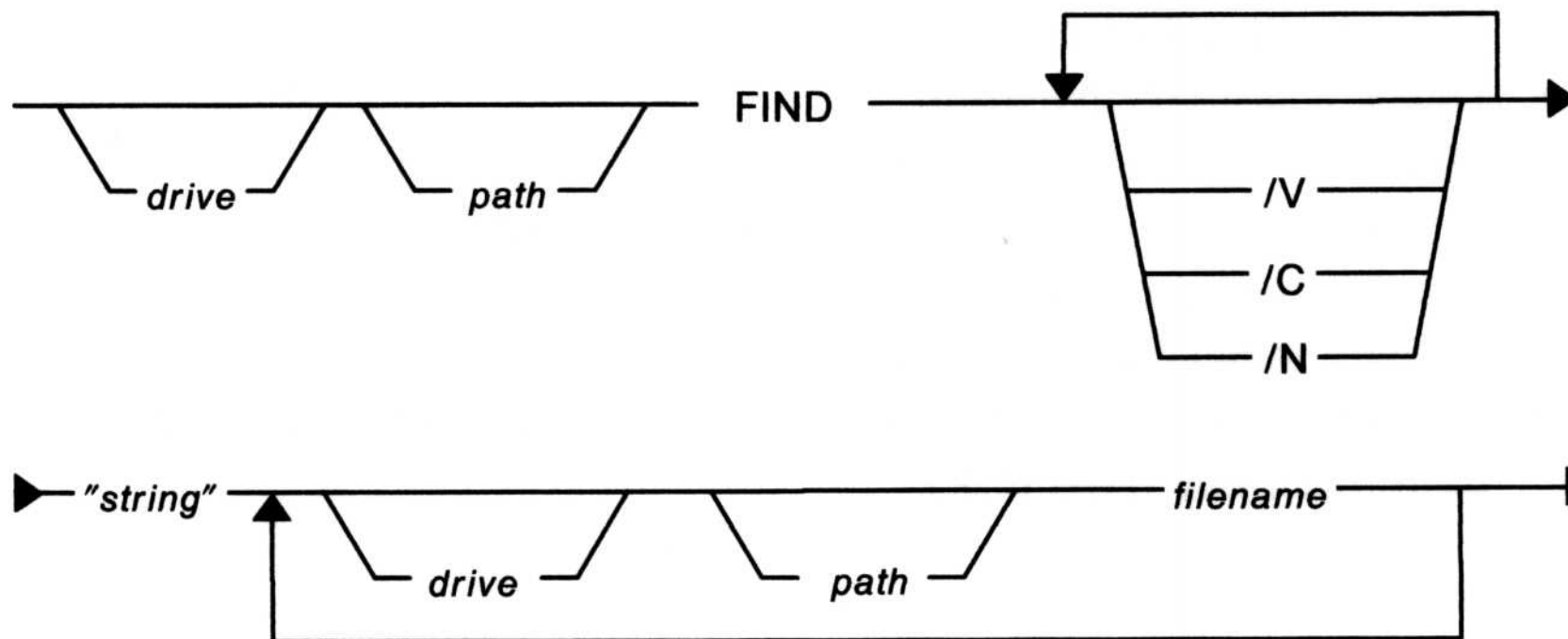
FIND

(Search Files for Strings of Text)

Purpose

This external command searches for a specific string of text in a file or files.

Syntax



Parameters

- /V** Displays all lines *not* containing the specified string.
- /C** Displays the count of lines that contain a match in each of the files. If /C is specified with /N, FIND ignores /N. If /C is specified with /V, FIND returns the count of lines *not* containing the string.
- /N** Puts the relative line number in front of each matching line.
- string** Specifies the string to search for. Remember to enclose the string in double quotes. An uppercase string does not match a lowercase string.

FOR

(Process Commands Repetitively)

Purpose

Allows repetitive running of OS/2 commands.

Syntax

To use FOR from the OS/2 command prompt:

```
FOR variable IN (set) DO command
```

To use FOR from a batch file:

```
FOR %%c IN set DO command
```

Parameters

variable Specifies any character, including 0 through 9, entered at the command prompt.

%%*c* Specifies any character, including 0 through 9. Note that this does not conflict with the %0 through %9 batch parameter substitution because two % symbols (%%) are required with batch processing when doing the FOR loop substitution.

set Specifies one or more items.

command Specifies the OS/2 command to be run.

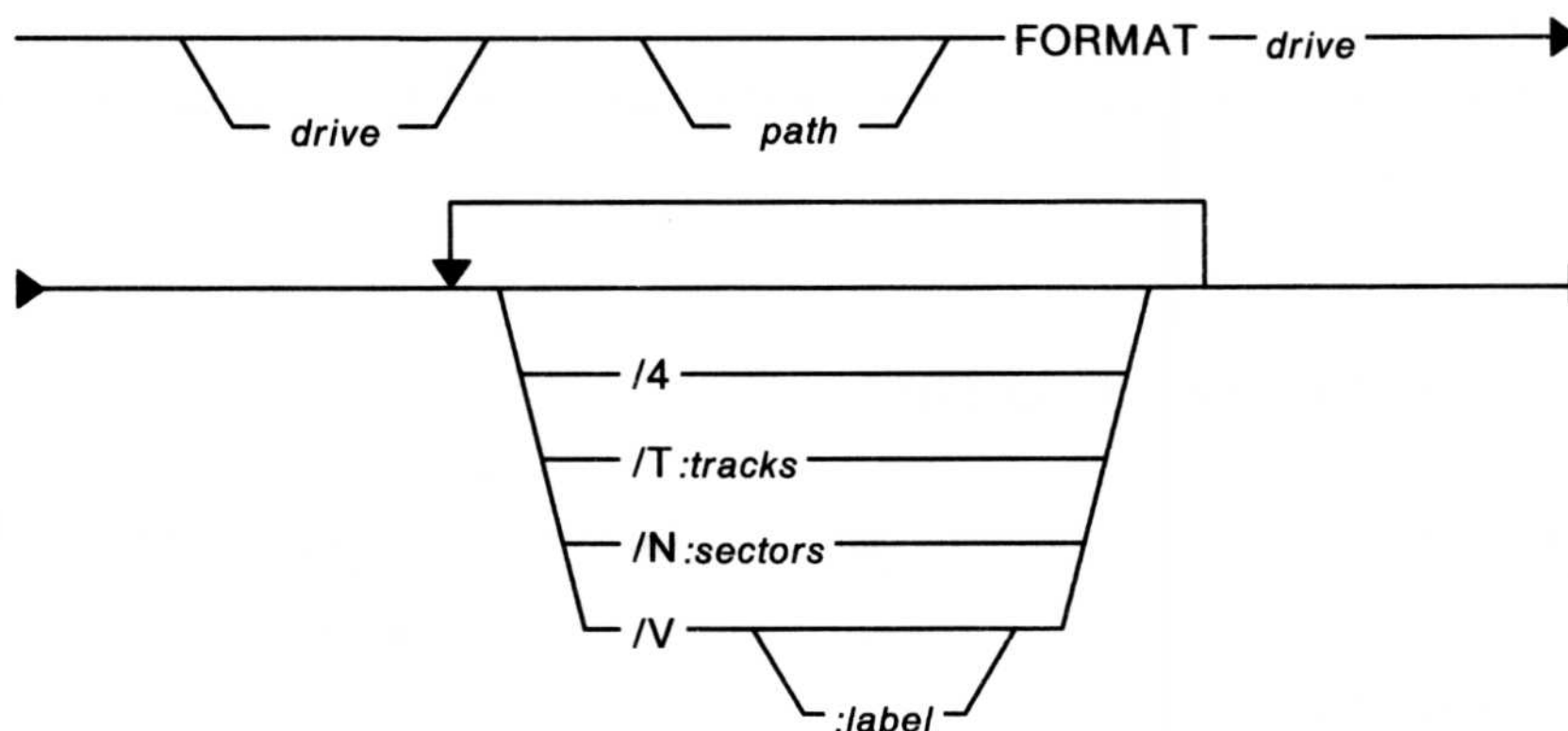
FORMAT

(Prepare a Disk for Use)

Purpose

This external command marks the directory and file allocation tables on a disk. Formats a disk in the specified drive to accept OS/2 files.

Syntax



Parameters

/4 Formats a 360KB diskette in a 1.2MB drive. This parameter is only intended to allow use of 360KB diskettes in 1.2 MB drives.

Warning: Diskettes formatted with this parameter on a 1.2MB drive may not be reliably read or written in a 360KB drive.

/T:tracks Formats a diskette to the number of *tracks* specified. For 720KB and 1.44MB, this value is 80. If /T is not specified, the default is 80.

FORMAT

(Prepare a Disk for Use)

/N:sectors Formats a 3.5-inch diskette to the number of *sectors* per track specified. For 720KB, this value is 9; for 1.44MB, this value is 18. The default depends on the diskette drive being used. However, if you are using a diskette that is less than the capacity of the diskette drive, you must specify a value for /N.

Note: Although the /T and /N parameters can also be used with 5.25-inch diskettes, their intended use is for 3.5-inch diskettes. See *Preparing and Maintaining Disks* in the “Using the Base Operating System” section of the *User's Guide*.

/V:label Specifies the volume *label*. A volume label can be up to 11 characters long and is used to identify the disk. Blanks are not allowed in the volume label you specify. If you do not specify a volume label, you are prompted to do so after formatting is completed.

Note: FORMAT does not work in DOS mode on drives involved with ASSIGN, JOIN, or SUBST.

Warning: Be sure to set up an OS/2 partition on all fixed disks before formatting them. FORMAT does not recognize a fixed disk as being an OS/2 disk if an OS/2 or an IBM DOS partition does not exist. When FORMAT finds such a disk, it skips to the next disk and begins formatting that one.

Note: The FORMAT command formats all media to the maximum capacity of the drive by default.

GOTO

(Transfer Processing to a Specified Label)

Purpose

Transfers control to the line following the one containing the appropriate label.

Syntax

GOTO _____ *label* _____|

Parameters

label Directs commands to run beginning with the line immediately after *:label*.

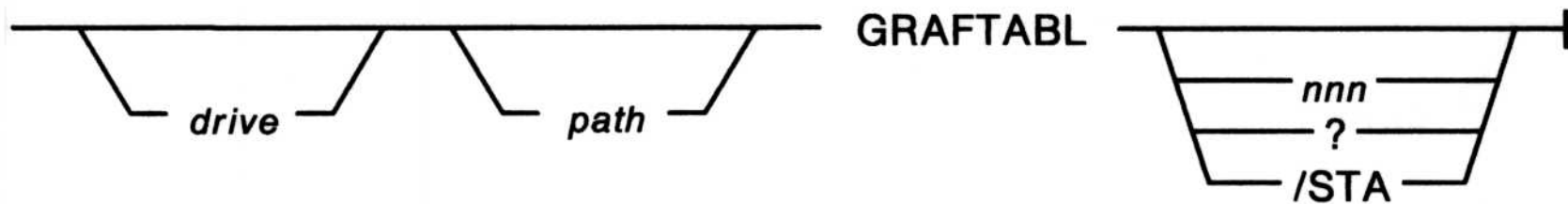


GRAFTABL (Display Extended Character Sets)

Purpose

This external command allows additional characters from a language code page to be displayed when using display adapters in graphics mode.

Syntax



Parameters

nnn Specifies a three-digit number, indicating the code page to be used, which can have a value of:

- 437** U.S.
- 860** Portuguese
- 863** Canada (French-speaking)
- 865** Nordic

? Displays the number of the code page currently being used and a list of code page options.

/STA Displays the number of the code page currently being used.

Note: Typing this command without a parameter displays the current graphic code page table that is loaded.

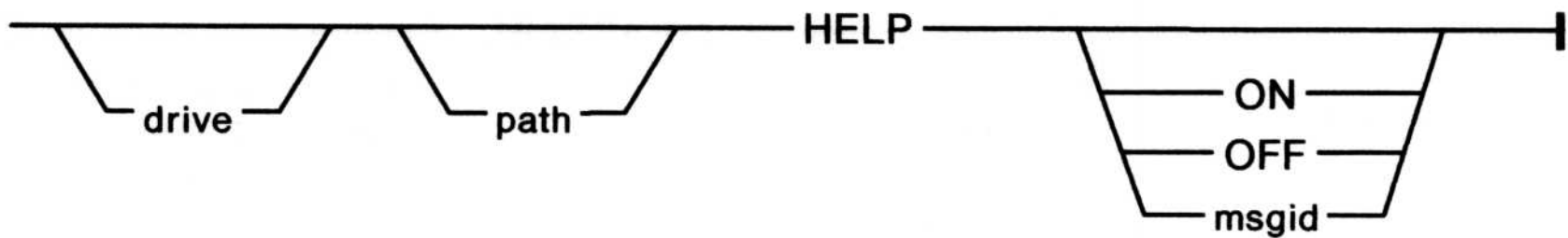
HELP

(Provide System Help)

Purpose

This external command provides a help line as part of the command prompt, a help panel, and information related to warning and error messages.

Syntax



Parameters

ON Turns on the display of the help line.

OFF Turns off the display of the help line.

msgid Identifies the system message prefix, when necessary, and number for which additional information is being requested.

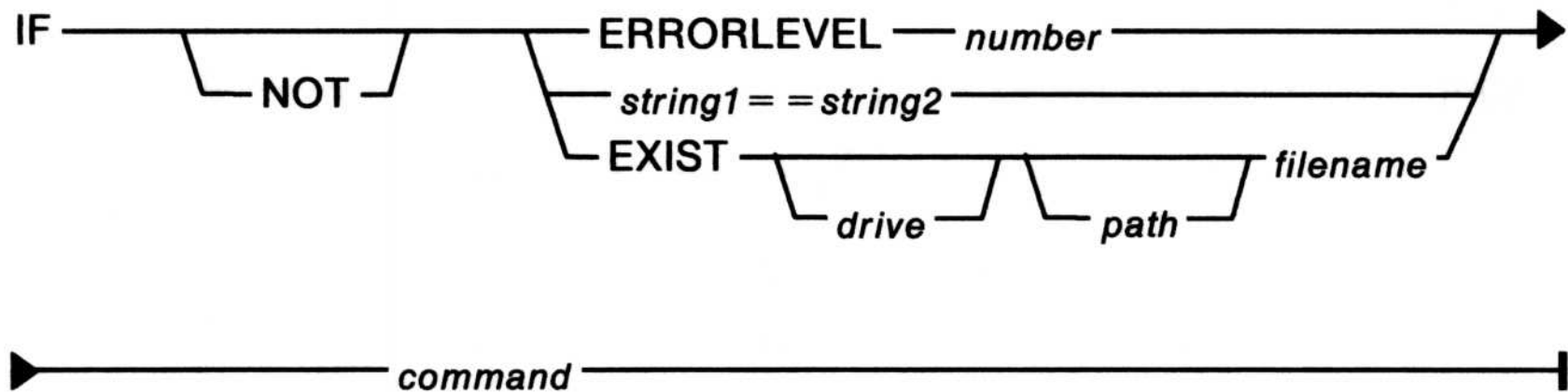
Note: Typing this command without a parameter displays the HELP options available for the current mode of operation.

IF (Allow Conditional Processing of Commands)

Purpose

Allows conditional operation of OS/2 commands.

Syntax



Parameters

NOT

Reverses the truth value of conditional parameters that follow.

ERRORLEVEL *number*

True, if the previous program has an exit code of *number* or higher. The number is specified as a decimal value.

string1 == *string2*

True, when *string1* and *string2* are identical.

The corresponding characters of *string1* and *string2* must both be uppercase or lowercase to be identical.

A blank string is not a valid syntax.

EXIST *filename*

True, if *filename* is found in the specified directory. Global characters (? and *) are allowed in *filename*.

command

Specifies the OS/2 command to be run when one of the conditional parameters listed above is true.

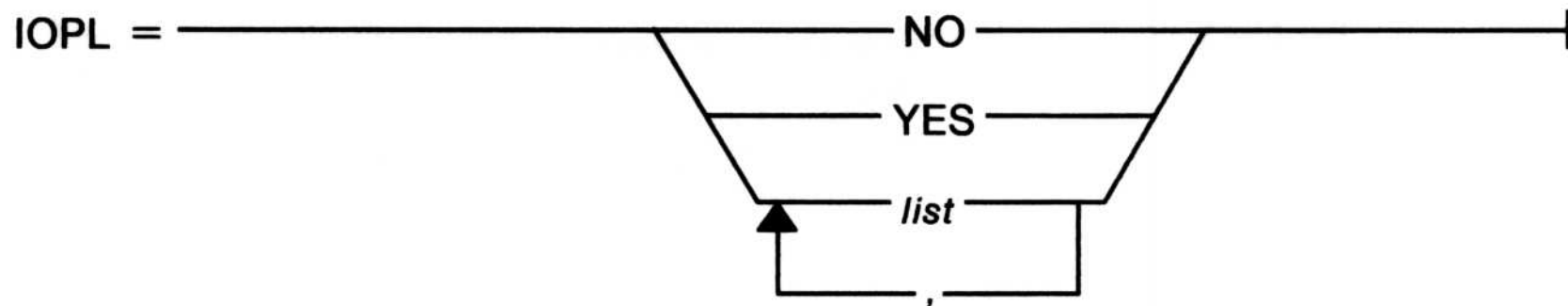
IOPL (Allow Input/Output Privilege)



Purpose

Allows I/O privilege to be granted to requesting processes in OS/2 mode.

Syntax



Parameters

NO Prevents I/O privilege from being granted.

YES Permits I/O privilege to be granted.

list Allows the restriction of the set of programs with this privilege. These are the names of the programs that require I/O privilege. Program names are provided by the suppliers of IOPL enabled programs.

, (*comma*) Used as a separator for repetitive variable parameters.



JOIN (Join Drives)

Purpose

This is an external command that logically connects a drive to a directory on another drive. You can only join a drive at the root directory.

Syntax

To join a drive to a directory on another drive:

```
drive path JOIN drive drive\directory
```

To display joined drives currently in effect:

```
drive path JOIN
```

To delete joined drives currently in effect:

```
drive path JOIN drive /D
```

Parameters

drive drive\directory The first drive is the one you want identified as *\directory* on the second drive.

/D Turns off a JOIN in effect.

JOIN

(Join Drives)



If the directory name does not exist, the base operating system creates a directory on the drive you specify. If the directory already exists, it must be empty for the JOIN to work. After you issue the JOIN command, the first drive name becomes incorrect and, if you try to use it, the base operating system displays an error message.

Notes:

1. Typing this command without a parameter displays the joined drive currently in effect.
2. The following commands do not work in DOS mode on drives involved with ASSIGN, JOIN, or SUBST:

BACKUP

CHKDSK

DISKCOMP

DISKCOPY

FORMAT

LABEL

RECOVER

RESTORE.



KEYB (Select a Keyboard Layout)

Purpose

This external command selects a keyboard layout to replace the current keyboard layout for all OS/2 and DOS full-screen sessions and all OS/2 window sessions.

Note: If you do not specify *layout* and your CONFIG.SYS file contains a keyboard DEVINFO statement, your keyboard is changed to the U.S. layout. If your CONFIG.SYS file does not have a keyboard DEVINFO statement and you enter KEYB without any parameters, an error message is displayed.

Syntax



Parameters

- layout* Specifies the keyboard layout. The values for *layout* are shown in the figure on the following page.
- subcountry code* Specifies a specific keyboard layout for countries that have more than one layout. The values are shown in the figure on the following page.

Note: Subcountry codes are associated with IBM enhanced keyboards only. Because France, Italy, and the United Kingdom have more than one IBM enhanced keyboard, the subcountry code allows you to select the keyboard you want.

KEYB

(Select a Keyboard Layout)



If you do not specify a subcountry code for these countries, then the layout codes for all machine types will default to 166 for the United Kingdom, 189 for France, and 141 for Italy. You must specify if you want subcountry code 168 for the United Kingdom, 120 for France, or 142 for Italy. Refer to the *IBM Operating System/2 Extended Edition Version 1.1 Keyboard Layouts* (referred to as *Keyboard Layouts* in the remainder of this book), if you do not know the subcountry code associated with your enhanced keyboard.

Layout	Keyboard	Subcountry Code
BE	Belgium	120
CF	Canada (French)	058
DK	Denmark	159
SU	Finland	153
FR	France	189, 120
GR	Germany	129
IT	Italy	141, 142
LA	Latin America	171
NL	Netherlands	143
NO	Norway	155
PO	Portugal	163
SP	Spain	172
SV	Sweden	153
SF	Switzerland (French)	150F
SG	Switzerland (German)	150G
UK	United Kingdom	166, 168
US	United States	103

**(Select a Keyboard Layout)**

Warning: If KEYB cannot load the keyboard layout for the system prepared code page, then KEYB asks you if you want to load the default translate table for that country's primary code page. If you respond with a **Y**, the default translate table is loaded, whereas if you respond with an **N**, the translate table is left unchanged.

If you choose to have the default code page loaded (refer to page 2-90), the keyboard may be using a different code page from the one used by the display. There are two valid tables of characters. In this case, certain characters entered from the keyboard might display incorrectly on the screen.

If you want characters that are displayed in the same default code page for the layout that you specified using KEYB, change the **CODEPAGE** statement in the **CONFIG.SYS** file to reflect the same default code page as shown in the "Code Pages Supported" table on page 2-90. In addition, if you do not know whether a character might be displayed differently on your screen, refer to the code page tables in *Keyboard Layouts*.

Note: Typing this command without a parameter displays the currently installed keyboard layout if a **DEVINFO** statement designating a keyboard layout is specified in your **CONFIG.SYS** file.

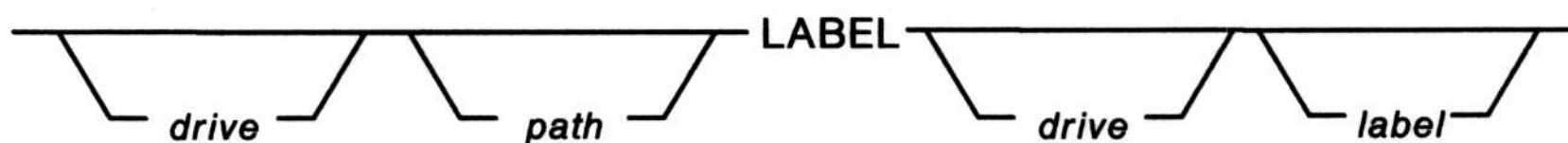
LABEL

(Create or Change Volume Label)

Purpose

This external command creates or changes the volume identification label on a disk.

Syntax



Parameters

label Specifies the volume label that identifies the disk. This can be up to 11 characters. All characters acceptable in file names are acceptable in the volume label. Unlike file names, however, the volume label does not contain a period between the eighth and ninth characters.

Notes:

1. Typing this command without a parameter displays the current label and prompts you if you want to change it.
2. LABEL does not work in DOS mode on drives involved with ASSIGN, JOIN, or SUBST.



LIBPATH (Identify Dynamic Link Libraries)

Purpose

Identifies the locations of dynamic link libraries for OS/2 programs.

Syntax

LIBPATH= _____
 ↑
 └───┬───┘
 drive path
 └──────────┬──────────┘
 ;──────────┘

Parameters

drive path Specifies a directory to be searched for dynamic link libraries. More than one directory may be searched by specifying additional paths, separated by semicolons. The paths are searched in the order specified.

MAXWAIT

(Set Maximum Wait)

Purpose

Sets the time limit for lack of access to the processor resource.

Syntax

MAXWAIT = _____ x _____

Parameters

x An integer number in seconds that must elapse before a regular class thread may receive a boost in priority because of a lack of access to the processor.

You may specify a number from 1 through 255 seconds.

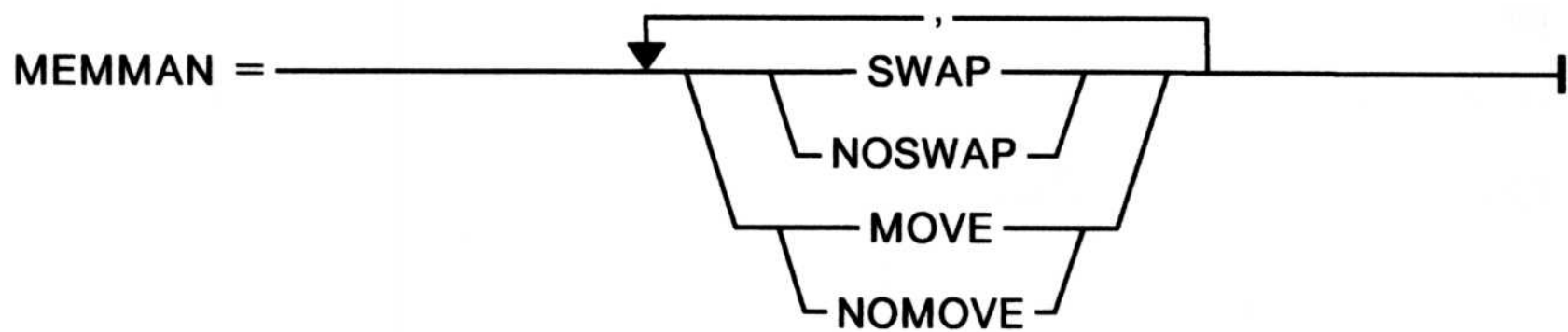


MEMMAN (Select Storage Allocation)

Purpose

Selects storage allocation options for OS/2 mode.

Syntax



Parameters

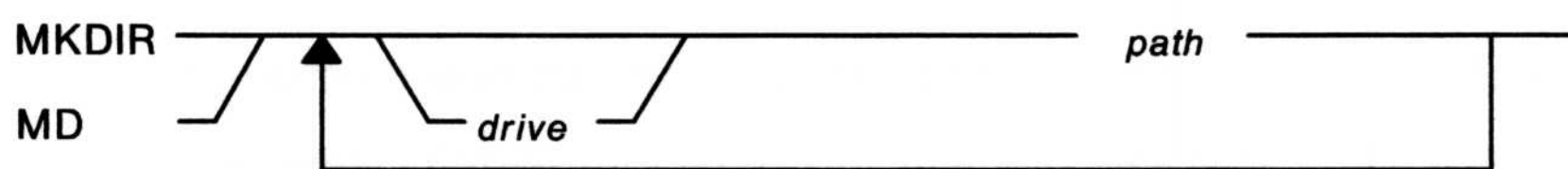
- SWAP** Permits segment swapping and storage compaction.
- NOSWAP** Prevents segment swapping and permits storage compaction.
- MOVE** On a diskette, permits storage compaction and prevents segment swapping. On a fixed disk, permits segment swapping and storage compaction.
- NOMOVE** Prevents storage compaction and segment swapping.

MKDIR or MD (Make Directory)

Purpose

Makes a new directory.

Syntax



Remarks

This command creates a multilevel directory structure, which is helpful in keeping related program or data files together.

MODE

(Set Communication Modes)

Purpose

Sets operation modes for devices.

- These operations include:
 - Setting asynchronous communication modes in OS/2 sessions for COM ports controlled by the COM0x.SYS device driver.
 - Querying asynchronous communication modes in OS/2 sessions for COM ports controlled by the COM0x.SYS device driver.
 - Setting asynchronous communication modes in DOS sessions for COM ports controlled by the COM0x.SYS device driver.
 - Setting color/graphics video modes
 - Setting parallel printer modes
 - Setting diskette I/O write verification or querying its status.

MODE

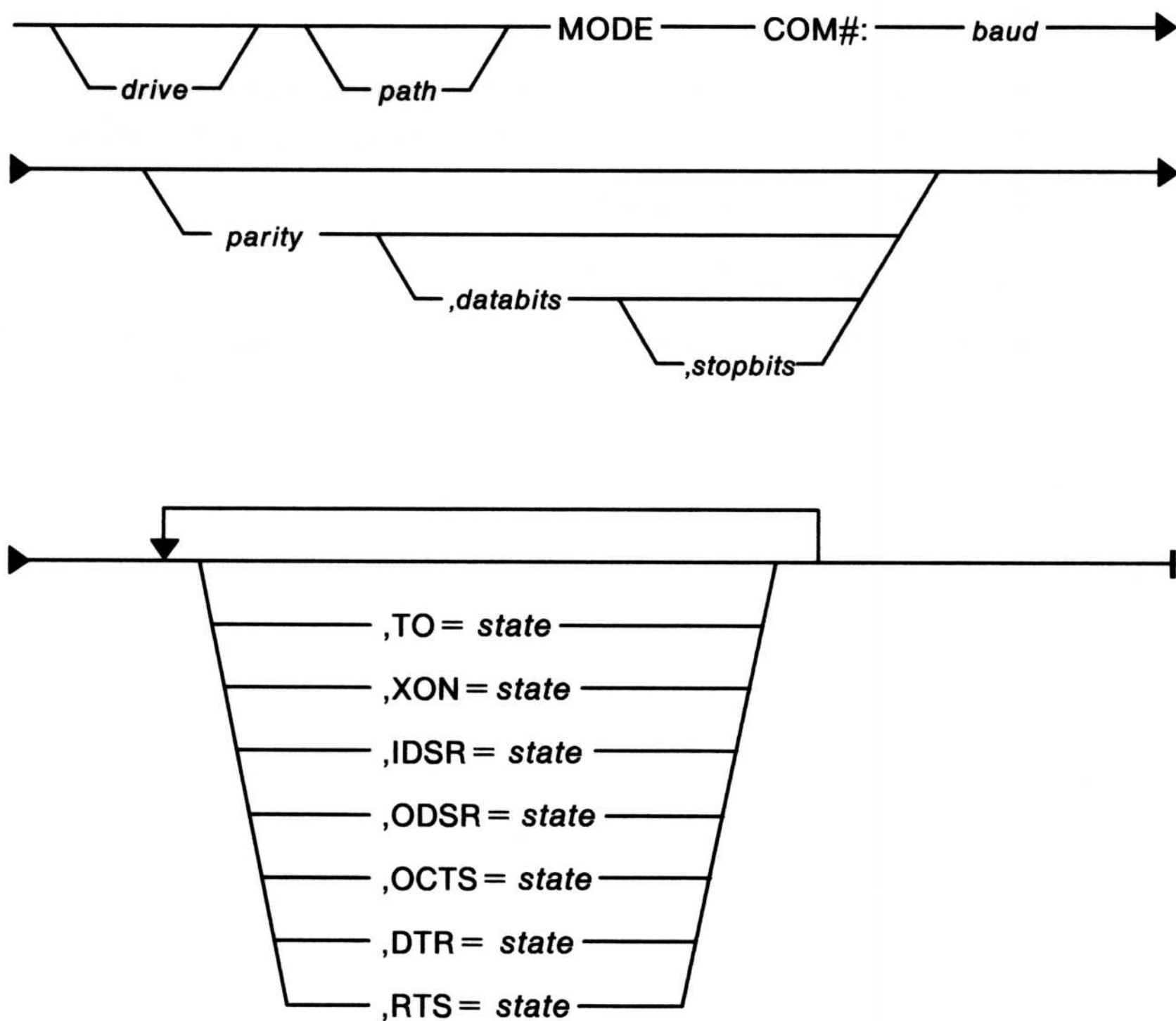
(Set Communication Modes)

Set Communication Modes

Syntax



To set OS/2 asynchronous communication modes for COM ports controlled by the COM0x.SYS device driver.



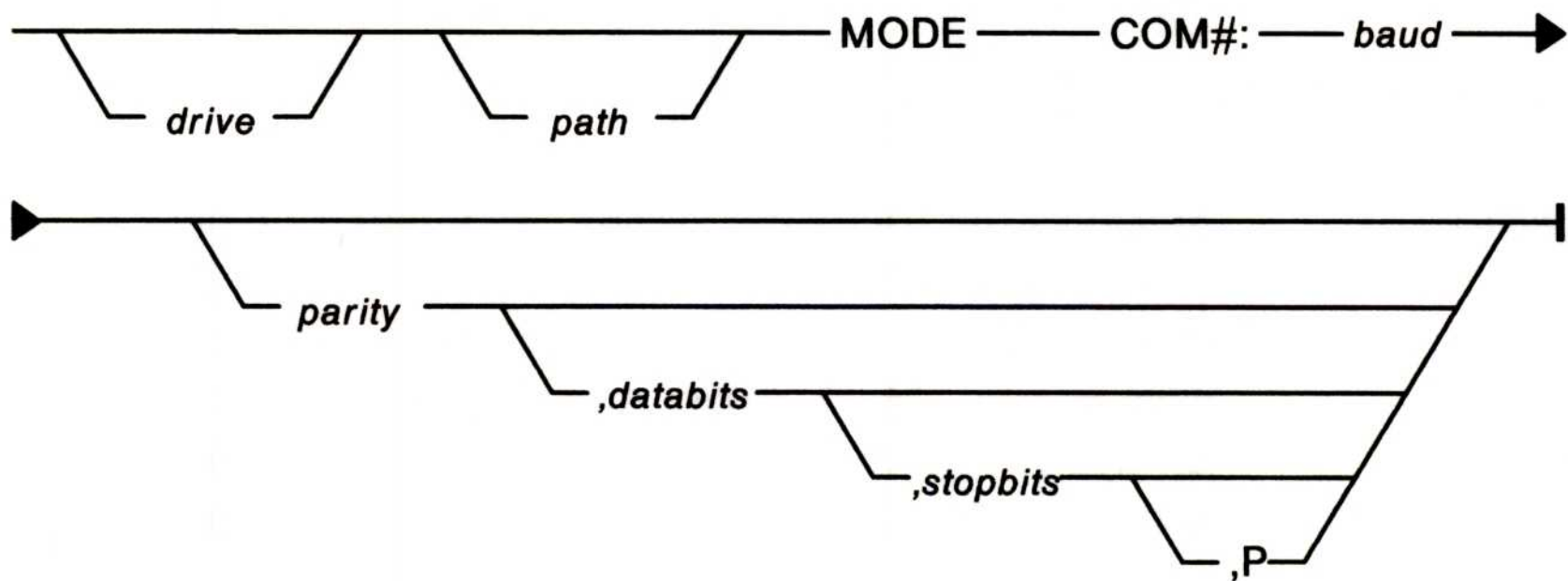
Note: The use of MODE to set COM port modes is allowed only when the Base Asynchronous Communication Device Driver (COM0x.SYS) is installed. For more information on this device driver, see “COM0x.SYS” on page 2-35. Also refer to the DEVICE statement on page 2-31.

MODE (Set Communication Modes)

Syntax



To set DOS asynchronous communication mode:



Note: The *parity*, *databits*, *stopbits*, and P-parameters are positional. When a positional parameter is omitted, but the one that follows it is specified, its position must be designated by a comma. For example, to accept the defaults for *parity*, *databits*, and *stopbits*, and specify the P-parameter, type:

```
MODE COM1:2400,,,P
```

Syntax



To query asynchronous communication port status in OS/2 mode of a COM port controlled by the COMO0x.SYS device driver:



MODE

(Set Communication Modes)

Parameters

COM#: Specifies the asynchronous communication port numbers and transmission rate. The # is asynchronous communication port numbers 1 through 8.

baud Specifies the transmission rate, which may be: 110, 150, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600, or 19200. (The baud parameter is required. You must specify at least the first two digits of the desired rate.) The default is 1200.

parity Specifies the following values: N (none), O (odd), E (even), M (mark), or S (space). None means that there is no parity bit. Odd means odd parity; even means even parity. Mark parity means that the parity bit is always 1. Space parity means that the parity bit is always 0. If parity is not specified, the default is E.

databits Specifies the value of data bits. Values are 5, 6, 7, or 8 bits of data, with the default of 7.

stopbits Specifies the value of stop bits. Values are 1, 1.5, or 2. Either 1 or 1.5 is valid if *databits* is specified as 5. If *stopbits* is specified as 1.5, only 5 is valid for *databits*. If *stopbits* is omitted, the default is 2 when *baud* is 110; otherwise, the default is 1.



P Specifies a time-out value of approximately 30 seconds in DOS mode for IBM DOS programs that go directly to the hardware. These programs need to know the port address; you must run the SETCOM40 command first before printing or plotting. The P-parameter does not provide infinite retries as in previous versions of IBM DOS.

MODE

(Set Communication Modes)



- TO = state** Specifies time-out processing ON or OFF. ON allows write infinite time-out processing. If OFF is specified, normal time-out processing is done. The default mode is TO=OFF. If this parameter is not specified, its current value is unchanged.
- XON = state** Sets automatic transmit flow control. ON allows and OFF prevents automatic transmit flow control. The default mode is XON=OFF. If this parameter is not specified, its current value is unchanged.
- IDSR = state** Sets input handshaking using 'data set ready' (DSR). ON allows and OFF prevents input handshaking using 'data set ready' (DSR). The default mode is IDSR=ON. If this parameter is not specified, its current value is unchanged.
- ODSR = state** Sets output handshaking using 'data set ready' (DSR). ON allows and OFF prevents output handshaking using 'data set ready' (DSR). The default mode is ODSR=ON. If this parameter is not specified, its current value is unchanged.
- OCTS = state** Sets output handshaking using 'clear to send' (CTS). ON allows and OFF prevents output handshaking using 'clear to send' (CTS). The default mode is OCTS=ON. If this parameter is not specified, its current value is unchanged.
- DTR = state** Sets 'data terminal ready' (DTR). ON allows 'data terminal ready' (DTR), OFF prevents DTR, and HS allows input handshaking using DTR. The default mode is DTR=ON. If this parameter is not specified, its current value is unchanged.
- RTS = state** Sets 'request to send' (RTS). ON allows 'request to send' (RTS), OFF prevents RTS, HS allows input handshaking using RTS, and TOG allows RTS toggling on transmit mode. The default mode is RTS=ON. If this parameter is not specified, its current value is unchanged.

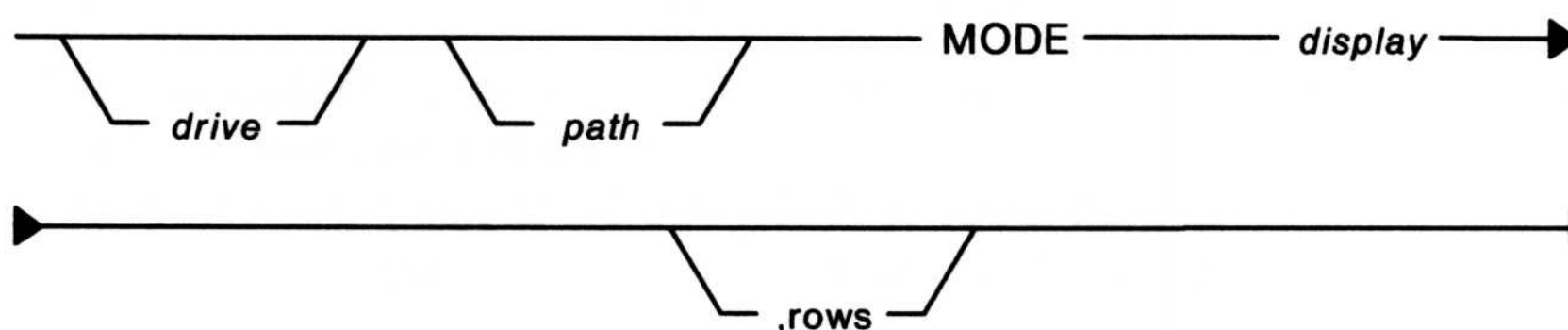
MODE

(Set Communication Modes)

Set Display Modes

Syntax

To set video modes:



Set video modes

Parameters

display Specifies the display mode. The options are: 40, 80, BW40, BW80, CO40, CO80, or MONO. The value 40 specifies a display width of 40 characters per line and 80 specifies a display width of 80 characters per line. CO and BW refer to a color graphics monitor adapter with color (CO) or without (BW). MONO specifies the monochrome mode, which always has a display of 80 characters per line.

Note: VIO windowed sessions only support 80 column modes.

rows Specifies the number of rows to select. Valid rows are 25, 43, or 50, depending on the display adapter attached. For example, with an enhanced display adapter, you can specify a 43-line display:

```
MODE C080,43
```

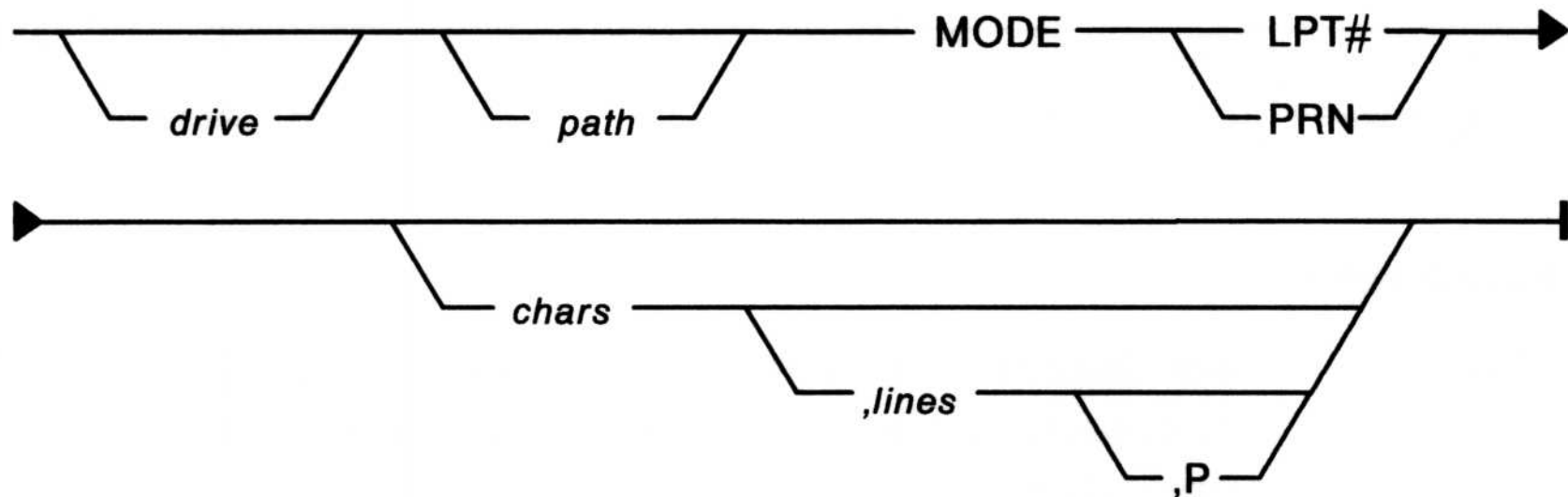
If *row* is not specified, the value does not change. The initial value for rows is 25.

MODE (Set Communication Modes)

Set Parallel Printer Mode

Syntax

To set parallel printer modes:



Note: The *chars*, *lines*, and *P* parameters are positional. When a positional parameter is omitted, but the one that follows it is specified, its position must be designated by a comma.

Parameters

- LPT#** Specifies the printer number. The # defines printer number 1, 2 or 3. PRN is an acceptable device name for LPT1.
- chars* Specifies the characters per line. The values are either 80 or 132 characters. The default is 80.
- lines* Specifies the lines per inch vertical spacing. The line values are either 6 or 8. The default is 6.
- P** Tells MODE to continuously try to send output to the printer if a time-out error occurs.

Note: Once you have issued the MODE command and specified values for either *chars* or *lines* or both these parameters, their defaults are the last valid values you typed for them.

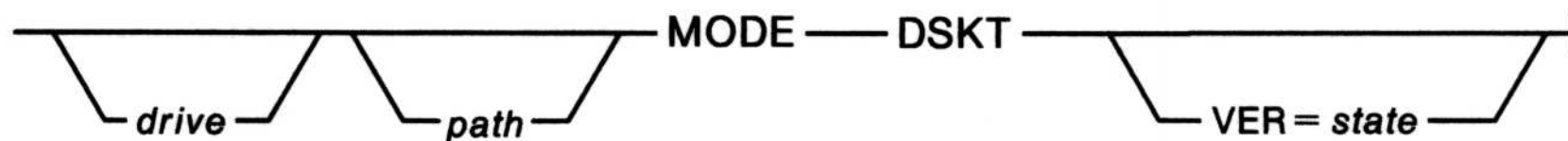
MODE

(Set Communication Modes)

Set Diskette I/O Write Verification

Syntax

To set diskette I/O write verification.



Parameters

state Sets diskette verification ON or OFF. VER = ON turns on diskette verification; VER = OFF turns off diskette verification.

MORE

(Display Output One Screen at a Time)

Purpose

This external command reads data from the standard input device and sends data to the standard output device (usually the display) one full screen at a time.

Syntax

A diagram illustrating the syntax of the MORE command. A horizontal line represents the command structure. On the left side, there are two trapezoidal shapes pointing downwards, labeled 'drive' and 'path' respectively, indicating optional arguments. To the right of these shapes, the word 'MORE' is written, followed by a horizontal line that ends in a vertical bar on the right side.

Note: One screen at a time does not imply one physical window of information at a time when used in a windowed command prompt.

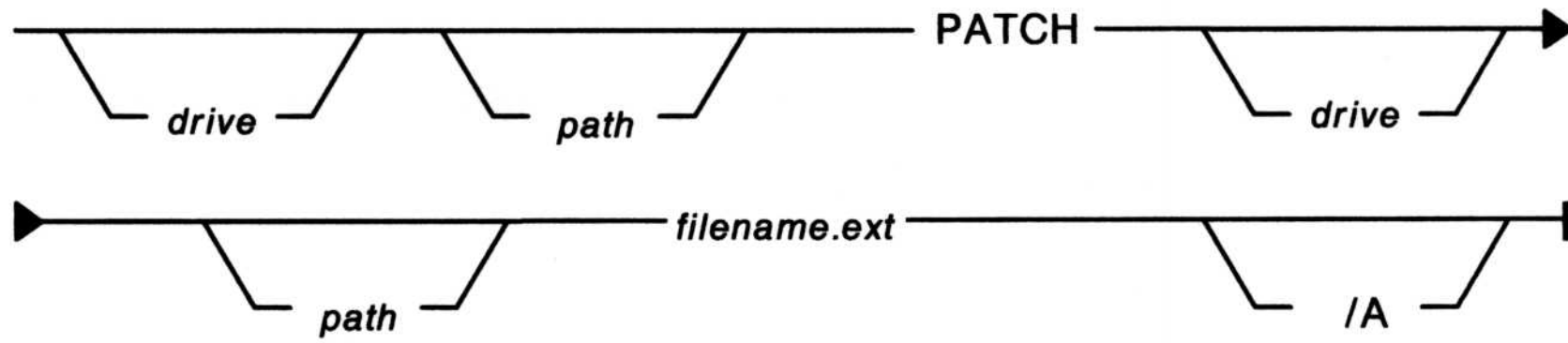
PATCH

(Apply Software Repairs)

Purpose

This external command allows you to apply IBM-supplied patches to make repairs to software.

Syntax



Parameters

/A Specifies automatic mode. In this mode, the file name is a file containing instructions for patching one or more files. If /A is not specified, interactive mode is assumed.

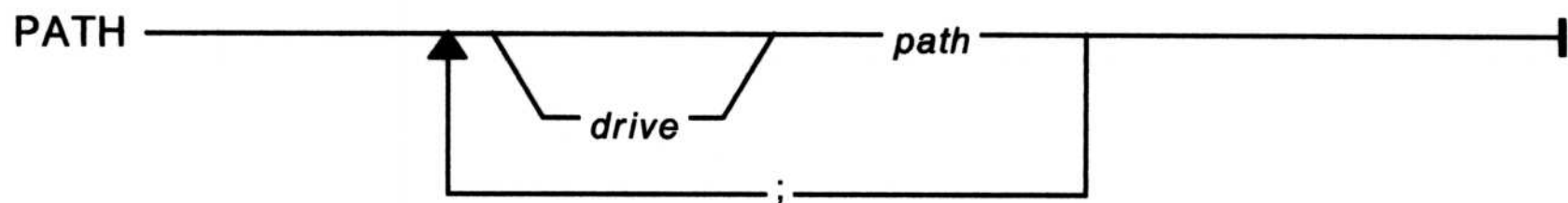
PATH (Set Search Path)

Purpose

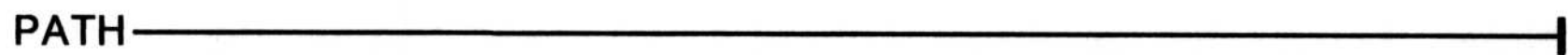
Sets a search path for commands.

Syntax

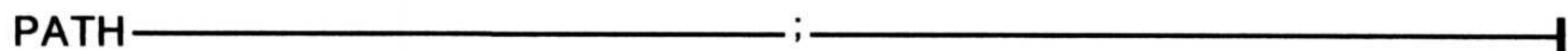
To specify paths to commands and programs:



To display paths currently in effect:



To delete use of the PATH command:



Note: In OS/2 mode, PATH can be specified with the SET command. For more information on the SET command, see page 2-128.

The default path set during installation is:

C:\SQLLIB;C:\CMLIB;C:\OS2;C:\OS2\SYSTEM;
C:\OS2\INSTALL;C:\;

PAUSE

(Suspend Batch Operation)

Purpose

Suspends running of the batch file and the following message is displayed:

Press any key when ready . . .

Syntax

```
PAUSE _____|
                \_____/
                 comment
```

Parameters

comment The *comment* can be any string of characters.

Note: Typing this command without a parameter displays the message:

"Press any key when ready...."

PAUSEONERROR (Pause CONFIG.SYS Messages)

Purpose

Allows or prevents pausing when error messages are issued during the processing of the CONFIG.SYS file.

Syntax

PAUSEONERROR = 

Parameters

- YES** Causes the system to pause. If errors have been detected in the CONFIG.SYS file, the system pauses before starting the specified user interface or command processor and displays a prompt. Processing does not continue until you press the Enter key.
- NO** Prevents the system from pausing before starting the user interface or command processor.

PRINT

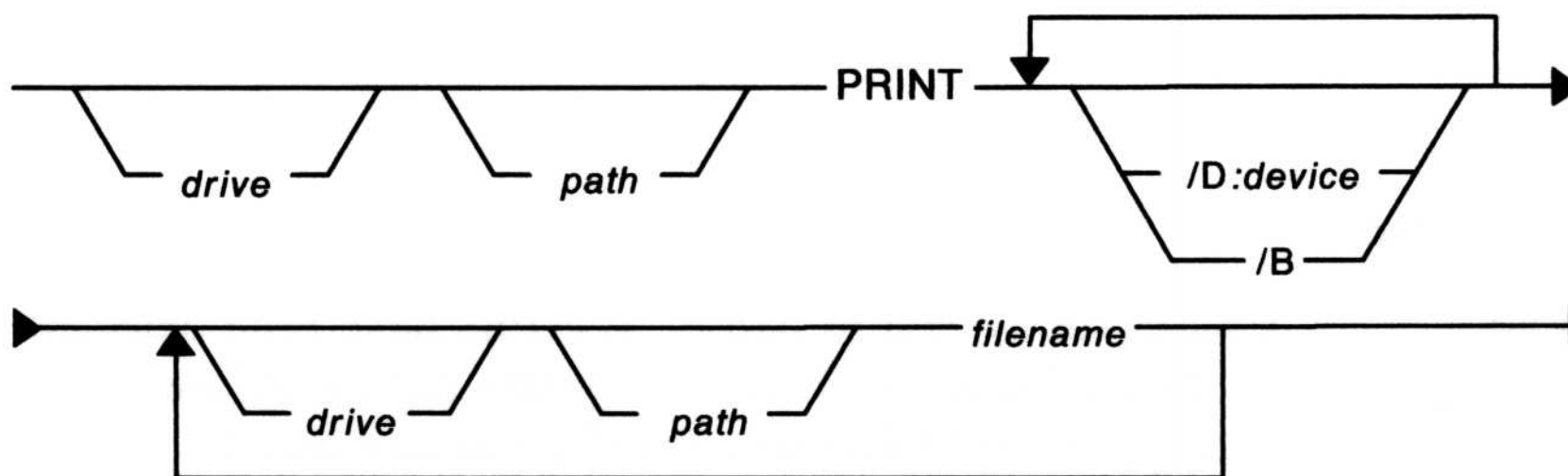
(Print Information)

Purpose

This external command prints or cancels printing of one or more files.

Syntax

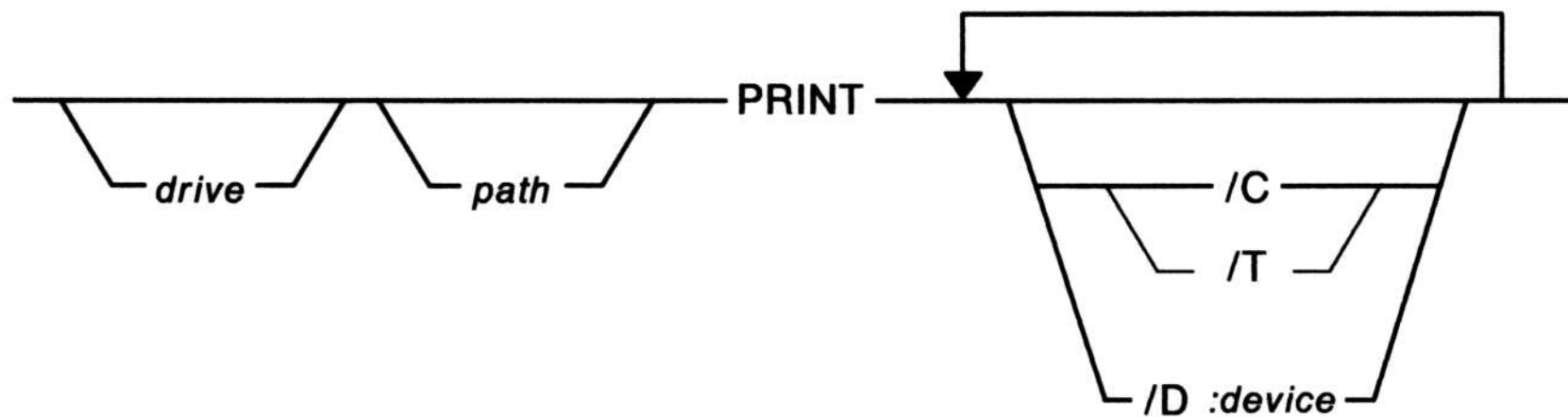
To print one or more files on a specified device:



Syntax



To cancel the printing of the current file or all files waiting to be printed on a specified device:



PRINT

(Print Information)

Parameters

/D:*device* Specifies the print device. If not specified, the default device is LPT1. The acceptable device names are PRN, LPT1, LPT2, and LPT3.

/B Causes the Ctrl and Z characters encountered not to be interpreted as end-of-file indicators. The entire file is printed as is, without end-of-file character (Ctrl and Z) processing.



/C Cancels the currently printing file if the spooling is active for the specified device. The /C parameter works only in OS/2 mode.



/T Cancels all files waiting to be printed if the spooling is active for the specified device. The /T parameter works only in OS/2 mode.

PROMPT

(Change Command Prompt)

Notes:

1. Typing this command without a parameter resets the system prompt to the system default.
2. In OS/2 mode, PROMPT can be specified with the SET command. For more information on the SET command, see page 2-128.

PROTSHELL

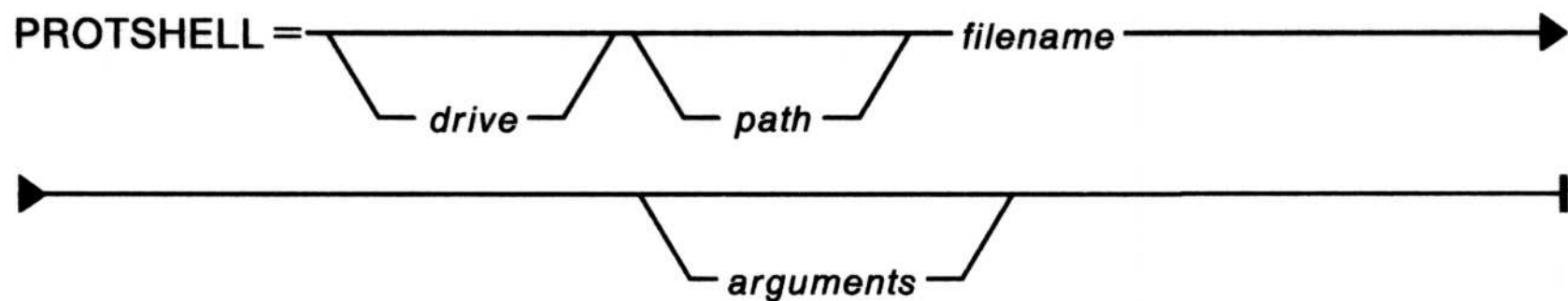
(Load a User Interface Program)



Purpose

Loads the user interface program and OS/2 command processor.

Syntax



Parameters

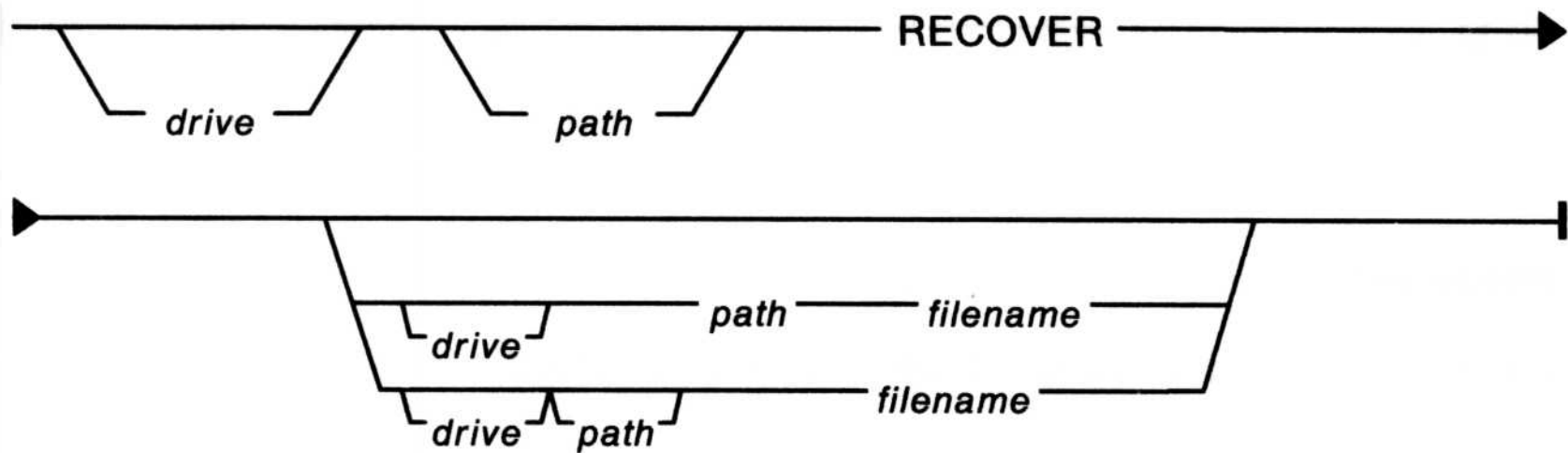
- filename* Represents the complete name of the file, including its extension, that contains the user interface.
- arguments* Specifies the parameters required by the user interface. The standard OS/2 user interface requires the name of the Presentation Manager configuration file, the name of the Presentation Manager program file, and the name of the OS/2 mode command processor.

RECOVER (Recover Files)

Purpose

This external command recovers files from a disk containing defective sectors.

Syntax



Note: RECOVER does not work in DOS mode on drives involved with ASSIGN, JOIN, or SUBST.

RENAME or REN (Rename Files)

Purpose

Changes the name of a file.

Syntax

RENAME _____ *filename 1* — *filename 2* —|
REN _____ *drive* _____ *path* _____

Parameters

filename1 Specifies the file to be renamed.

filename2 Specifies the new file name.

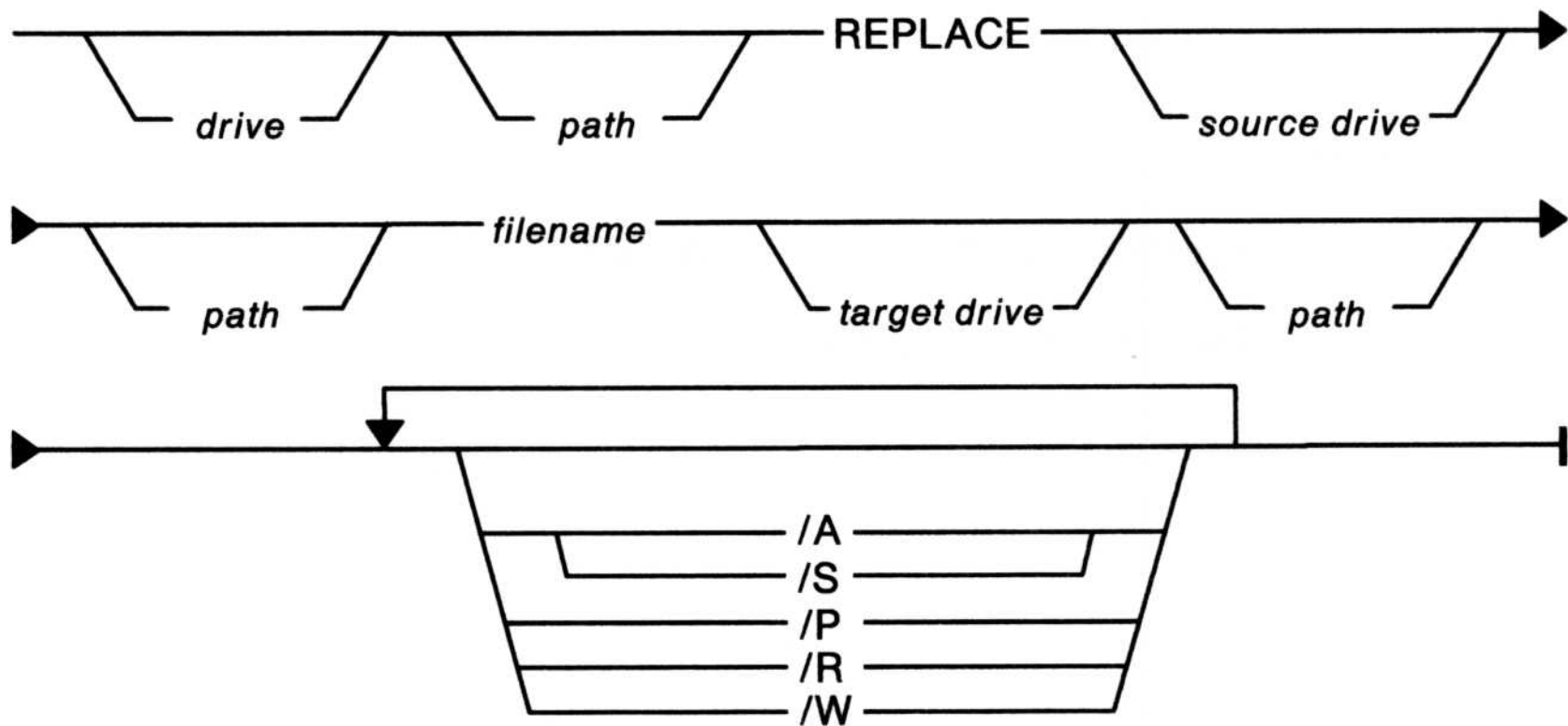
REPLACE

(Replace Files)

Purpose

This is an external command that selectively replaces files on the target drive with files of the same name from the source drive. Also, selectively adds files from the source drive to the target drive.

Syntax



Parameters

- /A** Copies all files specified by the source that do not exist on the target. This allows you to add files to the target without overwriting the files that already exist on the target. You cannot use /A and /S together.
- /S** Searches all directories of the target drive and path specified for the files matching the file name specified. If only the target drive is specified, the default path is assumed. You cannot use /A and /S together.
- /P** Prompts you as each file is encountered on the target. This allows selective replacing or adding.
- /R** Replaces files that are read-only on the target.
- /W** Waits for you to insert a diskette before beginning to search for source files.

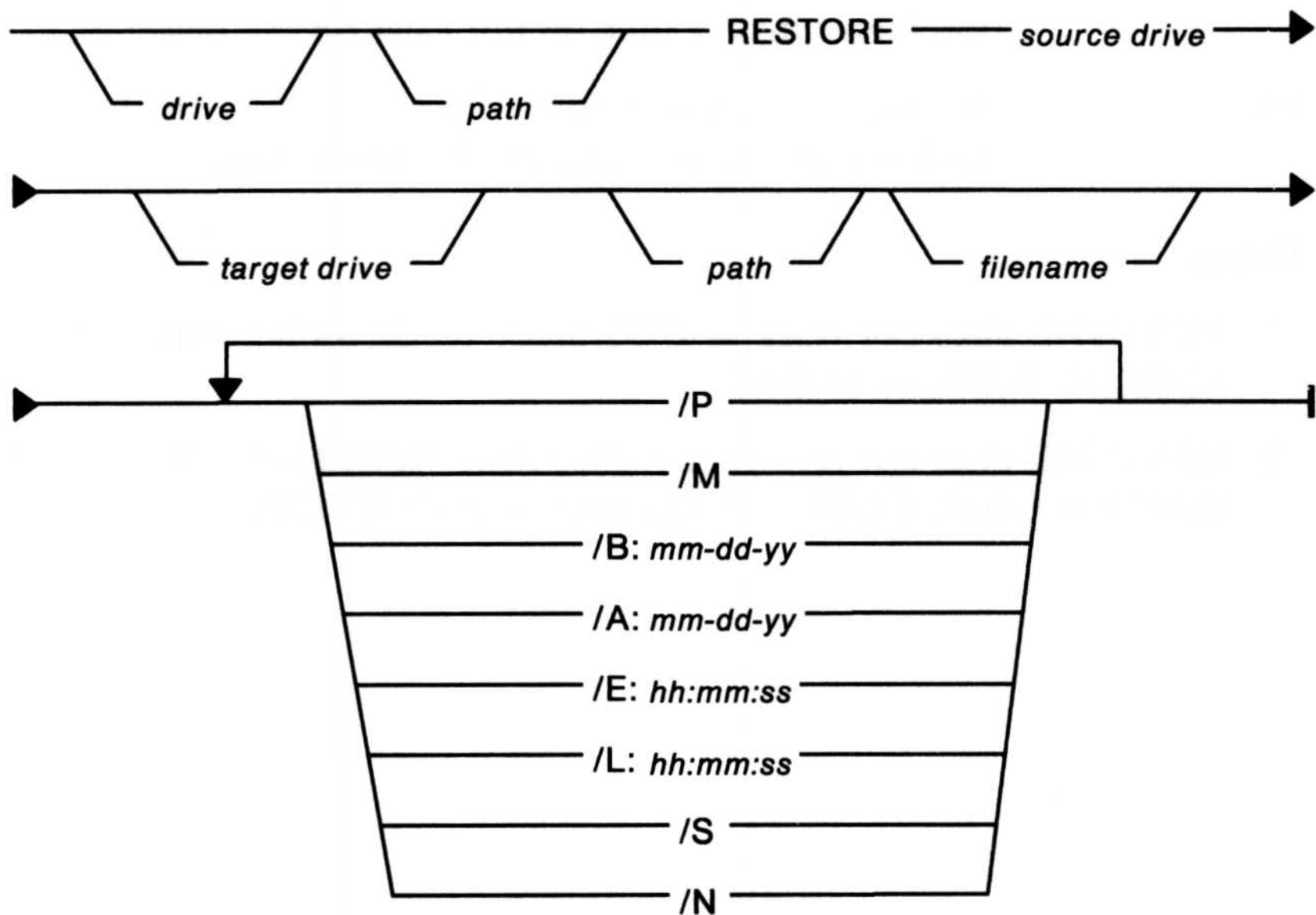
RESTORE

(Restore Backed-up Files)

Purpose

This external command restores one or more BACKUP files from one disk to another.

Syntax



Parameters

- /P** Prompts for permission to restore any files on the *target disk* that match the file specification and are read-only or have changed since they were last backed up.
- /M** Restores those files on the *target disk* that have been modified since the last backup.
- /B:mm-dd-yy** Restores those files on the *target disk* last modified on or before the given date.

RESTORE

(Restore Backed-up Files)

- /A:mm-dd-yy** Restores those files on the *target disk* last modified on or after the given date.
- /E:hh:mm:ss** Restores those files on the *target disk* last modified before the given time.
- /L:hh:mm:ss** Restores those files on the *target disk* last modified at or after the given time.
- /S** Restores any subdirectories from the *backup disk* to the target disk if they do not exist on the target disk.
- /N** Restores any files from the *backup disk* to the target disk if they do not exist on the target disk.

Notes:

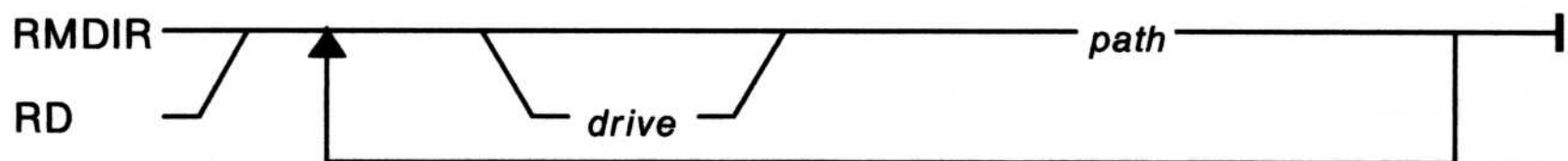
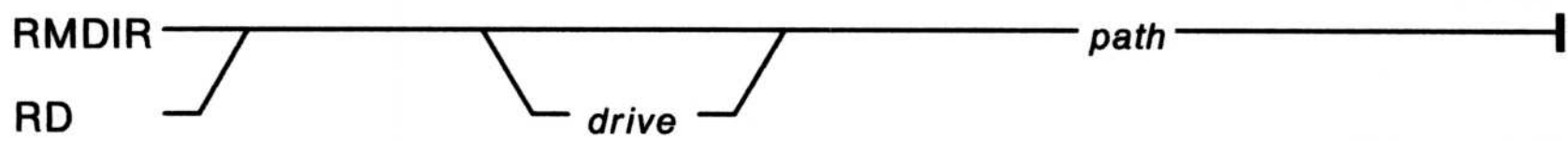
1. RESTORE does not work in DOS mode on drives involved with ASSIGN, JOIN, or SUBST.
2. RESTORE does not restore the OS/2 files: IBMBIO.COM, IBMDOS.COM, COMMAND.COM, and CMD.EXE.

RMDIR or RD (Remove Directory)

Purpose

Removes empty directories from a multilevel directory structure.

Syntax



Warning: You cannot remove the root directory or the current directory; you must be in a directory other than the current directory to remove a subdirectory.

RMSIZE

(Specify DOS Mode Size)



Purpose

Specifies the highest storage address allowed for the DOS operating environment.

Syntax

RMSIZE = _____ x _____

Parameters

x Specifies a number from 0 through 640, representing a multiple of 1024 bytes.

Default

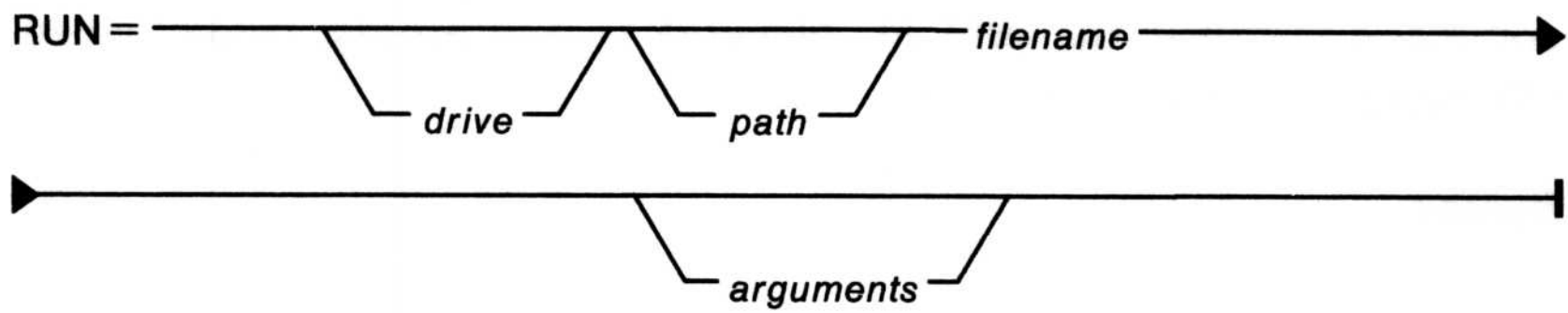
If you do not specify a RMSIZE statement, the default is the total memory installed, minus 512KB or 640KB, depending on whichever is installed in the lower address base. This is the largest usable size for DOS mode.

RUN (Start System Programs)

Purpose

Loads and starts a system program during system initialization.

Syntax



Parameters

- filename* Specifies the complete name of the file, including its extension, that contains the system program you want to run.
- arguments* Specifies information passed to the program.

RUN

(Start System Programs)

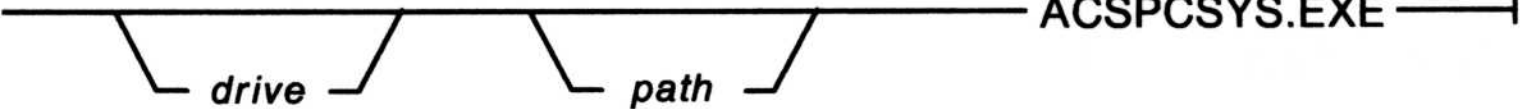
ACSPCSYS.EXE System Process



Purpose

Allows users to use the IBM PC Network system process. This command results in the IBM PC Network system process being loaded whenever the system is started.

Syntax

RUN = 

Parameters

- drive:* The drive where the ACSPCSYS.EXE file resides. Specify this drive only if it is not in the default drive.
- path* Path designator that specifies where the file is. Use this path designator if the file is not in the default path.

Remarks

- The PC Network device driver (PCNETDD.SYS) must be loaded before this system process is loaded.
- The RUN command must follow the device driver statements.

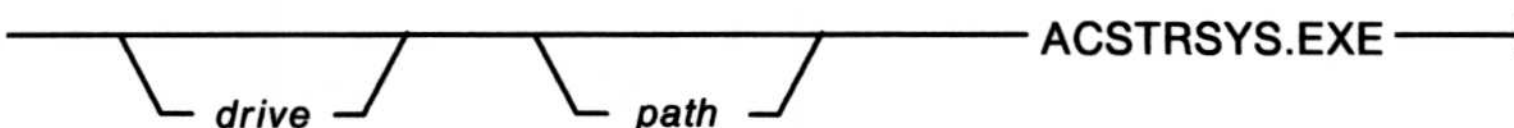
ACSTRSYS.EXE System Process



Purpose

Allows users to use the IBM Token-Ring system process. This command results in the IBM Token-Ring system process being loaded whenever the system is started.

Syntax

RUN = 

Parameters

- drive:* The drive where the ACSTRSYS.EXE file resides. Specify this drive only if it is not in the default drive.
- path* Path designator that specifies where the file is. Use this path designator if the file is not in the default path.

Remarks

- The Token-Ring device driver (TRNETDD.SYS) must be loaded before this system process is loaded.
- The RUN command must follow the device driver statements.

SET

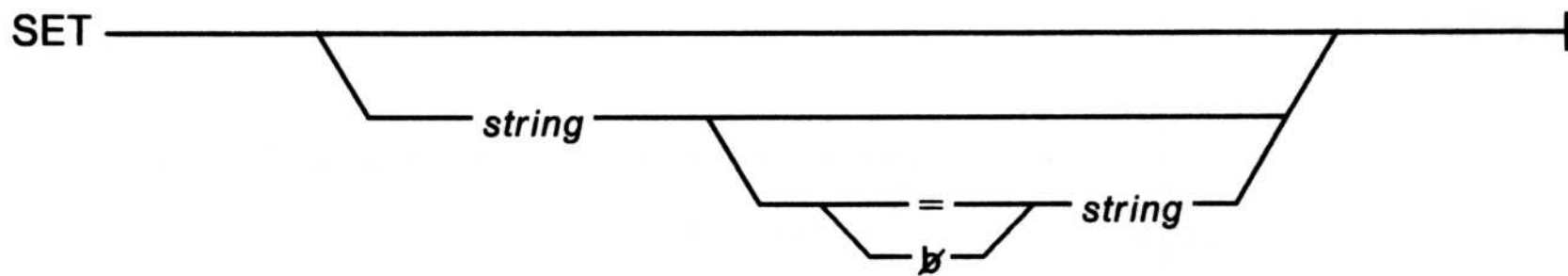
(Set Environment Variables)

Purpose

Sets one string in the environment equal to another string for later use in programs.

Syntax

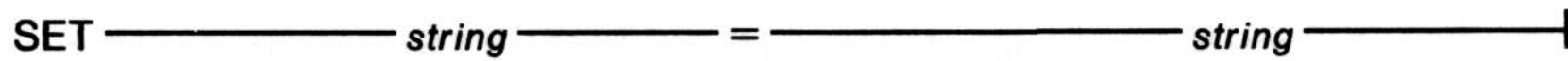
To use SET from the command prompt:



Syntax



To use SET in your CONFIG.SYS file



SET

(Set Environment Variables)

Parameters

string The first *string* is the name of a system variable or replaceable parameter. The second *string* is the value you want to assign to the system variable or replaceable parameter.

Note: Typing this command without a parameter displays the environment variables for the current mode of operation.

SETCOM40

(Set COM Port Address)

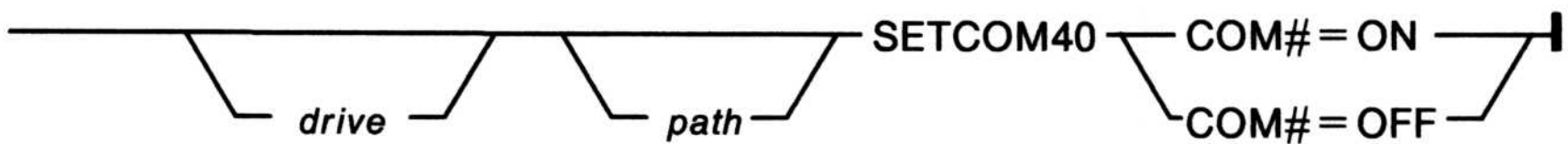


Purpose

Sets the COM port address so that a DOS program can access the COM port interface directly to support a serial device, such as a plotter, printer, or mouse, when the COM0x.SYS device driver is installed.

Warning: Do not use this command when the COM port specified with it can be accessed by currently running applications in OS/2 mode by means of the COM0x.SYS device driver or the ASYNCDDx.SYS device driver.

Syntax



Parameters

- | | |
|-------------|--|
| COM# | Specifies com port COM1, COM2, or COM3 |
| ON | Sets the port address for a serial device. |
| OFF | Removes the port address. |



SETLOCAL (Define Local Variables)

Purpose

Lets you define the drive, directory, and environment variables that are local to the current batch file. For information on restoring the drive, directory, and environment variables that were in effect before the SETLOCAL command was run, see the ENDLOCAL command on page 2-72.

Syntax

SETLOCAL _____|

SHELL

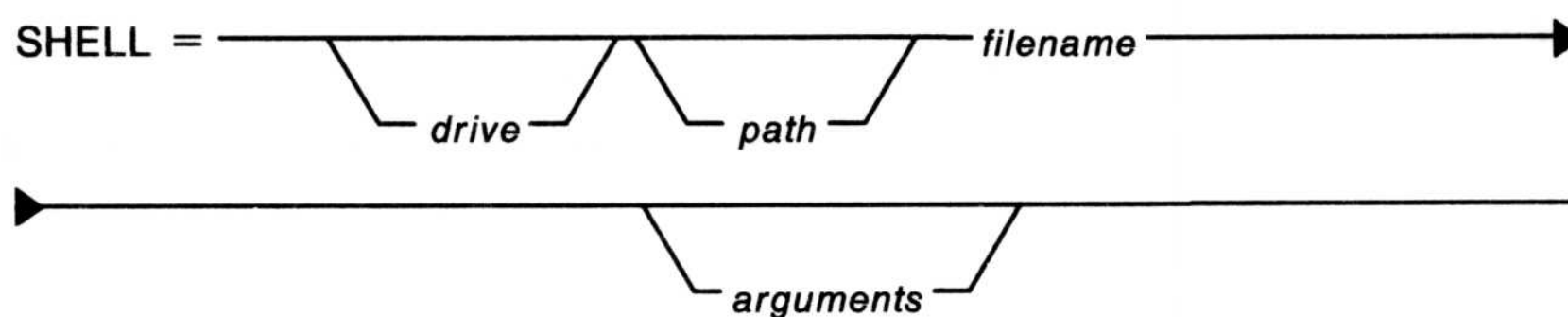
(Install a Command Processor)



Purpose

Loads and starts the DOS command processor, COMMAND.COM, or allows you to replace the DOS command processor with another command processor.

Syntax



Parameters

- filename* Specifies the complete name of the file, including its extension, that contains the DOS mode command processor.
- arguments* Specifies the parameters of your command processor.

Note: For more information on starting command processors in DOS mode, see the COMMAND command on page 2-19.

SHIFT

(Use More Than 10 Replaceable Parameters)

Purpose

Allows the use of more than 10 replaceable parameters in batch file processing.

Syntax

SHIFT

Examples

Assume:

```
%0 = 'big'  
%1 = 'golden'  
%2 = 'car'  
%3 - %9 are empty
```

A SHIFT results in the following:

```
%0 = 'golden'  
%1 = 'car'  
%2 - %9 are empty
```

For more information on using replaceable parameters in batch files, see *Creating and Changing Batch Files* in the “Using the Base Operating System” section of the *User's Guide*.

SORT

(Sort Information by Letter or Number)

Purpose

This external command reads data from standard input, sorts the data, and writes it to standard output.

Syntax

A diagram showing the syntax of the SORT command. A horizontal line represents the command line. The word "SORT" is centered on the line. To the left of "SORT", there are two trapezoidal shapes representing file paths, with the labels "drive" and "path" written below them. To the right of "SORT", there are two trapezoidal shapes representing options, with the labels "/R" and "+n" written below them. The line ends with a vertical bar on the right side.

Parameters

- /R** Reverses the sort; that is, sorts from Z through A.
- +n** Sorts starting with column *n*, where *n* is some number. If you do not specify this parameter, SORT begins sorting from column 1.

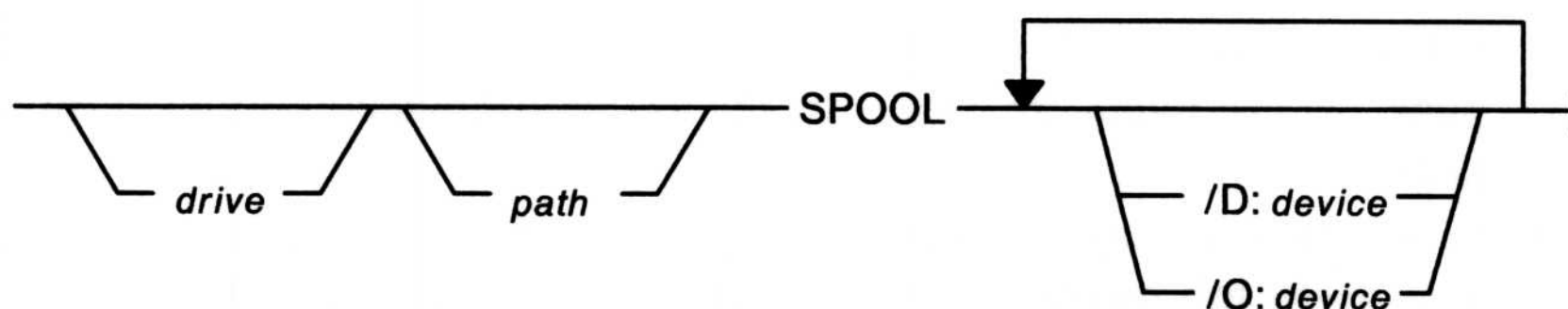


SPOOL (Specify Print Queue)

Purpose

This external command intercepts and separates data going to the printer from different applications so that printer output from running applications at the same time is not intermixed.

Syntax



Parameters

/D:device Specifies the input print device. This is the device the application program thinks it is printing to. Acceptable device names are LPT1, LPT2, LPT3, and PRN. If not specified, the default device is LPT1. COM# devices *cannot* be specified as input devices.

/O:device Specifies the output print device. This is the actual device on which output from the application program is printed. If not specified, the default is the same as the /D:device. COM1 through COM3 may be specified as output devices when a serial printer is attached to the serial port.

SPOOL

(Specify Print Queue)



Warning:

- When spooling to a serial printer, you may have to issue the MODE command so that the COM port interfaces correctly with the printer. You have the option of setting MODE in the Control Panel or using the MODE command at the command prompt. Note, however, that if you have MODE set in both the Control Panel and at the command prompt, the PM (Presentation Manager) device driver overwrites the configuration that you did not set in the Control Panel. The MODE command can be issued before or after starting SPOOL. Refer to the MODE command on page 2-97 to query the status of the asynchronous communication port.
- If you have set up the spooler for print pooling, it may be that your print job might print on another physical device. The reason for this is that the queue processor sends the highest priority print job to the next available printer (not currently printing). For example, if you specify the input device /D:LPT1 and specify the output device /O:LPT2, the output will go into a queue for the LPT2 device but because of the multiple output devices listed in the Control Panel, the data might print on an LPT3 device. If you associate one queue to LPT2, you are guaranteed to have all output print on the port you specified.

Note: The Base Asynchronous Communications Device Driver (COM0x.SYS) supports only COM1 through COM3 for the IBM Personal System/2; and COM1 and COM2 for the IBM Personal Computer AT and the IBM Personal Computer XT. For the system to spool to a serial printer, it must load the COM0x.SYS device driver during system initialization. This requires a DEVICE statement in your CONFIG.SYS file. For more information on this device driver, see "COM0x.SYS" on page 2-35.



START (Start Programs Automatically)

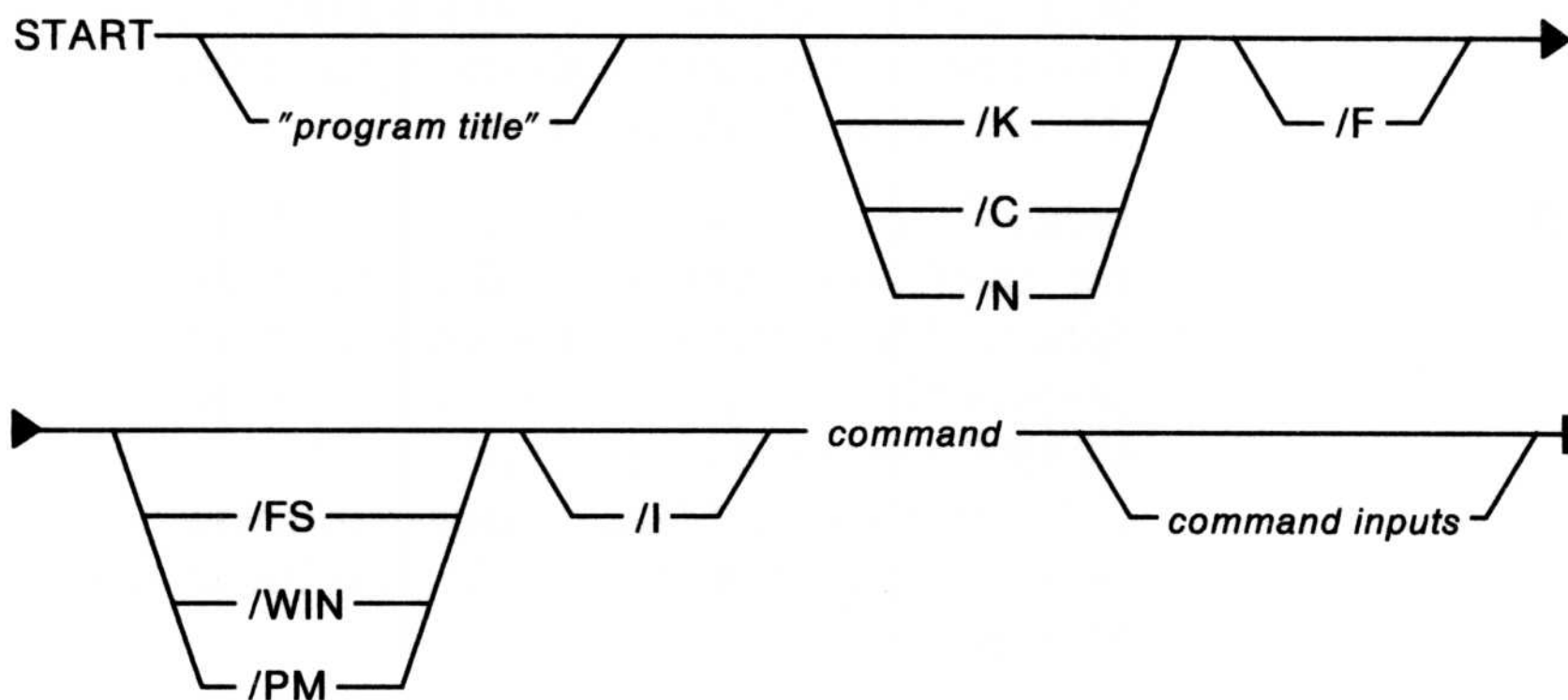
Purpose

Starts an OS/2 mode program in another session.

Note: The primary use for START is to automatically start programs at system startup. The special batch file, STARTUP.CMD, allows you to do this. For more information on using the STARTUP.CMD file, see *Creating and Changing Batch Files* in the "Using the Base Operating System" section of the *User's Guide*.

Syntax

To imbed redirectional signals into the command session, enclose the *command* and *command inputs* in quotes:



Parameters

"program title" Specifies the title you want for your program, which is displayed in the Presentation Manager menu. The program title must be enclosed in double quotes and cannot exceed 60 bytes.

If you do not specify *program title*, the file name of your program is displayed on the Presentation Manager menu.

START

(Start Programs Automatically)



Note: The use of a DBCS double-byte character counts as two bytes. If a title is longer than 60 bytes, it is shortened. DBCS means a set of characters in which each character is represented by two bytes. Languages such as Japanese, Chinese, and Korean, which contain more characters than can be represented by 256 code points, require double-byte character sets. As each character requires two bytes, the entering, displaying, and printing of DBCS characters requires hardware and software that can support DBCS.

/K Indicates to start the program indirectly through the command processor, CMD.EXE, with the /K option. This parameter keeps the session when the program has finished. If the application is a Presentation Manager application, the default is /N; otherwise, the default is /K.

/C Indicates to start the program indirectly through the command processor, CMD.EXE with the /C option. This switch ends the session when the *command* is complete. If the session created by the START command is displayed when the *command* ends, Presentation Manager determines which session to display next. Usually, it is Presentation Manager itself.

**(Start Programs Automatically)**

- /N** Indicates to start the program directly without invoking the command processor, **CMD.EXE**. When using the **/N** parameter, the command cannot be enclosed in quotes, be an internal command, or be a batch file.
- /F** Makes the program the foreground session. If this parameter is not specified, the program becomes a background session. If the **/FS**, **/WIN** or **/PM** parameter is specified, the program becomes the foreground session.

START attempts to determine the type of application and operate it accordingly. If the application is a DOS mode application, **START** outputs an error message. If **START** is unable to determine the application type, it starts the application as a full-screen application (**/FS**). However, the user has the option to override the determined default as follows:

- /FS** Indicates that this application is a full-screen application that must run in a separate session independent of Presentation Manager.

START

(Start Programs Automatically)



- /WIN** Indicates that this is a normal OS/2 application that runs within a Presentation Manager window.
- /PM** Indicates that this application is a Presentation Manager application.
- /I** Causes the new session to inherit the environment as defined by the SET statements in the CONFIG.SYS file instead of the current session's CMD.EXE environment.
- command* Specifies an OS/2 internal or external command, a .CMD batch file, or any OS/2 program you want to pass to the command processor you are starting.
- command inputs* Arguments for the command you requested.



SUBST (Substitute Drives)

Purpose

This is an external command that substitutes a drive letter for another drive and path.

Syntax

To substitute a drive letter for another drive and path:

```
drive path SUBST drive drive\path
```

To display drive substitutions currently in effect:

```
drive path SUBST
```

To delete drive substitutions currently in effect:

```
drive path SUBST drive /D
```

Parameters

drive drive\path

The first drive is the drive letter you want to substitute for the drive and path that follow.

/D

Deletes a substituted drive and path.

SUBST

(Substitute Drives)



Note: The following commands do not work in DOS mode on drives involved with ASSIGN, JOIN, or SUBST:

BACKUP

CHKDSK

DISKCOMP

DISKCOPY

FORMAT

LABEL

RECOVER

RESTORE.



Purpose

Displays the current version and level of corrective service for the OS/2 program features installed on your personal computer.

Syntax

SYSLEVEL

Remarks

A heading for each of the OS/2 program features is displayed followed by the OS/2 program version number, the current level of corrective service for the feature, and the previous level of corrective service for the feature.

For example:

```
IBM OS/2 Base Operating System  
Extended Edition 1.10      Component ID: 566933601  
Current CSD Level: WR00000  
Prior CSD Level: WR00000
```

In this example, no corrective service has ever been installed for the base operating system of the OS/2 program.

You may have to press the Enter key several times to view the complete list of features.



THREADS

(Determine Independent Actions)

Purpose

Determines the maximum number of independent actions, known as threads, for OS/2 mode.

Syntax

THREADS = _____ x _____

Parameters

x Specifies a numeric range from 32 through 255.

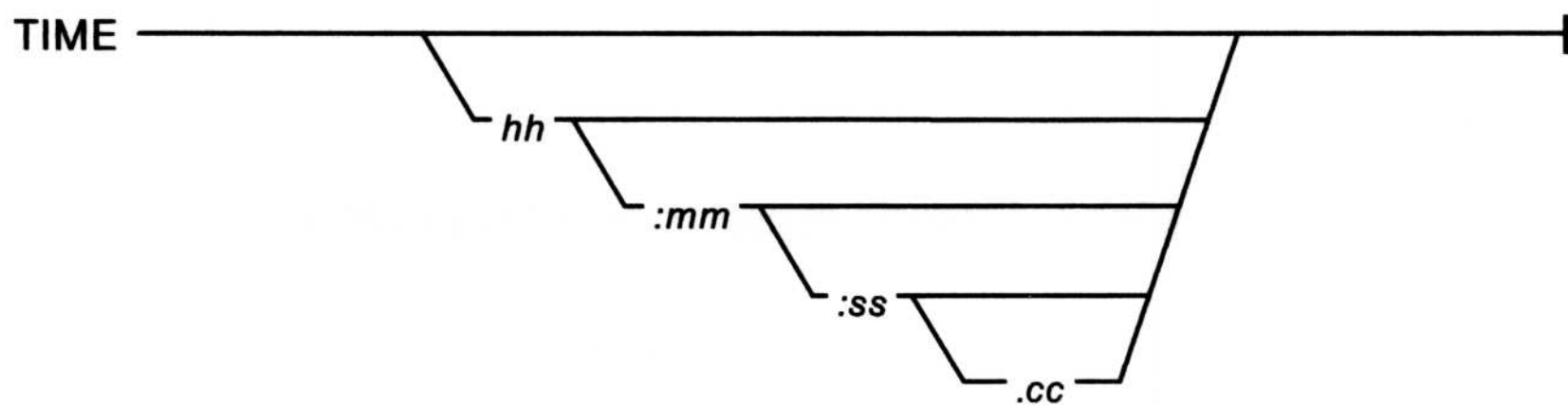
TIME

(Set System Time)

Purpose

Displays or changes the time known to the system and resets the time on your computer clock. This time is recorded in the directory when you create or change a file.

Syntax



Parameters

- hh* Specifies the hours. The new hours must be entered using 0 – 24 only.
- mm* Specifies the minutes. The new minutes must be entered using 0 – 59 only.
- ss* Specifies the seconds. The new seconds must be entered using 0 – 59 only.
- cc* Specifies the hundredths of seconds. The new hundredths of seconds must be entered using 0 – 99 only.

Note: Typing this command without a parameter displays the system time and asks if you want to change it.

TIMESLICE (Allocate Time)

Purpose

Sets the amount of processor time allocated to processes and programs for both modes.

Syntax

TIMESLICE = _____ x _____,y _____

Parameters

x Selects the minimum TIMESLICE value in milliseconds. This value must be an integer greater than or equal to 32.

y Selects the maximum TIMESLICE value in milliseconds. This value must be an integer greater than or equal to the minimum value and less than 65536.

Note: If *y* is not specified, the default is equal to *x*.

TRACE

(Select or Set System Trace)

Purpose

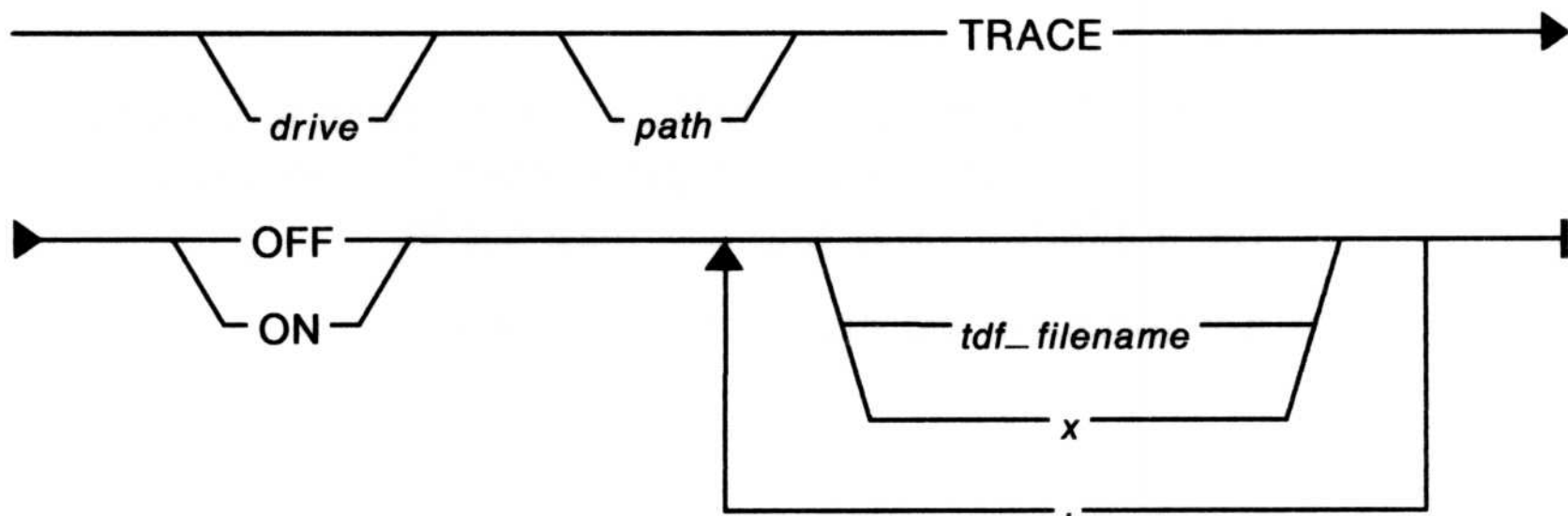
Select or set the tracing of system events.

Note: This command is intended for use with aid from an IBM service representative. For more information, see the "Using the Base Operating System" section of the *User's Guide*.

Syntax

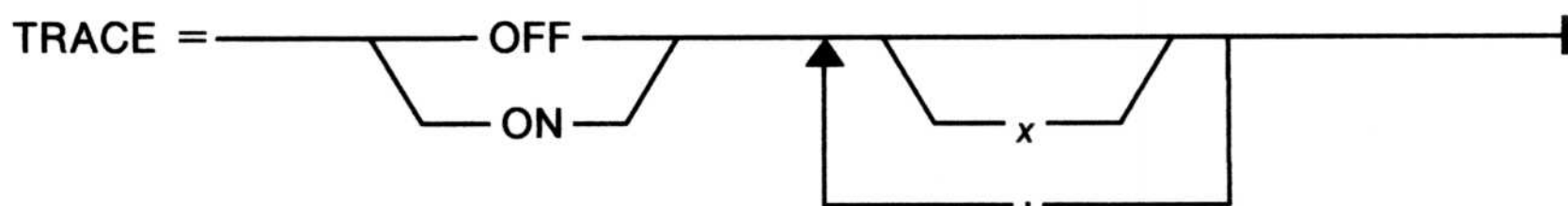


To select system trace from the command prompt:



Syntax

To allow system trace in your CONFIG.SYS file:



TRACE

(Select or Set System Trace)

Parameters

<i>OFF</i>	Turns OFF module trace points.
<i>ON</i>	Turns ON module trace points.
<i>x</i>	Specifies a number from 0 through 255, indicating the major event code to be traced or <i>not</i> to be traced. The numbers for these codes are supplied by your IBM service representative.
<i>tdf_filename</i>	Specifies the name of a trace definition file containing dynamic trace points.
<i>, (comma)</i>	Used as a separator for repetitive variable parameters.

Note: If TRACE or TRACEBUF is not specified in your CONFIG.SYS file, this command does nothing. Message 1397 is displayed:

System Trace information was not specified in your CONFIG.SYS file.

TRACEBUF

(Set Size of Trace Buffer)

Purpose

Sets the size of the circular trace buffer.

Note: This statement is intended for use with aid from an IBM service representative. For more information, see Problem Determination and System Software Repair in the “Using the Base Operating System” section of the *User's Guide*.

Syntax

TRACEBUF = _____ x _____

Parameters

x Specifies a number from 1 through 63, representing a multiple of 1024 bytes.



TRACEFMT

(Display Formatted Trace Records)

Purpose

This external command displays formatted trace records in reverse time stamp order.

Note: This command is intended for use with aid from an IBM service representative. For more information, see *Problem Determination and System Software Repair* in the “Using the Base Operating System” section of the *User's Guide*.

Syntax



Examples

You have already entered a TRACE statement that has been tracing system events recommended by your IBM service representative. To format the trace records contained in the trace buffer and have the formatted output printed on your printer, type the following at the OS/2 command prompt and press the Enter key:

```
TRACE OFF
```

Type the following and press the Enter key:

```
TRACEFMT >LPT1
```

Note: Typing this command without a parameter displays the contents of the system trace buffer. The information is scrolled on the screen if system tracing is available.

TREE

(Display Directory Structure)

Purpose

This external command displays all the directory paths found on the specified drive and optionally lists the files in the root directory and in each subdirectory.

Syntax



Parameters

/F Displays the names of files in the root directory and in all the subdirectories.

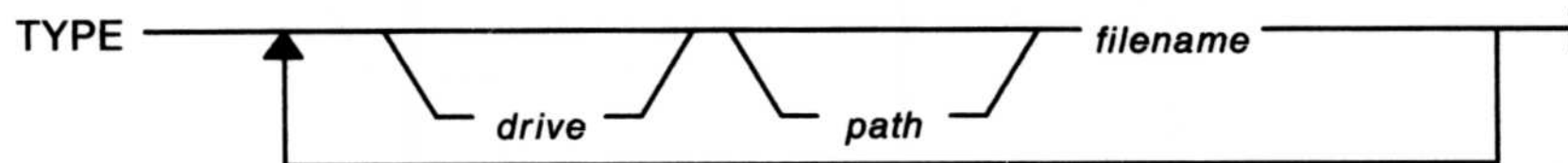
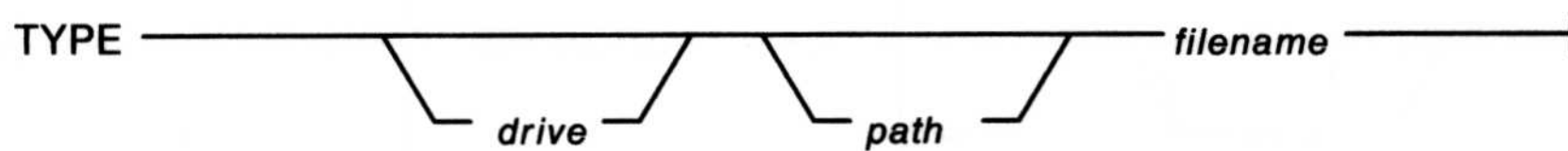
Note: Typing this command without a parameter displays all directory paths found on the current drive.

TYPE (Display File Contents)

Purpose

Displays the contents of a file on the screen.

Syntax



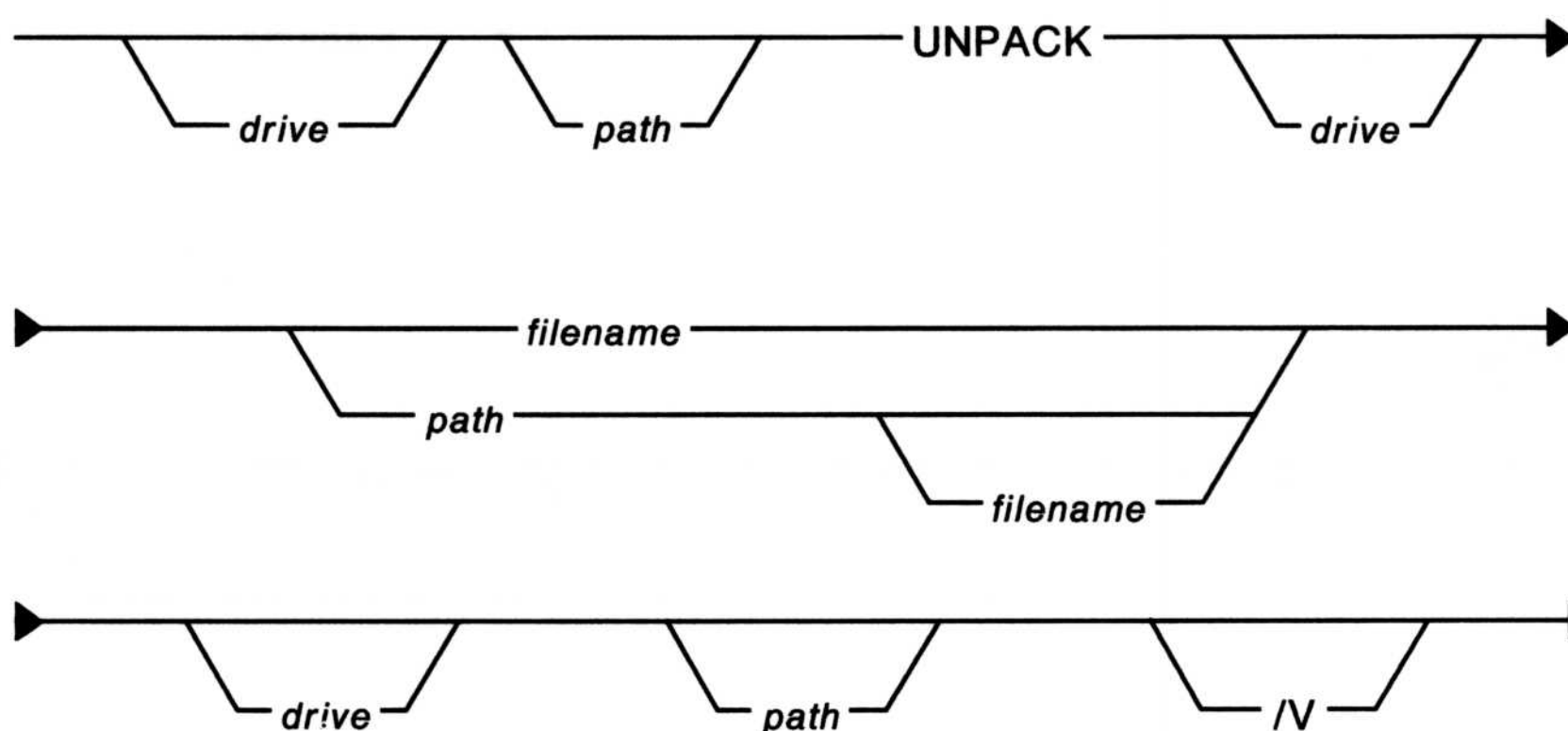
UNPACK

(Decompress a File)

Purpose

Decompresses files that have been compressed on the shipped diskette. Compressed files are designated by an @ in their file extension. Files that are not compressed are copied.

Syntax



Parameters

- filename* A valid OS/2 file specification. This is required for either an existing compressed or non-compressed file.
- /V* Verifies that data written to a disk has been correctly written. This option causes UNPACK to run slower.

VER

(Display OS/2 Version Number)

Purpose

Displays the OS/2 version number.

Syntax

VER _____|

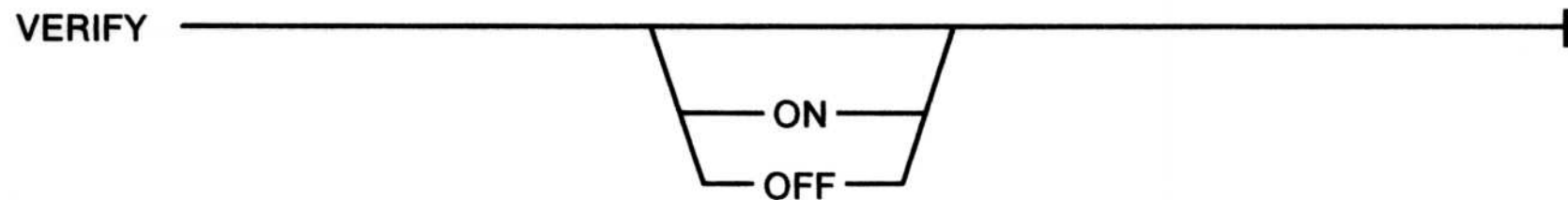
VERIFY

(Set Data Verification)

Purpose

Confirms that data written to a disk is correctly written.

Syntax



Parameters

ON Verifies that your files are written correctly to the disk.

OFF Turns verification OFF. This is the default value.

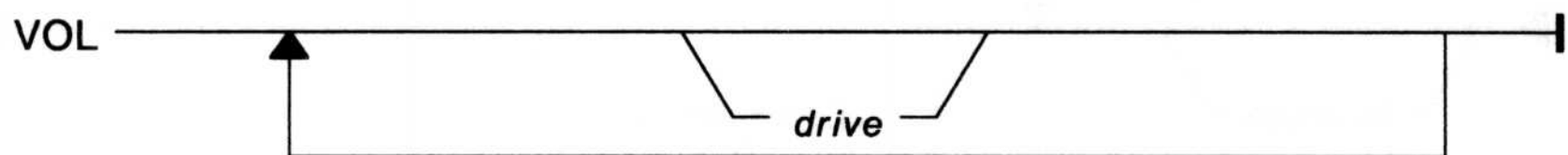
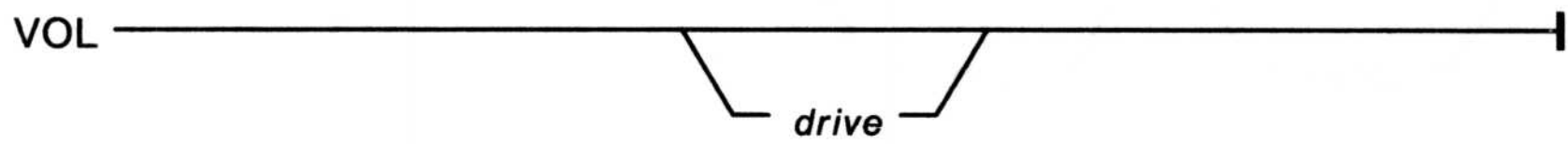
Note: Typing this command without a parameter displays the current status of on or off.

VOL (Display Disk Volume Label)

Purpose

Displays the disk volume label if it exists.

Syntax



Note: Typing this command without a parameter displays the volume label of the current drive.

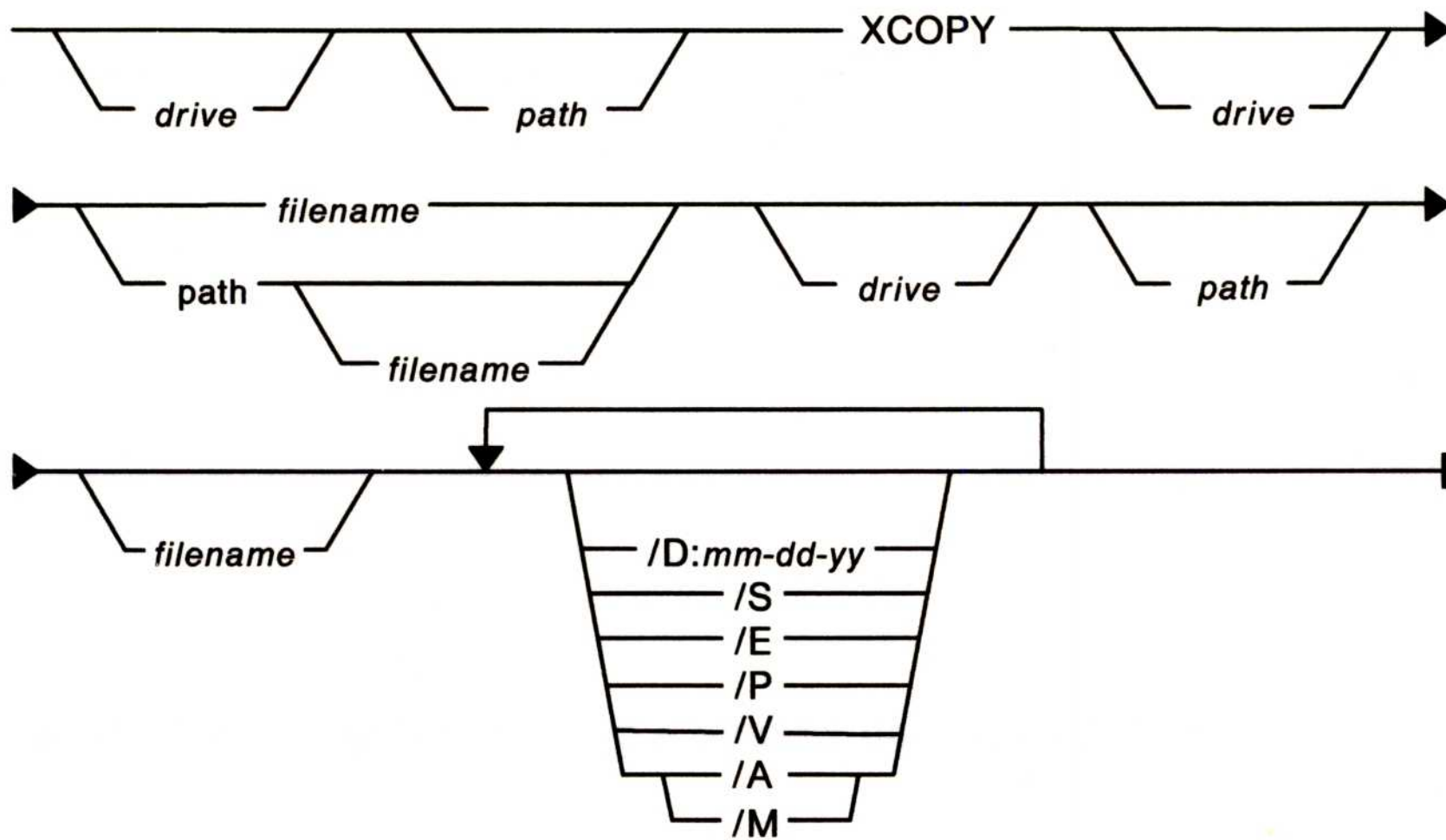
XCOPY

(Copy Subdirectories)

Purpose

This external command selectively copies groups of files, which can include lower-level subdirectories.

Syntax



Parameters

`/D:mm-dd-yy` Copies files that have been modified on or after the specified date. The date format varies depending on the `COUNTRY =` statement in the `CONFIG.SYS` file. The date formats are as follows:

`/D:mm-dd-yy`

`/D:dd-mm-yy`

or

`/D:yy-mm-dd`

XCOPY **(Copy Subdirectories)**

- /S** Copies subdirectories. Without this parameter, XCOPY works only within a directory. This parameter does not create the corresponding source directory on the target if no files have been found on the source.
- /E** When used with the /S parameter, this copies the entire tree structure while creating corresponding source directories on the target even if no files have been found on those source directories.
- /P** Prompts you before performing a physical copy. You respond to a (Y/N)? prompt for a selective copy. With this parameter, XCOPY copies on a file-by-file basis.
- /V** Verifies that data written to a disk has been correctly written. This option causes XCOPY to run slower.
- /A** Copies archived files only. This parameter does not turn off the archive bit of the source file.
- /M** Copies archived files only. Unlike /A, this parameter turns off the archive bit of the source file. So when you regularly back up the source disk with XCOPY, using this parameter increases the efficiency of copying. You should be aware that the BACKUP command also uses the archive bit. If both the /A and /M parameters are specified, the one specified last is the one used.

Chapter 3. Communications Manager Commands

In addition to using the menu interface to perform Communications Manager functions, you can use the OS/2 full-screen command prompt to issue commands to start certain functions. You cannot use these commands from the DOS command prompt or the windowed command prompt.

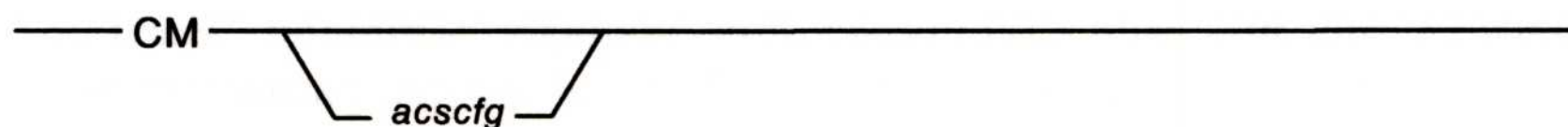
This chapter lists each Communications Manager command. Each command description has a purpose, format or syntax, parameter list, example (including steps for using the command), and explanatory notes or hints, where appropriate. For a detailed description of the syntax diagrams, see Appendix A.



Purpose

Starts the command file that contains the commands necessary to start Communications Manager from the OS/2 full-screen command prompt.

Syntax



Parameters

acscfg Name of the configuration file that you want to activate when you start Communications Manager. If you do not specify this parameter, Communications Manager uses the configuration file that you used the last time you started Communications Manager. Do not include the file extension. In the following example, **MYPRO** is the name of the configuration file that you want to activate.

Note: For starting Communications Manager from the Start Programs window, see page 3-3.

Examples

To start Communications Manager from the OS/2 command prompt:

1. Type the CM command and optionally the configuration file name at the OS/2 command prompt. For example:

```
CM MYPRO
```

2. Press the Enter key.



CM (Start Communications Manager)

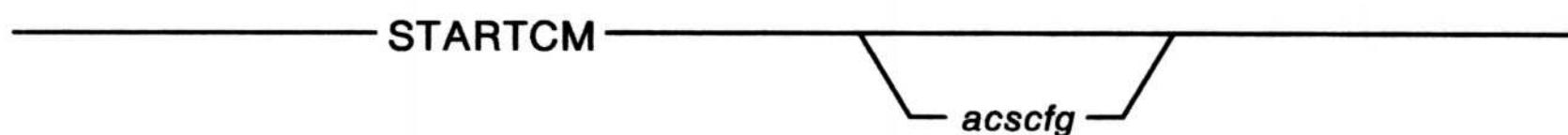
STARTCM

Purpose

Starts the command file that contains the commands necessary to start Communications Manager from the Start Programs window.

Syntax

STARTCM *acscfg*



Parameters

acscfg Name of the configuration file that you want to activate when you start Communications Manager. If you do not specify this parameter, Communications Manager uses the configuration file that you used the last time you started Communications Manager. Do not include the file extension.

Note: For starting Communications Manager from the OS/2 full-screen command prompt, see page 3-2.

CM (Start Communications Manager)



Examples

In the Start Programs window, the following values are set for Communications Manager:

Program Title: Communications Manager

Path and File Name: C:\OS2\CMD.EXE

Parameters: /K C:\CMLIB\STARTCM.CMD

Working Directory: C:\CMLIB

Presentation Manager application: No

How should it be run?: Run on full screen

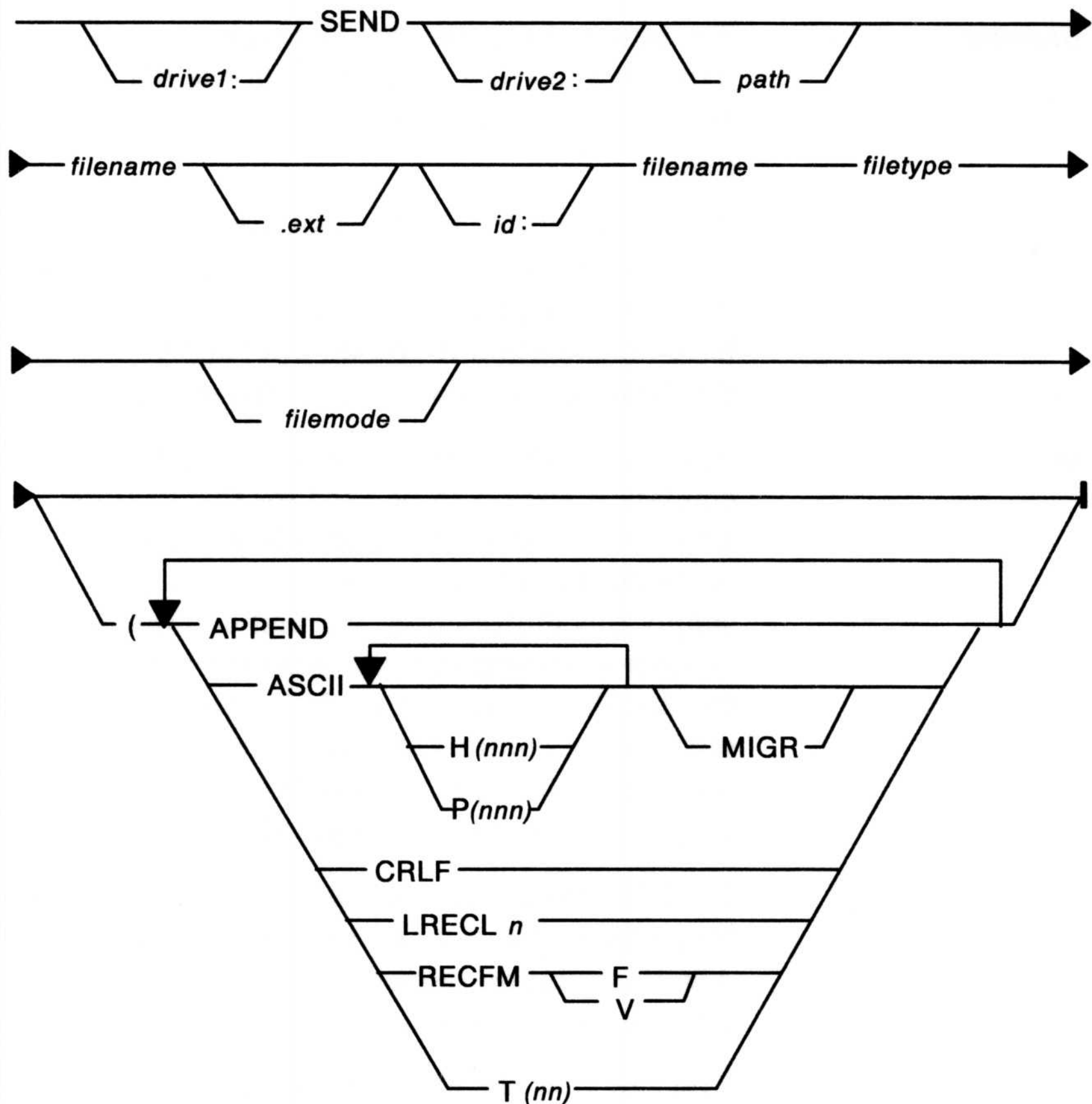


SEND for VM Host

Purpose

Sends a file to an IBM VM host computer.

Syntax



Parameters

drive1: The drive where the SEND.EXE file resides. Specify this only if it is not in the default drive.

SEND for VM Host



- drive2:* The drive where the workstation file that you want to send to the host resides. Specify this only if it is not in the default drive. In the following example, **a:** is the drive.
- path* Path designator that specifies where the file is. Use this only if the file is not in the default path.
- filename* The name of the file that you want to send. You are allowed 8 characters for the file name. In the following example, **report** is the file name.
- .ext* The extension for the file that you want to send. You are allowed 4 characters for the extension, including the period. The file name will not have an extension if you do not supply one. In the following example, **.dat** is the extension.
- id:* The 1- to 8-character name that identifies the emulation session to be used for the 3270 file transfer, or the short session identifier you specified during configuration. If you do not supply this information, the first configured emulation session that has been started is used for the file transfer. If no 3270 emulation sessions are configured and started, the file transfer facility uses an ASCII terminal emulation session, if configured.
- For 3270 emulation sessions, **id** can be either a long session ID or a short session ID. For ASCII emulation sessions, **id** must be the currently active session.
- You can view the identifiers in the configuration file and from the File Transfer Operations Menu.
- In the following example, **stat2** is the ID. Note that the ID is separated from the file name by a colon (:).



SEND for VM Host

filename

The name of the file on VM into which you want to receive the data. In the following example, **report** is the file name.

filetype

The VM file type of the file that you want to transfer. In the preceding example, **script** is the file type.

Do not specify the words **APPEND**, **ASCII**, or **CRLF** as a file type. If any of these words is specified as a file type, Communications Manager performs the action described for that option in the following option descriptions.

Also, do not specify the terms **P(*nnn*)** or **H(*nnn*)** as a file type. If one of these terms is specified as a file type and the **ASCII** option is also specified, the host returns an error. If one of these terms is specified as a file type without the **ASCII** option, the host ignores the file type but returns no errors.

filemode

The VM file mode of the file that you want to transfer. This is an optional parameter. If this parameter is omitted, the default is **a1**. In the following example, **a** is the file mode.

Options

The following parts of the command are the options. The options are separated from the last parameter by a required left parenthesis with one space between each option. In the example, **CRLF ASCII** are the options.

APPEND

Allows you to attach the workstation file to the end of an existing host file. The **APPEND** option overrides any other values specified for **LRECL** and **RECFM**.

Warning: If **append** is not specified and a host file already exists with the same name, the host file is replaced.



ASCII

Causes the host program to translate workstation files from ASCII to EBCDIC so that the file is readable to the host. When sending text files to the host, specify the **ASCII** option. If the workstation file is already in EBCDIC form, do not supply this option. Do not use this option when you are transferring a binary file.

H(*nnn*)

Specifies the host file code page to be used for the host file being received, where *nnn* is the code page number. Choices are **037**, **273**, **277**, **278**, **280**, **284**, **285**, **297**, or **500**. If you do not specify a code page, the system uses the code page for the host character set of the emulation session that is specified in the keyboard remap profile.

The host character set will also specify the file transfer command program to be used.

Note: This parameter is valid only when the ASCII option is used.

P(*nnn*)

Specifies the workstation file code page to be used for the workstation file being sent, where *nnn* is the code page number. Choices are **437**, **850**, **860**, **863**, or **865**. If you do not specify a code page, the system uses the active code page for the command line screen group.

Note: This parameter is valid only when the ASCII option is used.



SEND for VM Host

MIGR

Allows you to send or receive files that were transferred using Communications Manager Version 1.0. This only applies to files that were transferred in one direction using the ASCII option. Use this option to transfer the file back to the source location.

This option is valid only if the ASCII option is also specified. If MIGR is specified but the ASCII option is not, MIGR is ignored.

CRLF

Causes the host program to delete carriage return and line feed characters from the workstation file before storing it at the host. When sending an ASCII or EBCDIC file to the host, it is recommended that you supply the **CRLF** option so the file will be readable when on the host.

Do not use this option when transferring a binary file.

LRECL *n*

Provides the logical record length of the host file, where *n* is the number of characters in each record. The default value for new files is **80**. If you are replacing an existing host file, its record length is used as the record length of the new version. If you are appending a file, the current record length of the file is used and the **LRECL** value specified is ignored. When using with variable-length records, this is the maximum record size the host will accept. However, if you do not send a record of the maximum size, the logical record length will be reduced to the longest record actually sent. Do not use this option when transferring a binary file.

RECFM F/V

Specifies the record format and characteristics of the host file.

- f** specifies that the file contains fixed-length records
- v** specifies that the file contains variable-length records



For new files, the default is *f*, unless the CRLF option is specified, in which case *v* is the default.

T(*nn*)

Specifies the period of time that the file transfer program waits for the host to respond before the file transfer request is cancelled. *nn* is a two-digit integer in a range from 00 through 99. *nn* indicates the number of 20-second increments the file transfer program waits for the host to respond. If **T(*NN*)** is not specified, **T(00)** is the default, and the file transfer program waits indefinitely for the host to respond.

While waiting for the host to respond, the file transfer program sends a message to the screen every 30 seconds that the host has not responded.

Examples

To send an ASCII file to an IBM VM host from an OS/2 command prompt on an active session:

1. Type the SEND command with the necessary parameters at an OS/2 full-screen command prompt. For example:

```
send a:report.dat stat2:report script a (CRLF ASCII
```

2. Press the Enter key.



Remarks

- You can also type the SEND command in a command file (.CMD) and run that file only from the OS/2 full-screen command prompt.
- You can transfer more than one file at a time as long as each file transfer is on a different emulation session and is initiated from a different OS/2 full-screen command prompt. The maximum number of concurrent file transfers is nine.
- To issue multiple file transfers, specify a different emulation session for each file transfer with the ID in the file transfer command. Note that the emulation session corresponding to the ID that you specify must be started and you must be logged on and ready to receive a command.
- If your host file transfer program name is not IND\$FILE, use the OS/2 SET command before issuing the file transfer command. To use the SET command, type:

```
SET IND_FILE=xxxxxxx
```

where `xxxxxxx` is your host file transfer program name.

It is recommended you use the OS/2 SET command in the CONFIG.SYS file to avoid having to do this each time.

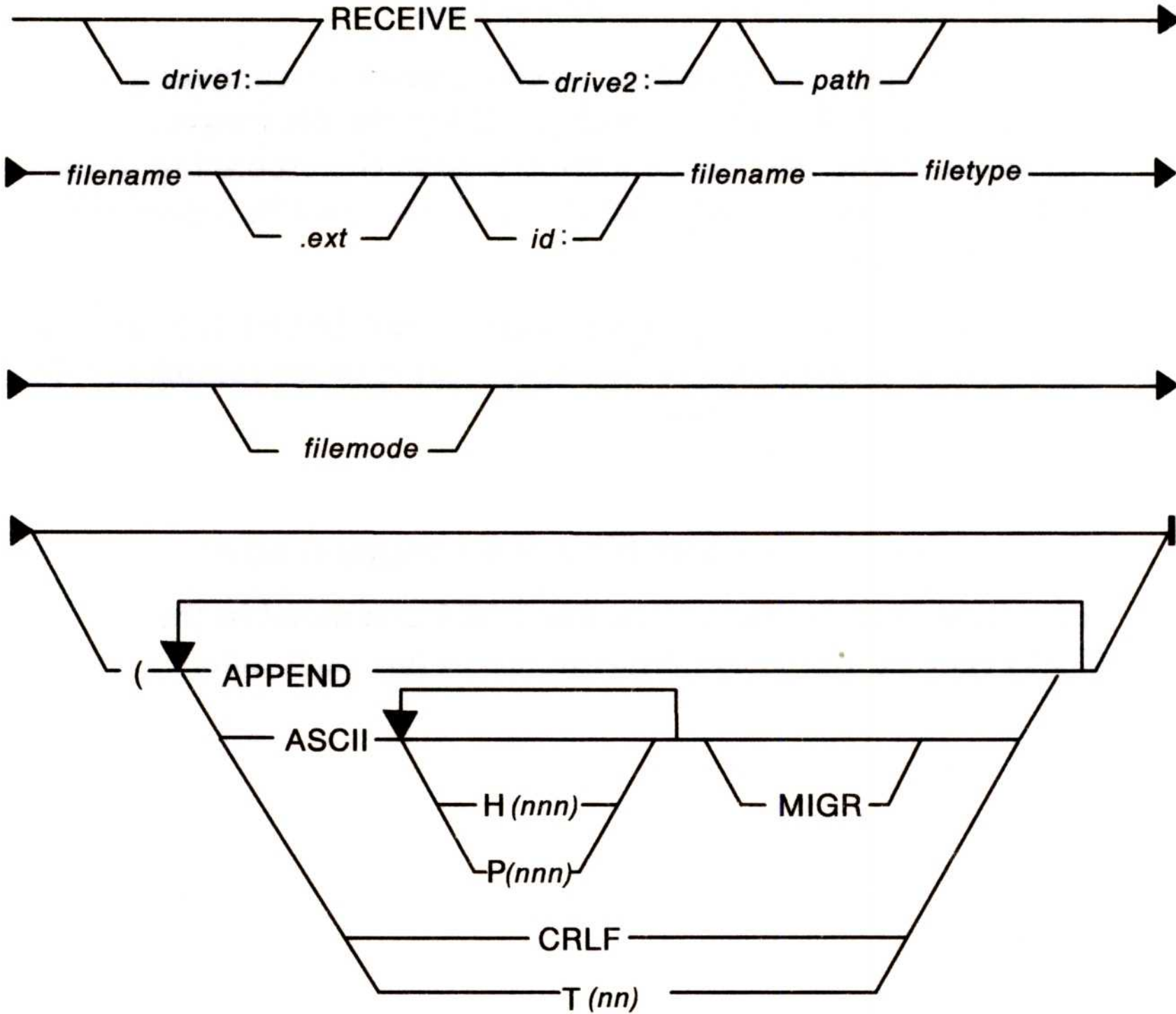
RECEIVE for VM Host



Purpose

Receives a file from an IBM VM host computer.

Syntax



Parameters

- drive1:* The drive where the RECEIVE.EXE file resides. Specify this only if it is not in the default drive.
- drive2:* The drive where you want the file that you are receiving from the host to reside. Specify this only if it is not in the default drive. In the following example, **a:** is the drive.



RECEIVE for VM Host

- path* Path designator that specifies where the file is to reside. Use this only if it will not be in the default path.
- filename* The name of the workstation file that is to receive the host data. You are allowed 8 characters for the file name. In the preceding example, **report** is the file name.
- .ext* The extension for the workstation file that is to receive the data. You are allowed 4 characters for the extension, including the period. The file name will not have an extension if you do not supply one. In the following example **.dat** is the extension.
- id:* The 1- to 8-character name that identifies the emulation session to be used for the file transfer, or the short session identifier you specified during configuration. If you do not supply this information, the first configured emulation session will be used for the file transfer. If no 3270 emulation sessions are configured, the file transfer facility will use an ASCII terminal emulation session, if configured.
- For 3270 emulation sessions, **id** can be either a long session ID or a short session ID. For ASCII emulation sessions, **id** must be the currently active session.
- You can view the identifiers in the configuration file and from the File Transfer Main Menu.
- In the following example, **stat2** is the ID. Note that the ID is separated from the file name by a colon (:).
- filename* The VM file name of the file that you want to transfer. In the following example, **report** is the file name.



filetype The VM file type of the file that you want to transfer. In the following example, `script` is the file type.

Do not specify the words `APPEND`, `ASCII`, or `CRLF` as a file type. If any of these words is specified as a file type, Communications Manager performs the action described for that option in the following option descriptions.

Also, do not specify the terms `P(nnn)` or `H(nnn)` as a file type. If one of these terms is specified as a file type and the `ASCII` option is also specified, the host returns an error. If one of these terms is specified as a file type without the `ASCII` option, the host ignores the file type but returns no errors.

filemode The VM file mode of the file that you want to transfer. This is an optional parameter. If this parameter is omitted, the default is `a1`. In the following example, `a` is the file mode.

Options

The following parts of the command are the options. The options are separated from the last parameter by a required left parenthesis with one space between each option. In the example, `(CRLF` and `ASCII` are the options.

APPEND

Allows you to attach the information from the host file to the end of an existing workstation file. The append option overrides any other values specified for `LRECL` and `RECFM`

Warning: If append is not specified and a workstation file already exists with the same name, the file is replaced with the received file.



RECEIVE for VM Host

- ASCII** Causes the host program to translate the host file from EBCDIC to ASCII so that the file is readable on the workstation. When receiving text files from the host, specify the **ASCII** option. If the host file is already in ASCII form, do not supply this option.
- H(*nnn*)** Specifies the host file code page to be used for the host file being sent, where *nnn* is the code page number. Choices are **037**, **273**, **277**, **278**, **280**, **284**, **285**, **297**, or **500**. If you do not specify a code page, the system will use the code page for the host character set of the emulation that is specified in the keyboard remap profile.
- The host character set will also specify the file transfer command program to be used.
- P(*nnn*)** Specifies the workstation file code page to be used for the workstation file being received, where *nnn* is the code page number. Choices are **437**, **850**, **860**, **863**, or **865**. If you do not specify a code page, the system uses the code page for the active command line screen group.
- MIGR** Allows you to send or receive files that were transferred using Communications Manager Version 1.0. This only applies to files that were transferred in one direction using the **ASCII** option. Use this option to transfer the file back to the source location.
- This option is valid only if the **ASCII** option is also specified. If **MIGR** is specified but the **ASCII** option is not, **MIGR** is ignored.

RECEIVE for VM Host



CRLF

Causes the host program to insert carriage return and line feed characters as the last two characters of each line before sending it to the workstation.

T(*nn*)

Specifies the period of time that the file transfer program waits for the host to respond before the file transfer request is cancelled. *nn* is a two-digit integer in a range from 00 through 99. *nn* indicates the number of 20-second increments the file transfer program waits for the host to respond. If `T(NN)` is not specified, `T(00)` is the default, and the file transfer program waits indefinitely for the host to respond.

While waiting for the host to respond, the file transfer program sends a message to the screen every 30 seconds that the host has not responded.

Examples

To receive a file from an IBM VM host from an OS/2 command prompt:

1. Type the RECEIVE command with the necessary parameters at an OS/2 full-screen command prompt. For example:

```
receive a:report.dat stat2:report script a (CRLF ASCII
```

2. Press the Enter key.

Remarks

- You can also type the RECEIVE command in a command file and run that file only from the OS/2 full-screen command prompt.
- You can transfer more than one file at a time as long as each file transfer is on a different emulation session and is initiated in a separate screen group. The maximum number of concurrent file transfers is nine.



RECEIVE for VM Host

- To issue multiple file transfers, specify a different emulation session for each file transfer with the ID in the file transfer command. Note that the emulation session that corresponds to the ID that you specify must be started and ready to receive a command.
- If your host file transfer program name is not IND\$FILE, use the OS/2 SET command before issuing the file transfer command. To use the SET command, type:

```
SET IND_FILE=xxxxxxx
```

where xxxxxxx is your host file transfer program name. It is recommended you use the OS/2 SET command in the CONFIG.SYS file to avoid having to do this each time.

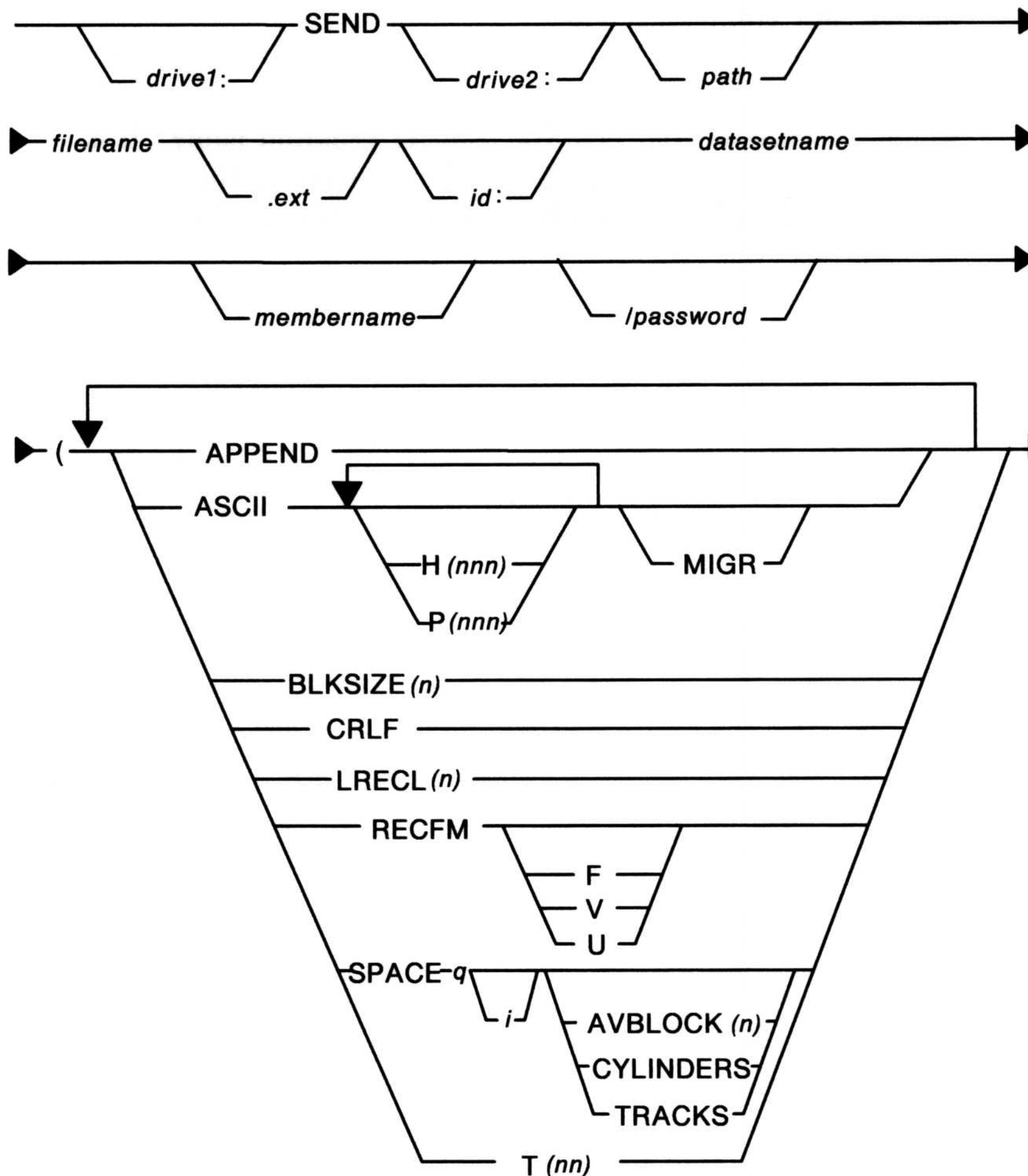
SEND for TSO Host



Purpose

Sends a file to an IBM TSO host computer.

Syntax



Parameters



SEND for TSO Host

- drive1:* The drive where the SEND.EXE file resides. Specify this only if it is not in the default drive.
- drive2:* The drive where the workstation file that you want to send to the host resides. Specify this only if it is not in the default drive. In the following example, **a:** is the drive.
- path* Path designator that specifies where the file is. Use this only if the file is not in the default path.
- filename* The name of the workstation file that you want to send. You are allowed 8 characters for the file name. In the following example, **myfile** is the file name.
- .ext* The extension of the workstation file that you want to send. You are allowed 4 characters for the extension, including the period. In the following example, **.txt** is the extension.
- id:* The 1- to 8-character name that identifies the emulation session to be used for the file transfer, or the short session identifier you specified during configuration. If you do not supply this information, the first configured emulation session will be used for the file transfer. If no 3270 emulation sessions are configured, the file transfer facility will use an ASCII terminal emulation session, if configured.
- You can view the identifiers in the configuration file and from the File Transfer Main Menu.
- In the following example, **stat2** is the ID. Note that the ID is separated from the data set name by a colon (:).
- datasetname* The TSO data set name for the file that you are sending. To show that the user ID is not prefixed, you can enter the data set name and member name together and enclose the information in quotation marks. In the following example, **boss** is the data set name.

SEND for TSO Host



(membername) The name of one of the members of a partitioned data set. The data set must already exist. In the following example, *(bosname)* is the member name. Note that the member name is enclosed in parentheses, ().

/password The password that allows access to password-protected TSO data sets. Required only if the data set is password-protected. In the following example, */gky5hf* is the password. Note that the password is preceded by a slash (/).

Options

The following parts of the command are the options. The options are separated from the last parameter with one space and one additional space between each option. In the example, *ASCII* is the option.

APPEND Allows you to attach the workstation file to the end of an existing host data set. The *APPEND* option overrides any other values specified for *LRECL* and *RECFM*.

You may not specify *append* for members of a partitioned data set.

ASCII Causes the host program to translate workstation files from ASCII to EBCDIC so that the file is readable to the host. When sending text files to the host, always specify the *ASCII* option. If the workstation file is already in EBCDIC form, do not supply this option.

H(*nnn*) Specifies the host file code page to be used for the host file being received, where *nnn* is the code page number. Choices are *037*, *273*, *277*, *278*, *280*, *284*, *285*, *297*, or *500*. If you do not specify a code page, the system will use the code page for the host character set of the emulation that is specified in the keyboard remap profile.

The host character set will also specify the file transfer command program to be used.



SEND for TSO Host

Note: This parameter is valid only when the ASCII keyword is used.

P(*nnn*)

Specifies the workstation file code page to be used for the workstation file being sent, where *nnn* is the code page number. Choices are 437, 850, 860, 863, or 865. If you do not specify a code page, the system will use the code page for the active command line screen group.

Note: This parameter is valid only when the ASCII keyword is used.

MIGR

Allows you to send or receive files that were transferred using Communications Manager Version 1.0. This only applies to files that were transferred in one direction using the ASCII option. Use this option to transfer the file back to the source location.

This option is valid only if the ASCII option is also specified. If MIGR is specified but the ASCII option is not, MIGR is ignored.

BLKSIZE(*n*)

Specifies the size of the host data set, where (*n*) is the length in bytes of a data block. If you omit BLKSIZE, the default is LRECL for new files. If you are replacing or appending to a file, BLKSIZE is ignored.

CRLF

Causes the host program to delete carriage return and line feed characters from the workstation file before storing it at the host. When sending an ASCII or EBCDIC file to the host, it is recommended that you always supply the CRLF option so the file will be readable when on the host.



LRECL (*n*)

Provides the logical record length of the host data set, where (*n*) is the number of characters in each record. The host default value is 80 for new data sets. If you are replacing an existing data set, its record length is used as the record length of the new version. If you are appending a data set, the current record length of the data set is used and the LRECL value specified is ignored. When using variable-length records, this is the maximum size record the host will accept. However, if you do not send a record of the maximum size, the logical record length will be reduced to the longest record actually sent.

RECFM F/V/U

Specifies the record format and characteristics of the host file. RECFM is followed by one of the following characters:

- f specifies that the data set contains fixed-length records
- v specifies that the data set contains variable-length records
- u specifies that the data set contains undefined-length records.

For new files, the default is f, unless the CRLF option is specified, in which case v is the default.

SPACE(*q*[,*i*])

Specifies the amount of space allocated for a new data set, where *q* is quantity of space allocated initially for a data set and *i* is the increment of space added to the data set each time the previously allocated space is filled.



SEND for TSO Host

If you use **SPACE**, you can use one of these three options to specify the units used for quantity and increment:

AVBLOCK(*n*) specifies the average block length of the records that will be written to the data set, where *n* is the desired average block length in bytes

TRACKS specifies that the unit of space is to be a track

CYLINDERS specifies that the unit of space is to be a cylinder.

Examples

To send a file to an IBM TSO host from an OS/2 full-screen command prompt:

1. Type the **SEND** command with the necessary parameters at an OS/2 command prompt. For example:

```
send a:myfile.txt stat2:boss(bossname)/gky5hf ASCII
```

2. Press the Enter key.

Remarks

- You can also type the **SEND** command in a command file and run that file only from the OS/2 full-screen command prompt.
- You can transfer more than one file at a time as long as each file transfer is on a different emulation session and is initiated in a separate screen group. The maximum number of concurrent file transfers is nine.
- To issue multiple file transfers, specify a different emulation session for each file transfer with the ID in the file transfer command. Note that the emulation session that corresponds to the ID that you specify must be started and ready to receive a command.
- The options **AVBLOCK**, **CYLINDERS**, **TRACKS**, **BLKSIZE**, **RECFM**, and **SPACE** are only valid for physical sequential data sets.

SEND for TSO Host



- The options `AVBLOCK`, `CYLINDERS`, and `TRACKS` are mutually exclusive.
- If neither `AVBLOCK`, `CYLINDERS`, or `TRACKS` is specified, the value specified for `BLKSIZE` will be used as the space allocation unit.
- If your host file transfer program name is not `IND$FILE`, use the OS/2 SET command before issuing the file transfer command. To use the SET command, type:

```
SET IND_FILE=xxxxxxx
```

where `xxxxxxx` is your host file transfer name. It is recommended you use the OS/2 SET command in the `CONFIG.SYS` file to avoid having to do this each time.

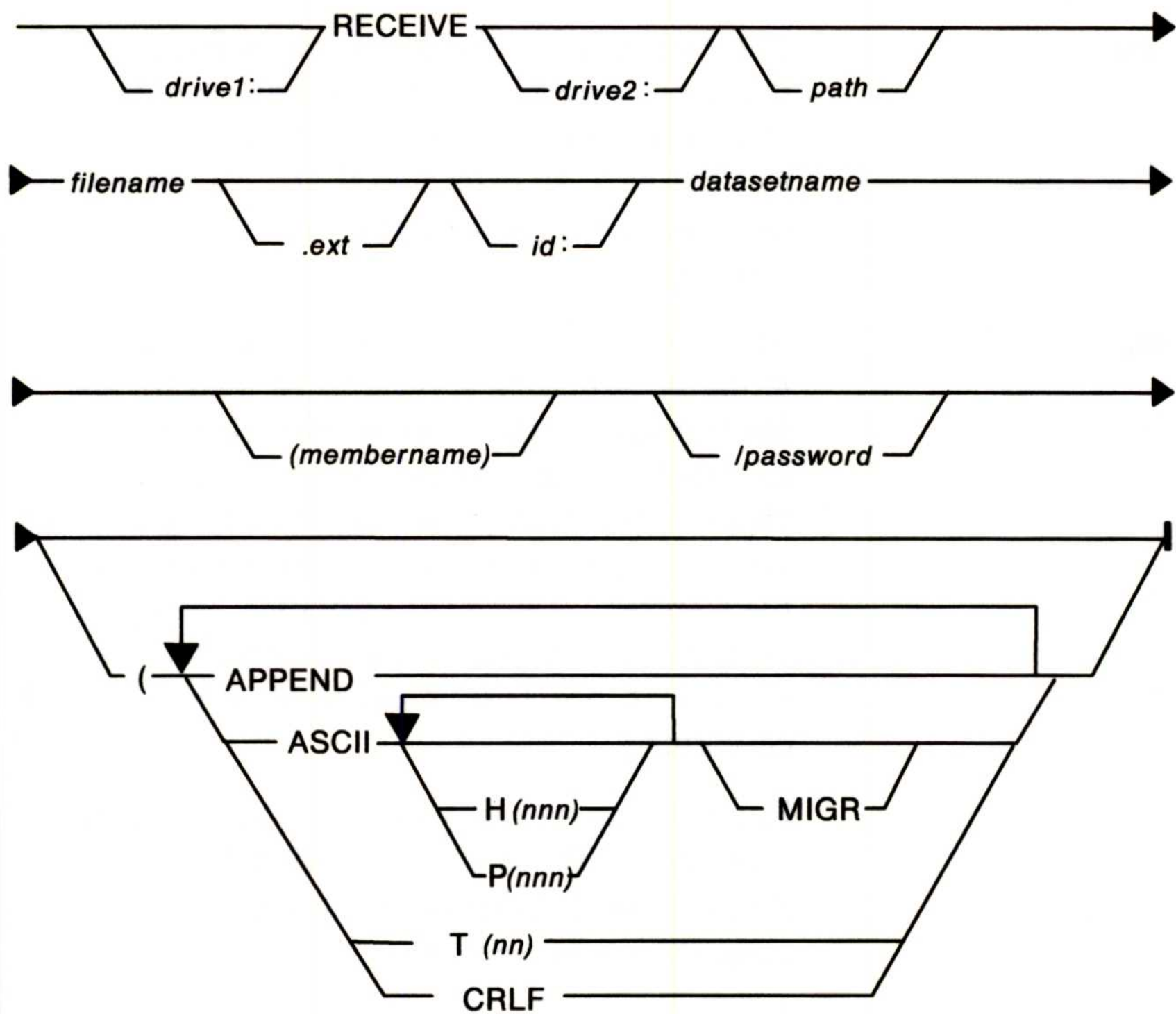


RECEIVE for TSO Host

Purpose

Receives a file from an IBM TSO host computer.

Syntax



Parameters

- drive1:** The drive where the RECEIVE.EXE file resides. Specify this only if it is not in the default drive.
- drive2:** The drive where you want the data set that you are receiving from the host to reside. Specify this only if it is not in the default drive. In the following example, **a:** is the drive.



- path** Path designator that specifies where the data set is to reside. Use this only if it will not be in the default path.
- filename** The name of the file being received from the host. You are allowed 8 characters for the file name. In the preceding example, `myfile` is the file name.
- .ext** The extension for the file being received from the host. You are allowed 4 characters for the extension, including the period. The file name will not have an extension if you do not supply one. In the following example, `.txt` is the extension.
- id:** The 1- to 8-character name that identifies the terminal emulation session to be used for the file transfer, or the short session identifier you specified during configuration. If you do not supply this information, the first configured and started emulation session will be used for the file transfer. If no 3270 emulation sessions are configured and started, the file transfer facility will use an ASCII terminal emulation session, if configured.
- For 3270 emulation sessions, `id` can be either a long session ID or a short session ID. For ASCII emulation sessions, `id` must be the currently active session.
- You can view the identifiers in the configuration file and from the File Transfer Operations Menu.
- In the example above, `stat2` is the ID. Note that the ID is separated from the data set name by a colon (:).



RECEIVE for TSO Host

<code>datasetname</code>	The name of the TSO data set being received. To show that the user ID is not prefixed, you can enter the data set name and member name together and enclose the information in quotation marks. In the following example, <code>boss</code> is the data set name.
<code>(membername)</code>	The name of one of the members of a partitioned data set. The data set must already exist. In the preceding example, <code>(bossname)</code> is the member name. Note that the member name is enclosed in parentheses, ().
<code>/password</code>	The password that allows access to password-protected TSO data sets. Required only if the data set is password-protected. In the following example, <code>/gky5hf</code> is the password. Note that the password is preceded by a slash (/).

Options

The following parts of the command are the options. The options are separated from the last parameter with one space and one additional space between each option. In the example, `ASCII` is the option.

APPEND Allows you to attach the host data set to the end of an existing workstation file.

Warning: If `APPEND` is not specified and a workstation file already exists with the same name, the host data set replaces the existing workstation file.

ASCII Causes the host program to translate workstation files from EBCDIC to ASCII so that the file is readable to the workstation. When receiving text data sets from the host, always specify the `ASCII` option. If the host data set is already in ASCII form, do not supply this option. Do not use this option when you are transferring a binary file.

RECEIVE for TSO Host



H(*nnn*)

Specifies the host file code page to be used for the host file being sent, where *nnn* is the code page number. Choices are 037, 273, 277, 278, 280, 284, 285, 297, or 500. If you do not specify a code page, the system will use the code page for the host character set of the terminal emulation session which is specified in the keyboard remap profile.

The host character set will also specify the file transfer command program to be used.

Note: This parameter is only valid when the ASCII keyword is used.

P(*nnn*)

Specifies the workstation file code page to be used for the workstation file being received, where *nnn* is the code page number. Choices are 437, 850, 860, 863, or 865. If you do not specify a code page, the system will use the code page for the active command line screen group.

MIGR

Allows you to send or receive files that were transferred using Communications Manager Version 1.0. This only applies to files that were transferred in one direction using the ASCII option. Use this option to transfer the file back to the source location.

This option is valid only if the ASCII option is also specified. If MIGR is specified but the ASCII option is not, MIGR is ignored.



RECEIVE for TSO Host

T(*nn*) Specifies the period of time that the file transfer program waits for the host to respond before the file transfer request is cancelled. *nn* is a two-digit integer in a range from 00 through 99. *nn* indicates the number of 20-second increments the file transfer program waits for the host to respond. If **T(*NN*)** is not specified, **T(00)** is the default, and the file transfer program waits indefinitely for the host to respond.

While waiting for the host to respond, the file transfer program sends a message to the screen every 30 seconds that the host has not responded.

crlf Causes the host program to insert carriage returns and line feed characters as the last 2 characters of each line before sending the data set to the workstation.

Examples

To receive a file from an IBM TSO host from an OS/2 command prompt:

1. Type the RECEIVE command with the necessary parameters at an OS/2 full-screen command prompt. For example:

```
receive a:myfile.txt stat2:boss(bossname)/gky5hf ASCII
```

2. Press the Enter key.

Remarks

- You can also type the RECEIVE command in a command file and run that file only from the OS/2 full-screen command prompt.
- You can transfer more than one file at a time as long as each file transfer is on a different emulation session and is initiated in a separate screen group. The maximum number of concurrent file transfers is nine.

RECEIVE for TSO Host



- To issue multiple file transfers, specify a different emulation session for each file transfer with the ID in the file transfer command. Note that the emulation session that corresponds to the ID that you specify must be started and ready to receive a command.
- If your host file transfer program name is not IND\$FILE, use the OS/2 SET command before issuing the file transfer command. To use the SET command, type:

```
SET IND_FILE=xxxxxxxx
```

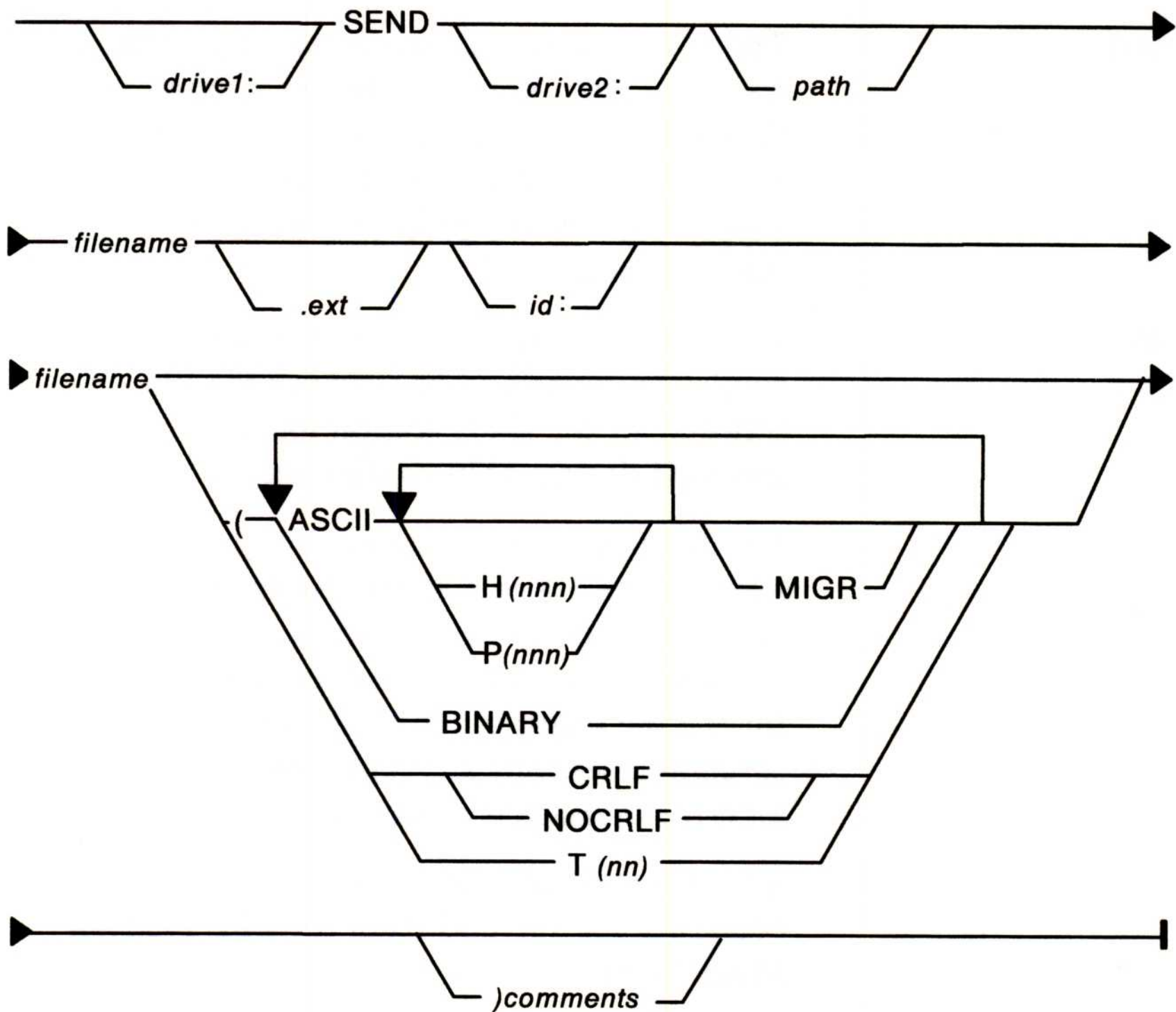
where `xxxxxxxx` is your host file transfer name. It is recommended that you use the OS/2 SET command in the CONFIG.SYS file to avoid having to do this each time.



Purpose

Sends a file to an IBM CICS host computer.

Syntax



Parameters

- drive1:** The drive where the SEND.EXE file resides. Specify this only if it is not in the default drive.
- drive2:** The drive where the workstation file that you want to send to the host resides. Specify this only if it is not in the default drive. In the following example, **a:** is the drive.

SEND for CICS Host



path	Path designator that specifies where the file is. Use this only if the file is not in the default path.
filename	The name of the workstation file that you want to send. You are allowed 8 characters for the file name. In the following example, <code>report</code> is the file name.
.ext	The extension for the workstation file that you want to send. You are allowed 4 characters for the extension, including the period. The file name will not have an extension if you do not supply one. In the preceding example, <code>.dat</code> is the extension.
id:	<p>The 1- to 8-character name that identifies the emulation session to be used for the file transfer, or the short session identifier you specified during 3270 configuration.</p> <p>For 3270 emulation sessions, <code>id</code> can be either a long session ID or a short session ID. If you do not supply this information, the first configured and started emulation session is used for the file transfer. If no 3270 terminal emulation sessions are configured and started, an error message is returned.</p> <p>You can view the identifiers in the configuration file and from the File Transfer Main Menu.</p>



SEND for CICS Host

In the preceding example, `stat2` is the ID. Note that the ID is separated from the file name by a colon (:).

filename

The CICS file name of the file that you want to transfer. The name can be up to 8 characters, and must begin with a letter. The file name may be a program name, a transaction identifier, or any other identifier selected by the CICS application programmer. This name is written into the first record of the host file to allow host users, such as another transaction, to identify the data sent from the workstation. In the preceding example, `report` is the file name.

Options

The following parts of the command are the options. The options are separated from the last parameter by a left parenthesis with one space between each option. In the example, `(ASCII` is the option.

ASCII

Causes the host program to translate workstation files from ASCII to EBCDIC so that the file is readable to the host. When sending text files to the host, specify the `ASCII` option. If the workstation file is already in EBCDIC form, do not supply this option. Do not use this option when you are transferring a binary file.

H(*nnn*)

Specifies the host file code page to be used for the host file being received, where *nnn* is the code page number. Choices are `037`, `273`, `277`, `278`, `280`, `284`, `285`, `297`, or `500`. If you do not specify a code page, the system uses the code page for the host character set of the emulation that is specified in the keyboard remap profile.

The host character set will also specify the file transfer command program to be used.

Note: This parameter is only valid when the `ASCII` keyword is used.



P(*nnn*) Specifies the workstation file code page to be used for the workstation file being sent, where *nnn* is the code page number. Choices are [437](#), [850](#), [860](#), [863](#), or [865](#). If you do not specify a code page, the system uses the code page for the active command line screen group.

MIGR Allows you to send or receive files that were transferred using Communications Manager Version 1.0. This only applies to files that were transferred in one direction using the ASCII option. Use this option to transfer the file back to the source location.

This option is valid only if the ASCII option is also specified. If MIGR is specified but the ASCII option is not, MIGR is ignored.

BINARY Causes binary files to be sent to the host without being changed. When sending binary files to the host, always specify the [BINARY](#) option.

CRLF Causes the host program to delete carriage return and line feed characters from the workstation file before storing it at the host. When sending an ASCII or EBCDIC file to the host, it is recommended that you always supply the [CRLF](#) option so the file will be readable when on the host.

Note: The file is written into CICS temporary storage using one item on the queue to represent each logical record. Null records in workstation files, indicated by 2 carriage return/line feed characters appearing directly after one another, will be represented in the file in CICS temporary storage by an item with two blanks in it.



SEND for CICS Host

If you specify the **ASCII** option, the blanks inserted into the file will be EBCDIC blanks.

The maximum length allowed for a logical record is 32767 bytes. If a logical record exceeds this size, you will receive a warning message at the end of the file transfer. Data should not be lost, but the data may be incorrectly formatted into the host file.

NOCRLF

Specifies that the workstation file being sent does not consist of logical records delimited by carriage return/line feed characters. The file transfer program will not split the records and the file is written into CICS temporary storage using one item on the queue to represent each inbound data buffer.

T(NN)

Specifies the period of time that the file transfer program waits for the host to respond before the file transfer request is cancelled. *nn* is a two-digit integer in a range from 00 through 99. *nn* indicates the number of 20-second increments the file transfer program waits for the host to respond. If **T(NN)** is not specified, **T(00)** is the default, and the file transfer program waits indefinitely for the host to respond.

While waiting for the host to respond, the file transfer program sends a message to the screen every 30 seconds that the host has not responded.

COMMENTS

The comments parameter allows you to supply comments or control information to be associated with the host file.

To insert comments, enter a right parenthesis, **)**, to indicate the end of the options, then follow the parenthesis with your comments. If you want to add comments but you do not want to specify any options, enter the left and right parenthesis immediately before the comments to indicate the absence of options.

SEND for CICS Host



Any comments will be written into the host file in the first record. It is then available for another transaction to use.

In the example, `)enddata` is the comment.

Examples

To send a file to an IBM CICS host from an OS/2 full-screen command prompt:

1. Type the SEND command with the necessary parameters at an OS/2 command prompt. For example:

```
send a:report.dat stat2:report (ASCII)enddata
```

2. Press the Enter key.

Remarks

- You can also type the SEND command in a command file and run that file only from the OS/2 full-screen command prompt.
- You can transfer more than one file at a time as long as each file transfer is on a different emulation session and is initiated in a separate screen group. The maximum number of concurrent file transfers is nine.

To issue multiple file transfers, specify a different emulation session for each file transfer with the ID in the file transfer command. Note that the emulation session that corresponds to the ID that you specify must be started and ready to receive a command.

- The options used are always written into the first record of the file in the host, regardless of whether they were entered at the workstation.
- The options `CRLF` and `NOCRLF` are mutually exclusive.
- The options `CRLF` and `ASCII` imply each other if one is specified and not the other.
- The options `BINARY` and `ASCII` are mutually exclusive.

CAUTION:

Unpredictable ASCII to EBCDIC translation will occur if neither `ASCII` nor `BINARY` is specified.



SEND for CICS Host

- The options **BINARY** and **NOCRLF** imply each other if one is specified and not the other.
- If no options are specified, **CRLF** and **ASCII** are assumed.
- If your host file transfer program name is not **IND\$FILE**, use the OS/2 **SET** command before issuing the file transfer command. To use the **SET** command, type:

```
SET IND_FILE=xxxxxxxx
```

where **xxxxxxxx** is your host file transfer name. It is recommended you use the OS/2 **SET** command in the **CONFIG.SYS** file to avoid having to do this each time.

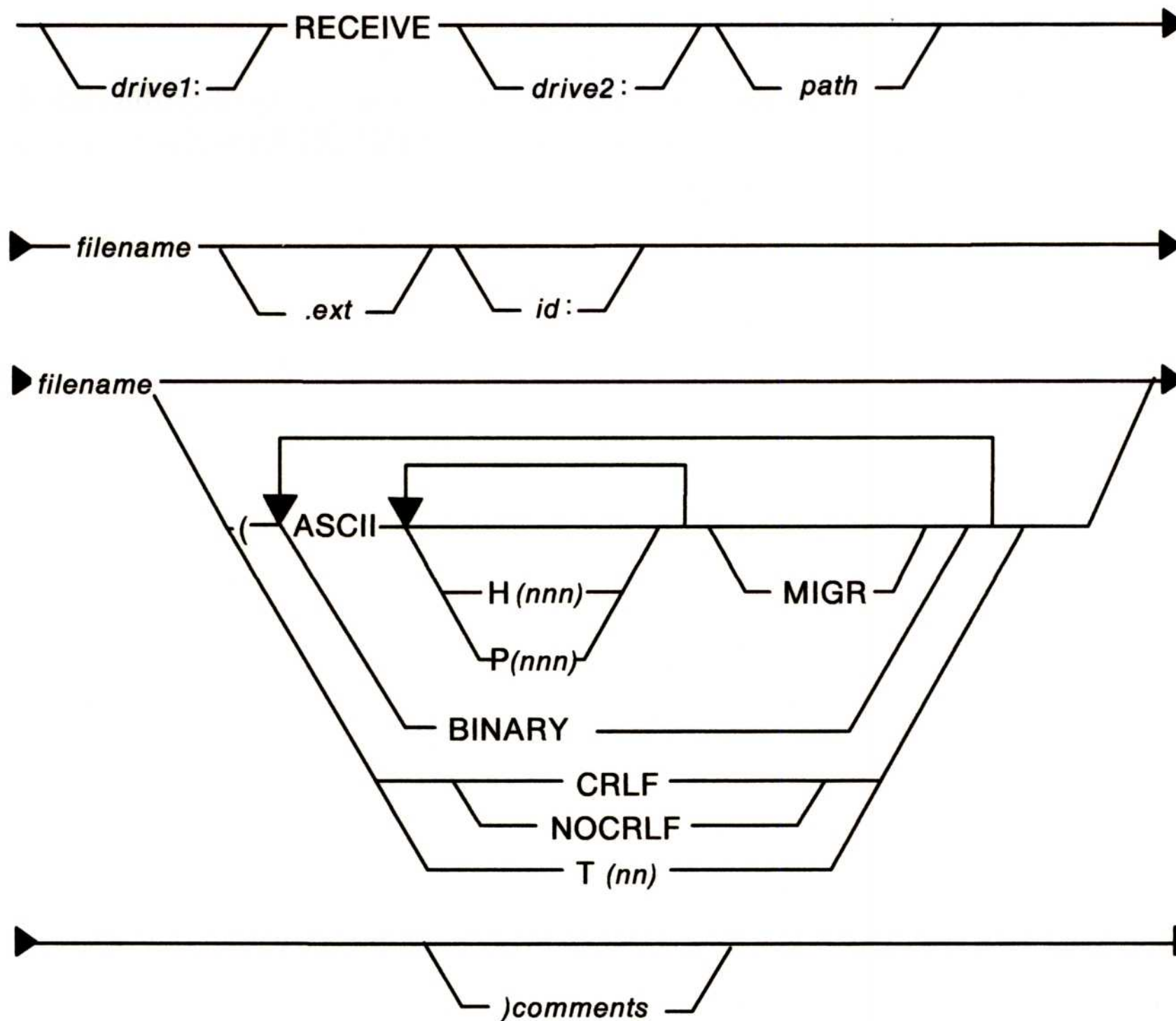
RECEIVE for CICS Host



Purpose

Receives a file from an IBM CICS host computer.

Syntax



Parameters

- drive1:** The drive where the RECEIVE.EXE file resides. Specify this only if it is not in the default drive.
- drive2:** The drive where you want the file that you are receiving from the host to reside. Specify this only if it is not in the default drive. In the following example, **a:** is the drive.



RECEIVE for CICS Host

path	Path designator that specifies where the file is to reside. Use this only if it will not be in the default path.
filename	The name of the workstation file being received from the host. You are allowed 8 characters for the file name. In the preceding example, <code>report</code> is the file name.
.ext	The extension for the workstation file being received from the host. You are allowed 4 characters for the extension, including the period. The file name will not have an extension if you do not supply one. In the following example, <code>.dat</code> is the extension.
id:	<p>The 1- to 8-character name that identifies the emulation session to be used for the file transfer, or the short session identifier you specified during 3270 configuration. For 3270 emulation sessions, <code>id</code> can be either a long session ID or a short session ID. If you do not supply this information, the first configured emulation session is used for the file transfer. If no 3270 terminal emulation sessions are configured, an error message is returned.</p> <p>You can view the identifiers in the configuration file and from the File Transfer Operations Menu.</p> <p>In the preceding example, <code>stat2</code> is the ID. Note that the ID is separated from the file name by a colon (:).</p>



filename The CICS file name of the file that you want to transfer. The file name may be a program name, a transaction identifier, or any other identifier selected by the CICS application programmer. This name is written into the host file to allow host users, such as another transaction, to identify the data sent from the workstation. In the preceding example, `report` is the file name.

Options

The following parts of the command are the options. The options are separated from the last parameter by a left parenthesis with one space between each option. In the example, `(CRLF ASCII` are the options.

ASCII Causes the host program to translate workstation files from EBCDIC to ASCII so that the file is readable to the workstation. When receiving text files from the host, specify the `ASCII` option. If the host file is already in ASCII form, do not supply this option. Do not use this option when you are transferring a binary file.

H(*nnn*) Specifies the host file code page to be used for the host file being received, where *nnn* is the code page number. Choices are `037`, `273`, `277`, `278`, `280`, `284`, `285`, `297`, or `500`. If you do not specify a code page, the system uses the code page for the host character set of the emulation that is specified in the keyboard remap profile.

The host character set will also specify the file transfer command program to be used.

Note: This parameter is only valid when the ASCII keyword is used.



RECEIVE for CICS Host

- P(*nnn*)** Specifies the workstation file code page to be used for the workstation file being sent, where *nnn* is the code page number. Choices are [437](#), [850](#), [860](#), [863](#), or [865](#). If you do not specify a code page, the system uses the code page for the active command line screen group.
- MIGR** Allows you to send or receive files that were transferred using Communications Manager Version 1.0. This only applies to files that were transferred in one direction using the ASCII option. Use this option to transfer the file back to the source location.
- This option is valid only if the ASCII option is also specified. If MIGR is specified but the ASCII option is not, MIGR is ignored.
- BINARY** Causes binary files to be sent to the workstation without being changed. When receiving binary files from the host, always specify the [BINARY](#) option.
- CRLF** Causes the host program to delete carriage return and line feed characters from the host file before sending it to the workstation.
- If you specified the [ASCII](#) option, EBCDIC blanks will be removed. If you specified the [BINARY](#) option, ASCII blanks will be removed.
- NOCRLF** Specifies that the host file being sent does not consist of logical records delimited by carriage return/line feed characters. The file transfer program will not split the records and the file is written into the workstation file as one long string of data.



T(NN) Specifies the period of time that the file transfer program waits for the host to respond before the file transfer request is cancelled. *nn* is a two-digit integer in a range from 00 through 99. *nn* indicates the number of 20-second increments the file transfer program waits for the host to respond. If **T(NN)** is not specified, **T(00)** is the default and the file transfer program waits indefinitely for the host to respond.

While waiting for the host to respond, the file transfer program sends a message to the screen every 30 seconds that the host has not responded.

Examples

To receive a file from an IBM CICS host from an OS/2 full-screen command prompt.

1. Type the **RECEIVE** command with the necessary parameters at an OS/2 command prompt. For example:

```
receive a:report.dat stat2:report (CRLF ASCII
```

2. Press the Enter key.

Remarks

- You can also type the **RECEIVE** command in a command file and run that file only from the OS/2 full-screen command prompt.
- You can transfer more than one file at a time as long as each file transfer is on a different emulation session and is initiated in a separate screen group. The maximum number of concurrent file transfers is nine.

To issue multiple file transfers, specify a different emulation session for each file transfer with the **ID** in the file transfer command. Note that the emulation session that corresponds to the **ID** that you specify must be started and ready to receive a command.



RECEIVE for CICS Host

- The options used are always those that you specify in the command, or those implied by the command. The options in the file at the host, if any, are never used.
- The options `CRLF` and `NOCRLF` are mutually exclusive.
- The options `CRLF` and `ASCII` imply each other if one is specified and not the other.
- The options `BINARY` and `ASCII` are mutually exclusive.

CAUTION:

Unpredictable ASCII to EBCDIC translation will occur if neither `ASCII` nor `BINARY` is specified.

- The options `BINARY` and `NOCRLF` imply each other if one is specified and not the other.
- If no options are specified, `CRLF` and `ASCII` are assumed.
- If your host file transfer program name is not `IND$FILE`, use the OS/2 `SET` command before issuing the file transfer command. To use the `SET` command, type:

```
SET IND_FILE=xxxxxxxx
```

where `xxxxxxxx` is your host file transfer name. It is recommended you use the OS/2 `SET` command in the `CONFIG.SYS` file to avoid having to do this each time.

Chapter 4. Query Manager Commands

This chapter provides an alphabetical list of the Query Manager commands. Each command description has a purpose, format or syntax, parameter list, example, and any notes or hints to help explain how to use the command. For a detailed description of the syntax diagrams, see Appendix A.

Notational Conventions

The following notational conventions are used in the example for each command:

- Single quotes (' ') around commands are only required if you are typing Query Manager commands within procedures. If you are typing the commands from the Command Line panel, no quotes are required.
- At least one blank space or a comma (,) is required as a separator between the words of a command. Multiple blanks are also permitted and blanks can precede the first word of a command. When words are next to special symbols (such as parentheses, equal sign, or comma), the blank separator is optional.



Purpose

To have Query Manager consider all commands or statements that follow the **BEGIN WORK** command as a single unit. No permanent update to the database is made until an **END WORK** command is encountered. These two commands allow you to run procedures that can control how database actions should be grouped into one unit.

Syntax

—— **BEGIN WORK** —————

Examples

The following example causes all changes in all of the panel sets updated by the panel **DEPT** to be committed at one time rather than each panel instance being committed.

```
'begin work'  
'run panel dept(mode=change'  
'end work'
```

Remarks

- The **BEGIN WORK** command can be entered at any time from the Command Line panel and from within panel, procedure, and menu definition objects.
- Do not issue an **IMPORT** or **EXPORT** command between a **BEGIN WORK** and an **END WORK** command.
- Do not enter this command while issuing either a **COMMIT** or **ROLLBACK** command.
- If a **BEGIN WORK** has been entered with no corresponding **END WORK**, an **END WORK** is implicitly issued when one of the following occurs:
 - You exit the Command Line panel.
 - The customized mode ends and you are returned to the OS/2 command prompt or the Start Programs window, depending on where the customized mode was started.



BEGIN WORK

- You return to the primary menu for the object or the definition panel for the object from which you started the **BEGIN WORK** command.

Note: In the case of a **BEGIN WORK** nested within a hierarchy of menus, panels, or procedures with no explicit **END WORK** specified, an implicit **END WORK** is issued when one of the following occurs:

- The top of the hierarchy is exited back to the menu or panel from which the **BEGIN WORK** was initially issued.
- You exit the Command Line panel.
- The customized mode ends and you are returned to the OS/2 command prompt or the Start Programs window, depending on where the customized mode was started.



Purpose

To cancel any updates to the database performed by Query Manager commands that have been processed since the last **BEGIN WORK** command was issued. From this point on, any commands that are issued are not considered part of a unit initiated by **BEGIN WORK**.

Syntax

—— CANCEL WORK —————

Examples

The **CANCEL WORK** command in the following example cancels work done by panel **DEPT** if there is an error that ends running the panel.

```
'begin work'  
'run panel dept(mode=add'  
if rc=0 then  
  'end work'  
else  
  'cancel work'  
end
```

Remarks

- The **CANCEL WORK** command can be entered at any time from the Command Line panel and from within panel, procedure, and menu definition objects.
- If **CANCEL WORK** is issued while a panel is being processed, and the panel is one in a series of nested panels, all levels of panel processing are cancelled.



Purpose

To commit all updates to the database performed by Query Manager commands that were processed since the **BEGIN WORK** command was issued. From this point on, any commands that are issued are not considered part of the unit initiated by a **BEGIN WORK**.

Syntax

— **END WORK** —————

Examples

The **END WORK** command in the following example commits the work done by panel **DEPT** if there is no error while running the panel.

```
'begin work'  
'run panel dept(mode=add'  
if rc=0 then  
    'end work'  
else  
    'cancel work'  
end
```

Remarks

- The **END WORK** command can be entered at any time from the Command Line panel and from within panel, procedure, and menu definition objects.
- Do not issue an **IMPORT** or **EXPORT** command before an **END WORK** command. In addition, do not enter the **END WORK** command while issuing a **COMMIT** or **ROLLBACK** command.
- If a **BEGIN WORK** is entered with no corresponding **END WORK**, an **END WORK** is implicitly issued when one of the following occurs:
 - You exit the Command Line panel.
 - The customized mode ends and you are returned to the OS/2 command prompt or the Start Programs window, depending on where the customized mode was started.

END WORK

- You return to the primary menu for the object or the definition panel for the object from which you started the **BEGIN WORK** command.

Note: In the case of a **BEGIN WORK** nested within a hierarchy of menus, panels, or procedures with no explicit **END WORK** specified, an implicit **END WORK** is issued when one of the following occurs:

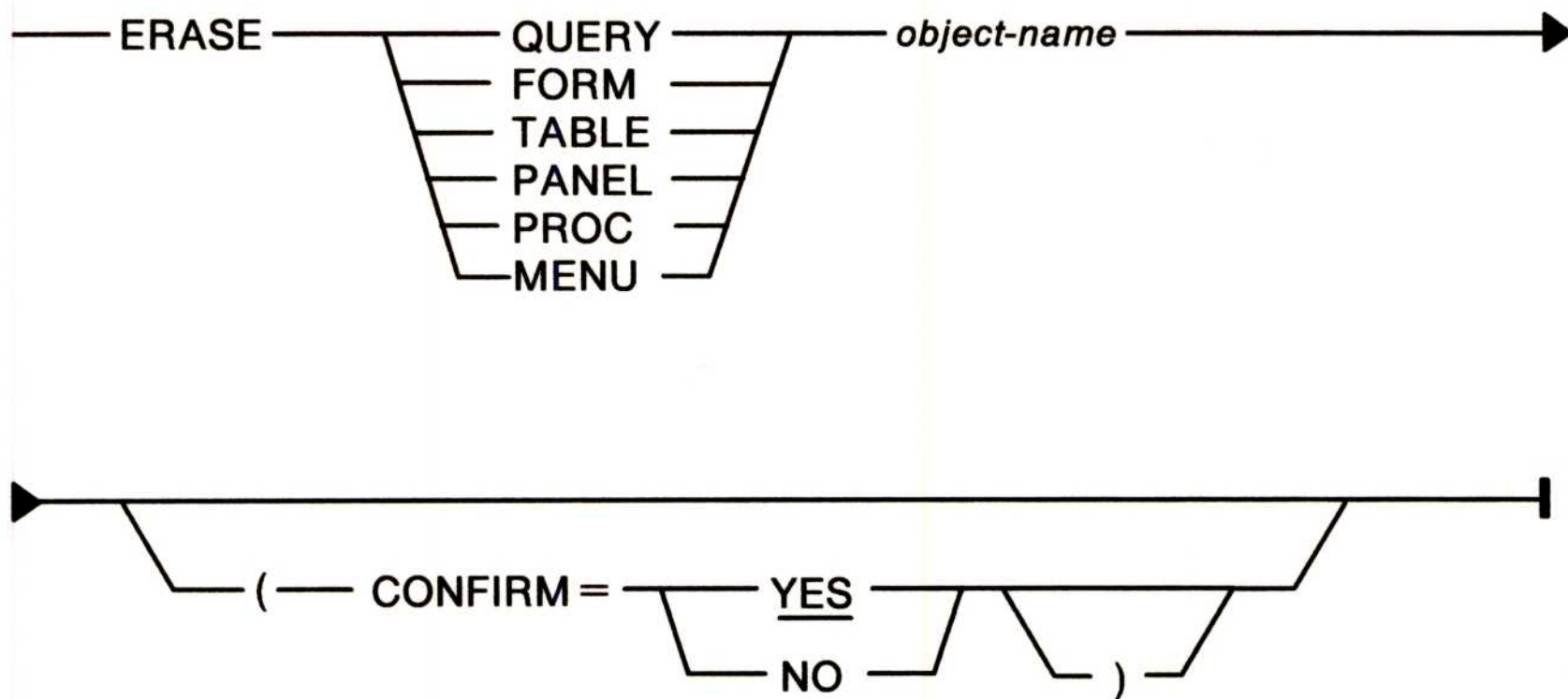
- The top of the hierarchy is exited back to the menu or panel from which the **BEGIN WORK** was initially issued.
- You exit the Command Line.
- The customized mode ends and you are returned to the OS/2 command prompt or the Start Programs window, depending on where the customized mode was started from.



Purpose

To delete an object from the database.

Syntax



Parameters

object-name

The name of the object to erase from the database. If **TABLE** is specified, the object can be either a table or a view.

CONFIRM

A Query Manager keyword. Specifying **YES** causes a prompt for confirmation to display before erasing the object. **YES** is the default.

Examples

The following erases the query named **TOTALSAL** without showing the user a confirmation message.

```
'erase query totalsal(confirm=no)'
```

ERASE



Remarks

- The ERASE command can be entered at any time from the Command Line panel and from within panel, procedure, and menu definition objects.
- Erasing a table erases any views and any indexes based on that table.

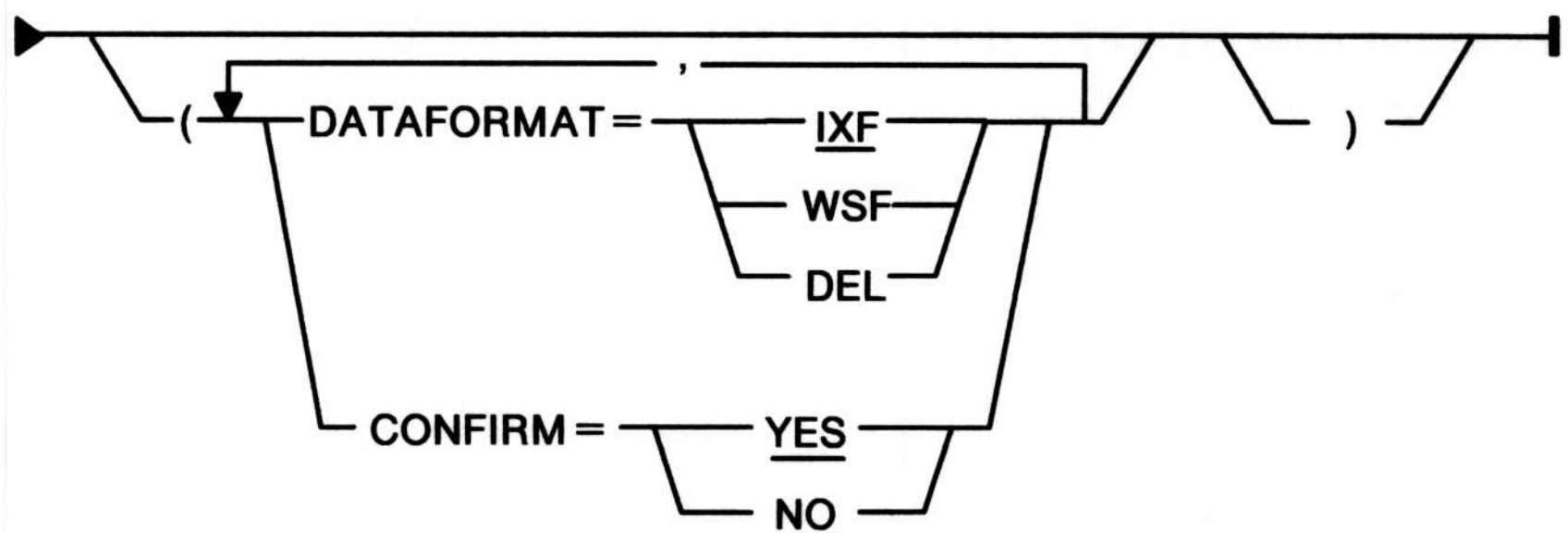


Purpose

To copy data from a table or view to an OS/2 file.

Syntax

▶ EXPORT TABLE — *object-name* — TO — *filename* —▶



Parameters

object-name

The name of the table or view from which to export data.

TO

A required keyword used to separate the object name from the file name.

filename

The path and name of the OS/2 file that will receive the exported data. If you do not specify an extension, an extension is appended based on the DATAFORMAT. For example, the extension .IXF, .DEL, and for WSF, either .WKS, .WRK, .WK1, or .WR1, would be appended depending on the format specified for WSF in the active profile. For more information on profiles, see “Maintaining Your Database” in the *User’s Guide*.

EXPORT



DATAFORMAT A keyword that identifies what type of data format is to be used in the OS/2 file. IXF is the default.

CONFIRM A Query Manager keyword. Specifying YES causes a prompt to confirm that you are about to replace an existing OS/2 file to display. YES is the default.

Examples

The following creates an OS/2 file named FILEA.DEL containing the exported table STAFF. If FILEA.DEL already exists, a confirmation message is not displayed.

```
'export table staff to filea(dataformat=del,confirm=no)'
```

Remarks

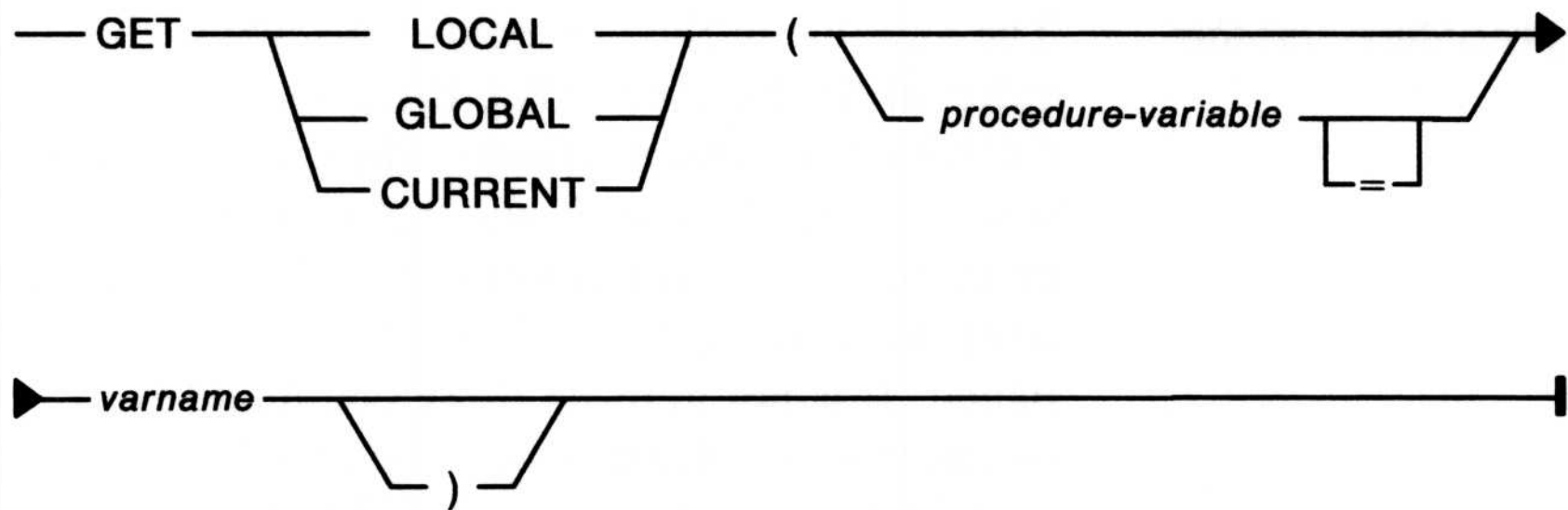
- The EXPORT command can be entered at any time from the Command Line panel and from within panel, procedure, and menu definition objects. However, the EXPORT command cannot be used in a nested procedure run within a panel.
- Messages received during export processing are stored in a file named QRWEXPRT.LOG. This file is stored in the SQLLIB directory. Review the contents of this file using a base operating system editor or using the OS/2 TYPE or PRINT commands. Record the message identification numbers. Return to Query Manager, press the Help (F1) key, press the Index (F5) key, and select **Import/Export Messages**. In the Import/Export Messages index, select the message identification number that matches the number you recorded and press the Enter key. The help panel for the message is displayed.
- For more information on data file formats, see “Database Manager Import and Export Considerations” in the *User's Guide*.
- Do not issue an EXPORT command between a BEGIN WORK and END WORK command.
- Although an ASCII file cannot be exported, it can be imported.



Purpose

To get the value of a Query Manager variable and either place it into a variable used in a procedure or display it. (If you want to display a variable from a procedure, use the SAY procedure language statement.)

Syntax



Parameters

LOCAL

A Query Manager variable that is at the level of the procedure being run. If the variable does not exist: 1) and the command was run from the Command Line panel or from within a menu or panel, an error message is displayed, or 2) the variable is created at the level of the procedure and the user is prompted for a value.

GLOBAL

A Query Manager variable that is global. If the variable does not exist: 1) and the command was run from the Command Line panel or from within a menu or panel, an error message is displayed, or 2) a global variable is created and the user is prompted for a value.



CURRENT

The latest Query Manager variable with the specified varname. This variable is searched for starting at the level of the procedure running and proceeding through all levels of variables including global variables. If the variable does not exist: 1) and the command was run from the Command Line panel or from within a menu or panel, an error message is displayed, or 2) the variable is created at the level of the procedure and the user is prompted for a value.

procedure-variable

The variable in the procedure into which the value of the LOCAL, GLOBAL, or CURRENT variable (Query Manager variable) is to be placed. This is allowed only in a procedure; you cannot issue the GET command with this variable from the Command Line panel. It is not allowed if the command is issued from the Command Line panel, a user-defined menu, or a user-defined panel.

varname

The name of the LOCAL, GLOBAL, or CURRENT variable (Query Manager variable) from which the value is to be retrieved.

Examples

The following procedure variables, JOB and NAME, are set to the values in the Query Manager variables J and N, respectively. This is done so the values in the variables J and N are available to the procedure.

```
'get current (job=j)'  
say job  
'get current (name=n'
```

The following example causes the value in the variable J to be displayed, since J is not a procedure variable.

```
'get current (j)'
```




GET variable

Remarks

- Two adjacent (serial) GET commands must be separated by a SAY procedure language statement so that both variable values can be viewed by the user.
- The GET command can be entered at any time from the Command Line panel or from a procedure.
- If a procedure variable is not included in the GET command (for example, GET CURRENT (J)), the value of the variable is also displayed.

IMPORT



Purpose

To bring data from an OS/2 file into a table or view in the database.

Syntax

IMPORT TABLE *object-name* FROM *filename*

(DATAFORMAT = IXF
WSF
DEL)

If the imported file has the ASC data format, the IMPORT command must specify the *filespec* parameter, as shown in the following example.

IMPORT TABLE *object-name* FROM *filename*

(DATAFORMAT = ASC ,

FILESPEC = *tablename*)

Parameters

object-name

The name of the table or view into which the data is to be imported. The table or view must already exist in the database unless the DATAFORMAT is IXF. Then, a new table is created if the table or view does not exist.



FROM A required keyword used to separate the object name from the file name.

filename The path and name of the OS/2 file containing the data to import. If you do not specify an extension, an extension is appended based on the DATAFORMAT; for example, the extensions .IXF, .DEL, .ASC, and for WSF, either .WKS, .WRK, .WK1, or .WR1 are appended depending on the format specified for WSF in the active profile. For more information on profiles, see “Maintaining Your Database” in the *User's Guide*.

DATAFORMAT A keyword that identifies the type of data format of the data in the OS/2 file. IXF is the default.

filespec A keyword that allows you to specify a table that defines the positions of the columns in an OS/2 file. This keyword can be used only if DATAFORMAT = ASC and also is required in that case.

Examples

In the following, table NEWTAB is replaced by the imported data in the OS/2 file FILEB.ASC; this file must be in the format expected for ASC files.

```
'import table newtab from fileb (dataformat=asc, filespec=tdept)'
```

Remarks

- The IMPORT command can be entered at any time from the Command Line panel and from within panel, procedure, and menu definition objects.
- The data in the specified table or view is replaced.

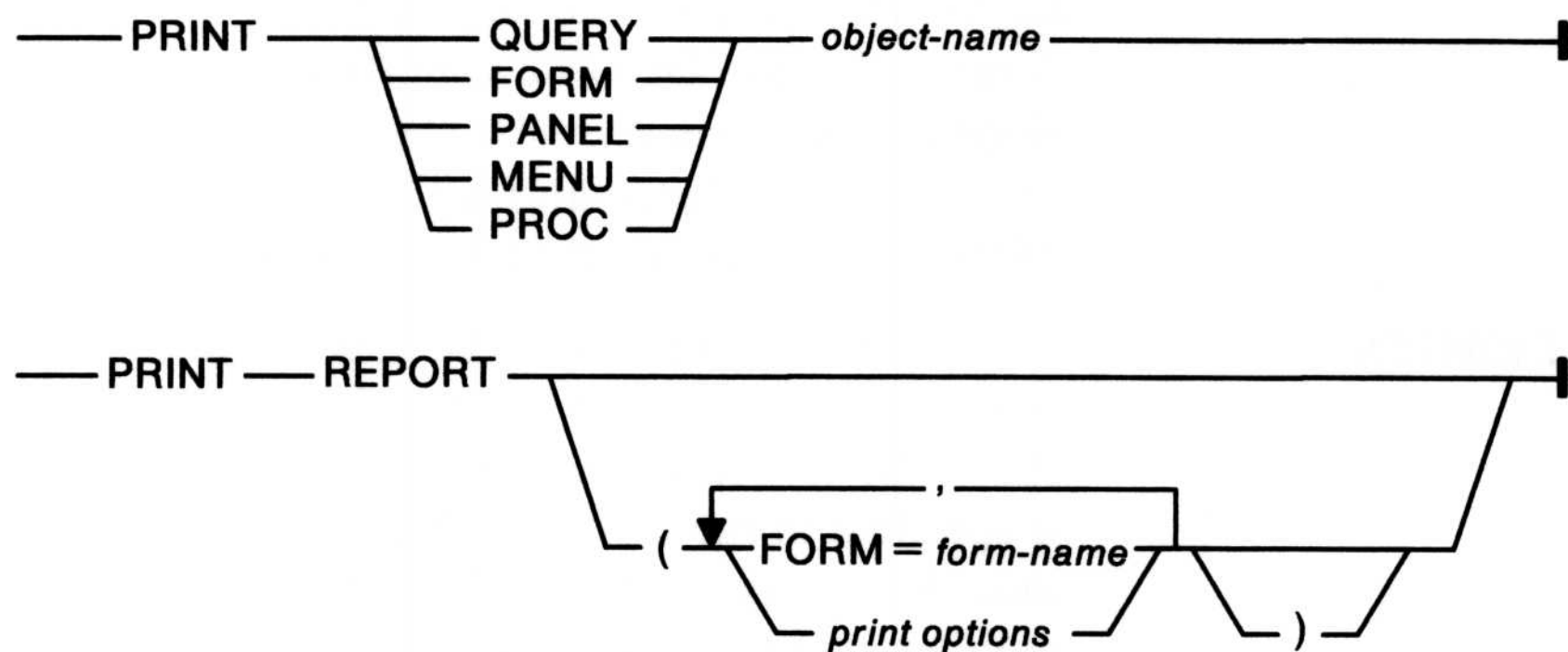
IMPORT



- Messages received during import processing are stored in a file named QRWIMPRT.LOG. This file is stored in the SQLLIB directory. Review the contents of this file using a base operating system editor or using the OS/2 TYPE or PRINT commands. Record the message identification numbers. Return to Query Manager, press the Help (F1) key, press the Index (F5) key, and select **Import/Export Messages**. In the Import/Export Messages index, select the message identification number that matches the number you recorded and press the Enter key. The help panel for the message is displayed.
- For more information on data file formats, see “Database Manager Import and Export Considerations” in the *User's Guide*.
- Do not issue an IMPORT command between a BEGIN WORK and END WORK command.

**Purpose**

To print objects that exist in the database or to print a report (PRINT REPORT) that shows the results of a RUN QUERY command. The destination or appearance of the printed output can be affected by settings in the profile or by using options from the print options in the PRINT command.

Syntax**Parameters**

object-name

The name of an object in the database.

FORM

A keyword, only used for REPORT, that indicates a report form is used to format the report.

form-name

The name of the form used to format a report that is the result of running a query.

print options

Print options used to print a report. More than one print option can be specified, separated by commas. The print options PRINTER, DATETIME, LENGTH, WIDTH, PAGENO, and PRINTTYPE have default values taken from the profile. For more information on profiles, see "Maintaining Your Database" in the *User's Guide*. The other options in the list can be FILE and CONFIRM.

PRINT



A description of each option follows:

PRINTER

A keyword that identifies the nickname of the printer on which the report is to be printed. This option cannot be used if the FILE option is also used. If the FILE option is not specified, the default is the PRINTER option. The format is `PRINTER=nickname`.

DATETIME

A keyword that causes the system date and time to print starting at the first character position on the last line of each page as defined in the LENGTH option. The date and time are formatted using the OS/2 country code for the session. The format is `DATETIME=YES` or `DATETIME=NO`. The default is YES, unless a different value is specified in active profile.

LENGTH

A keyword that identifies the maximum number of lines to print on any page. The range is 1 to 999. The default is 66, unless a different value is specified in the active profile. LENGTH must be at least 1 greater than: the total number of lines needed for column headings and page headings and footings; and, if either is specified, the PAGENO or DATETIME line that appears on each page of the report. The format is `LENGTH=x`, where `x` is 1 to 999.

WIDTH

A keyword that identifies the maximum number of characters to print on any line. The range is 22 to 999. The default is 80, unless a different value is specified in the active profile. The format is `WIDTH=x`, where `x` is 22 to 999.

**PAGENO**

A keyword that causes the page number to print at the right-most character position on the last line of each page as defined by the WIDTH and LENGTH options. The page number is four digits long and starts at 1 and goes to 9999. If the page number is higher than 9999, the pages are numbered 01 to 09999. The format is **PAGENO=YES** or **PAGENO=NO**. The default is YES, unless a different value is specified in the active profile. If this option is set to YES or the DATETIME option is set to YES, the last line of each page, as defined by the LENGTH option, is used. If the PAGENO option and the DATETIME option are set to NO, then all lines of the report, as defined by the form, are available.

PRINTTYPE

A keyword that controls the size of the print type. NORMAL is 10 pitch; COMPRESSED is either 15 or 17 pitch, depending on the printer you are using. The format is **PRINTTYPE=NORMAL** or **PRINTTYPE=COMPRESSED**. The default is NORMAL, unless a different value is specified in the active profile.

FILE

A keyword that specifies that the printed report should be stored in an OS/2 file instead of being printed on the printer. This option cannot be used if the PRINTER option is also used. The OS/2 file contains only normal printable characters, line ends, and page ejects. The format is **FILE=filename**.

CONFIRM

A Query Manager keyword that is used with FILE. Specifying **YES** causes a prompt for confirmation to display if an OS/2 file with the same name as the one specified for FILE = already exists; specifying **NO** in this case replaces the OS/2 file if one already exists. **YES** is the default. The format is **CONFIRM=YES** or **CONFIRM=NO**.

PRINT



Examples

The following PRINT REPORT command prints a report of the data accessed by query Q1 in the format defined by the form SALFORM.

```
'run query q1'  
'print report (form=salform)'
```

The following example prints a report of the data in default report form that was previously accessed by a query on printer LPT2 in compressed print, 60 lines per page with page numbers.

```
'print report (printer=lpt2, printtype=compressed,  
length=60, pageno=yes)'
```

The following prints the definition of panel ASTAFF.

```
'print panel astaff'
```

The following example creates an OS/2 file named REPORTA.PRT containing a default report of the data previously accessed by a query.

```
'print report (file=reportA.prt, confirm=no)'
```

Remarks

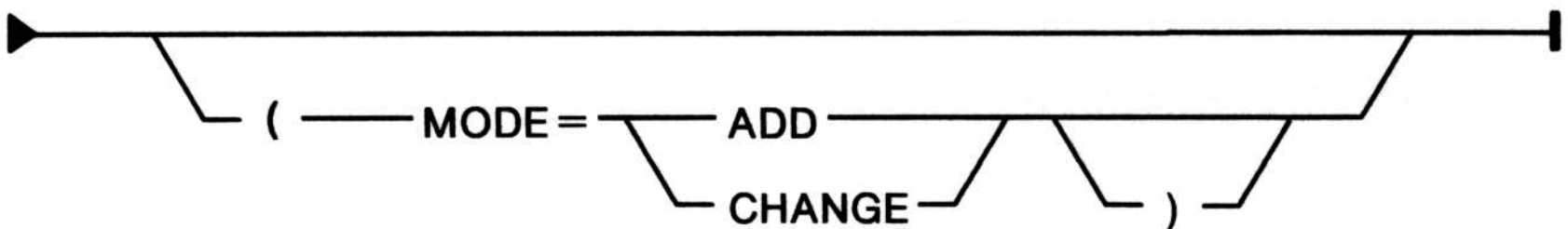
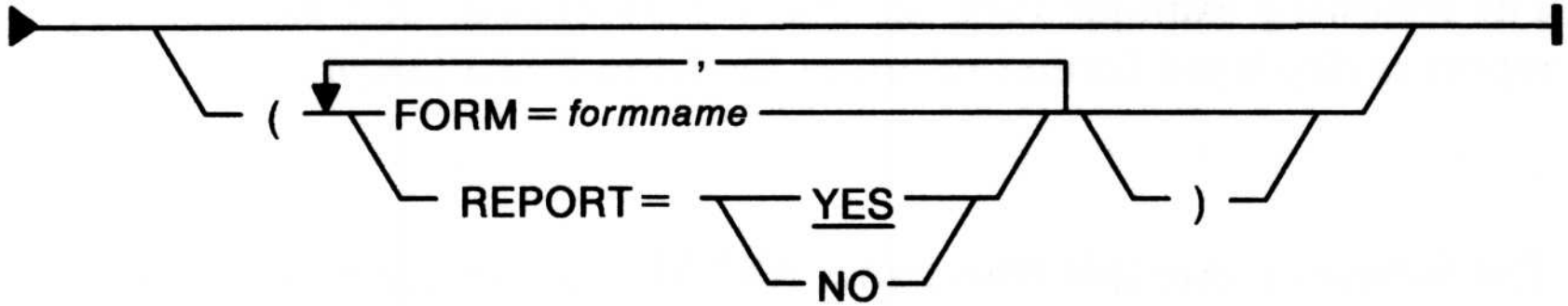
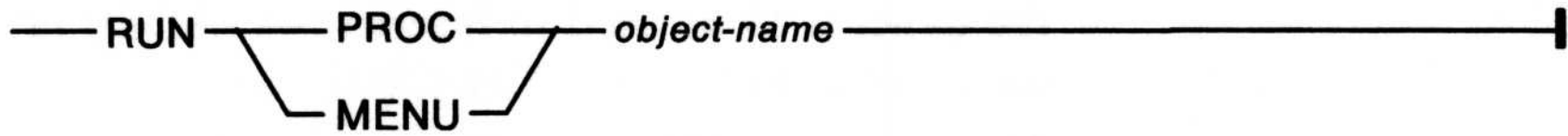
- The PRINT command can be entered at any time from the Command Line panel and from within panel, procedure, and menu definition objects.
- PRINT REPORT is only valid if the data has previously been accessed by a RUN QUERY.
- If an IMPORT or EXPORT command is performed after a RUN QUERY command, no data exists for the PRINT REPORT command to act on.



Purpose

To process objects that are located in the database.

Syntax



Parameters

object-name

The name of an object in the database.

FORM

A keyword, only used for `QUERY`, that indicates a report form is to be used to format the report.

form-name

The name of the form used to format a report that is the result of running a query.



REPORT

A keyword that allows you to determine whether to display the report. **REPORT** set to **YES** is the default, and causes the report to display in a browse-only Report panel. **REPORT** set to **NO** does not display the report.

MODE

A keyword that allows you to identify whether the panel is to be used to insert rows (**ADD**) or view and modify rows (**CHANGE**) in the database. If you do not specify **MODE**, a prompt is displayed so you can specify add or change.

Examples

The following example runs the query **TOTALSAL** and the resulting report is displayed formatted using the form **SALFORM**.

```
'run query totalsal (form=salform)'
```

The following example runs panel **ASTAFF** to change rows in the table.

```
'run panel astaff (mode=change)'
```

The following example runs the menu **PAPER** to be displayed.

```
'run menu paper'
```

The following runs the procedure named **STAFFCHG**.

```
'run proc staffchg'
```

Remarks

- The **RUN** command can be entered at any time from the Command Line panel and from within panel, procedure, and menu definition objects.
- If you use **RUN QUERY REPORT=YES**, the Report panel is displayed using the the default form or using the specified **FORM** if **FORM=formname**. Form name is the name of the form you want to use. The Report panel displayed is a browse-only panel; you cannot access the Query or Form panels from the Report panel in this case.



- If you use **RUN MENU** or **RUN PANEL**, the menu or panel is displayed. After using the menu or panel, you are returned to the environment from which the **RUN MENU** or **RUN PANEL** was issued.
- If you use the **RUN QUERY REPORT = YES** and the query is not a **SELECT** statement, you will receive an error message.
- Previously, **INTERACT** was supported instead of **REPORT**. **INTERACT** is still supported for migration purposes.

SAVE DATA AS

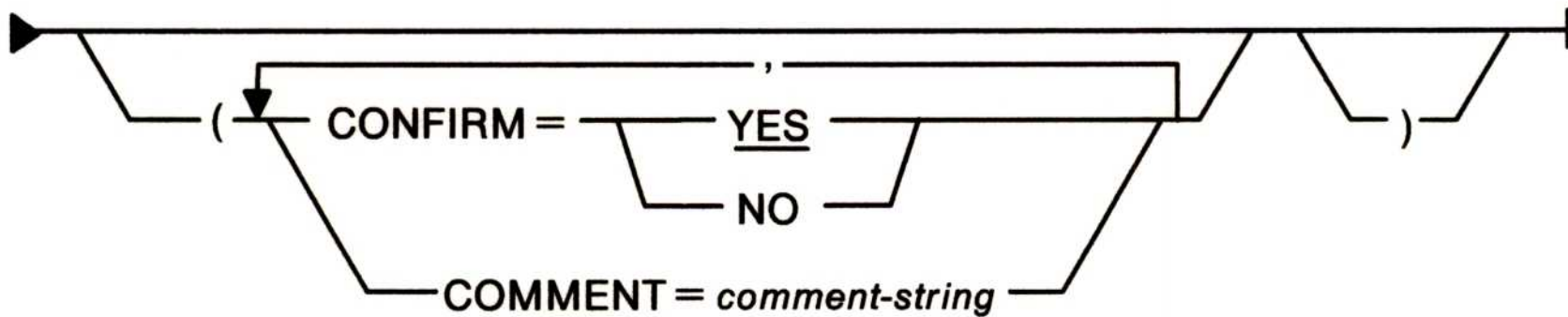


Purpose

To save the results (data) of a query. The query is either a prompted query or an SQL query that contains an SQL SELECT statement. The query must have been run using the RUN QUERY command.

Syntax

— SAVE DATA AS — *tablename* —————>



Parameters

table-name

The name of the table in which to save the data.

CONFIRM

A Query Manager keyword. Specifying **YES** causes a prompt for confirmation to display if the table already exists. **YES** is the default.

COMMENT

A keyword that allows you to type a comment to be attached to the table. The comment must be enclosed in single quotes.

comment-string

The comment-string can be from 1 to 55 characters in length and should be enclosed in single quotes.



SAVE DATA AS

Examples

The following saves the data accessed by query Q1 in a table named TOTSAL. A confirmation message will not be issued if TOTSAL already exists. The comment will be placed with the table name in the Tables and Views menu.

```
'run query q1'  
"save data as totalsal (confirm=no, comment='Data from query Q1')"
```

Remarks

- The SAVE command can be entered at any time from the Command Line panel and from within panel, procedure, and menu definition objects. DATA from a query must exist.
- If a new table is created, the original column names from the table on which the query is based are used as the new column names.
- All columns accessed are saved in the same order as originally accessed.
- If any column contains special data, the data cannot be saved as a table.
- If an IMPORT or EXPORT command is performed after a RUN QUERY command, no data exists for the SAVE DATA AS command to act on.

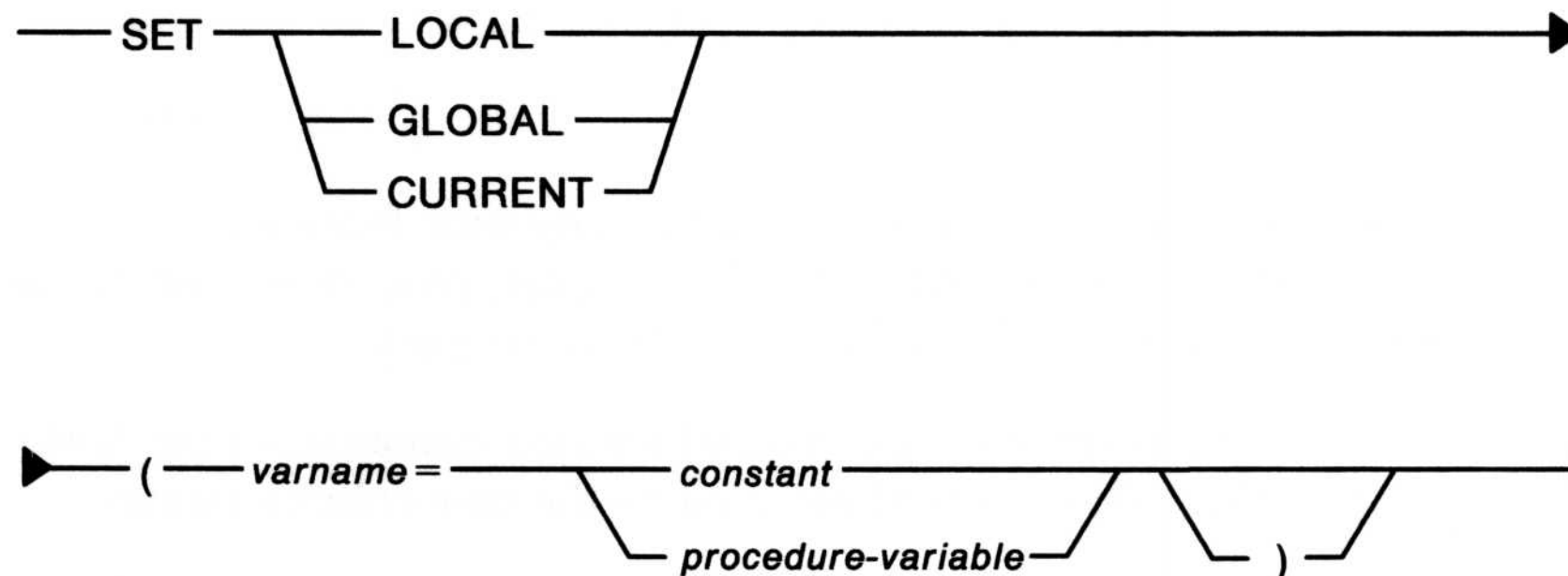
SET variable



Purpose

To set a Query Manager variable to a value.

Syntax



Parameters

LOCAL

The command updates an existing Query Manager variable at the level of the object being run. If the Query Manager variable does not exist, one is created.

GLOBAL

The command updates an existing Query Manager global variable. If the Query Manager variable does not exist, one is created.

CURRENT

The command updates the specified varname. This Query Manager variable is searched for starting at the level of the object running and proceeding through all levels of variables except global variables. If the Query Manager variable does not exist, one is created at the level of the object that issued the command.

varname

The name of the LOCAL, GLOBAL, or CURRENT variable (Query Manager variable) you want to set to a value. Do not place an & as the first character of the varname.



SET variable

constant

The value to be assigned to the Query Manager variable. A character constant should be enclosed in double quotes when it is used within a procedure, and single or double quotes when used within the Command Line panel. A numeric constant should not be enclosed in quotes.

procedure-variable

The variable in the procedure from which the value of the LOCAL, GLOBAL, or CURRENT variable (Query Manager variable) is taken. It is not allowed if the command is issued from the Command Line panel.

Examples

The following example sets the global variable VAR1 to the numeric value 12345.67.

```
'set global (var1=12345.67)'
```

The following example sets the most recent variable named VARN to the character constant Herzik. This is used to allow a value from a procedure to be available to other Query Manager objects.

```
'set current (varn="Herzik")'
```

Remarks

- The SET command can be entered at any time from the Command Line panel or from within panels, procedures, or menus.
- The value of the variable can be used in processing another function.



Purpose

The SQLQMFM command downloads and converts data that was originally stored in a DB2 table in an IBM MVS operating system into a file that can then be imported into a Database Manager table.

To use this command, you must first export the data from DB2 using QMF on the host. The data is exported by QMF, using the QMF data format, and is stored as a MVS data set on the host.

Once this is done, you start Communications Manager, log onto the appropriate host, and then issue the SQLQMFM command from the OS/2 command prompt. The MVS data set is downloaded in binary format and converted to an OS/2 delimited ASCII file. To import the data into a Database Manager table, you run the Import utility. To import the data, see To Import or Export Tables and Views in the “Using Database Manager” section of the *User’s Guide* for more information.

Certain prerequisites and considerations should be followed to use this facility. See the appendix for “Importing Data from DB2 and SQL/DS Databases” in the *User’s Guide* for more information.

Syntax

SQLQMFM *filename* *host session ID*

Parameters

filename

The file name specified should be the file name typed when exporting the data from QMF. This is the file name part of the MVS data set name (userid.filename.data). Remember, the QMF file name must adhere to the file name restrictions of both the base operating system and the host system, and must not be longer than eight characters.



host session id

The name of the host communications session where the QMF data is stored. This defaults to A but another session ID can be specified in the case where you are logged onto more than one communications session.

Examples

```
SQLQMFM ORGTABLE  
SQLQMFM ORGTABLE B
```

In the first example, the host session ID is assumed to be A. If you have more than one communications session running, you must specify the host session ID where QMF is running, if it is other than session ID A.

Remarks

As the data is downloaded, the SQLQMFM facility stores the data in the C:\SQLLIB\QMF directory. The facility automatically creates a column definition file and an Import file. The column definition file includes the necessary information for you to create a Database Manager table. The name of the column definition file is the same as the host file name with an extension of .COL. The name of the Import file is the same as the host file name with an extension of .DEL; for example:

```
ORGTABLE.COL  
ORGTABLE.DEL
```



Purpose

The SQLQMFV command downloads and converts data that was originally stored in an SQL/DS table in an IBM VM operating system into a file that can then be imported into a Database Manager table.

To use this command, you must first export the data from SQL/DS using QMF on the host. The data is exported by QMF, using the QMF data format, and is stored as a CMS file on the host.

Once this is done, you start Communications Manager, log onto the appropriate host session, and then issue the SQLQMFV command from the OS/2 command prompt. The CMS file is downloaded in binary format and converted to an OS/2 delimited ASCII file. To import the data into a Database Manager table, you run the Import utility. To import the data into a Database Manager table, you run the Import utility. To import the data, see To Import or Export Tables and Views in the “Using Database Manager” section of the *User's Guide* for more information.

Certain prerequisites and considerations should be followed to use this facility. See the appendix for “Importing Data From DB2 and SQL/DS Databases” in the *User's Guide* for more information.

Syntax

SQLQMFV *filename* *host session ID*

Parameters

filename The file name specified should be the file name typed when exporting the data from QMF. Remember, the QMF file name must adhere to the file name restrictions of both the base operating system and the host system, and must not be longer than eight characters. The file type and file mode are assumed to be DATA A, when exported from QMF.



host

The name of the host communications session where the QMF data session is stored. This defaults to A, but another session ID can be specified in the case where you are logging onto more than one communications session.

Examples

```
SQLQMFV ORGTABLE  
SQLQMFV ORGTABLE B
```

In the first example, the host session ID is assumed to be A. If you have more than one communications session running, you must specify the host session ID where QMF is running, if it is other than session ID A.

Remarks

As the data is downloaded, the SQLQMF facility stores the data in the C:\SQLLIB\QMF directory. The facility automatically creates a column definition file and an Import file. The column definition file includes the necessary information for you to create a Database Manager table. The name of the column definition file is the same as the host file name with an extension of .COL. The name of the Import file is the same as the host file name with an extension of .DEL; for example:

```
ORGTABLE.COL  
ORGTABLE.DEL SQLQMFV COMMAND
```

Chapter 5. Query Manager Procedure Language Statements

This chapter contains an alphabetical list of the Query Manager procedure language statements. Each statement has a purpose, format or syntax, example, parameter list, and any remarks to help explain how to use the statement.

For a detailed description of the syntax diagrams, see Appendix A.

Procedure Language Definitions

The following is a definition of the parameters allowed within procedure language statements:

Variables

A variable name can consist of a string from 1 to 18 characters. Variable names must begin with an alpha character, a dollar sign (\$), a pound sign (#), or an at sign (@). The rest of the variable name can be alphabetic or numeric (0 through 9), including the symbols dollar sign (\$), pound sign (#), at sign (@), or an underscore (_). Variable names can be in mixed case. For more information on the naming rules for variables, see “Database Manager Naming Rules” in the *User’s Guide*.

The value assigned to a variable can be a character string or a number. The value of a character string is case-sensitive; for example, **YES** is not equal to **yes**.

Procedure variables

Variables can be used within procedures to control the flow of the procedure or to allow for the substitution of variable values into instructions. You do not need to declare procedure variables; they are created automatically when needed and do not persist after the procedure ends. For more information on using variables within procedures, see “Using Query Manager Variables in Customization Tasks” in the *User's Guide*.

Query Manager variables

Query Manager variables exist outside of a procedure. For more information on Query Manager variables, see “Using Query Manager Variables in Customization Tasks” in the *User's Guide*. Also see the GET and SET Query Manager commands in Chapter 4 on page 4-1 for additional information on accessing Query Manager variables from a procedure.

Expressions

An expression can be a single variable or constant, or it can be a formula for combining multiple variables or constants. Expressions can contain arithmetic, logical, string, or comparison operators. Arithmetic operators are add (+), subtract (-), multiply (*), and divide (/). Logical operators are And (&), Or (|), Exclusive Or (&&), and Not (\). String expressions can be concatenated by using the concatenation string operator (||). Comparison operators are equal to (=), less than (<), greater than (>), less than or equal to (<=), greater than or equal to (>=), and not equal to (<> or ><).

An expression can evaluate to a number, a character string, or a logical value of 0 (false) or 1 (true).

Note: The | symbol, which is used for OR, and the || symbol, which is used for concatenation, is the character produced by pressing the key engraved with the | symbol. Depending on options in your CONFIG.SYS file, the symbol used may actually be displayed as a solid vertical line (| and ||).

Predicates

A predicate is a test comparing two expressions to determine a true or false condition. The comparison operators are shown under Expressions. Note that if character strings are being compared, their case must match. For example, YES is not equal to yes.

Constants

Constants are character strings or numbers that do not change.

Character strings

Character strings must be enclosed in either single or double quotes; the quotes are not considered part of the character string. A character string can be from 0 to 64 characters long, with the exceptions of fields in panels, which can contain character strings longer than 64 characters. There are three methods for specifying character strings:

- If double quotes enclose a character string then single quotes within the string are treated as part of the string, as follows:

"John O'Hara" would be displayed as John O'Hara

- If single quotes enclose a character string then double quotes within the string are treated as part of the string, as follows:

'"Department 10"' would be displayed as "Department 10"

- If there are two consecutive occurrences of the enclosing quote within the string, it is treated as a single occurrence of the quote and is considered to be part of the string, as follows:

'John O' 'Hara' would be displayed as John O'Hara

""Department 10"" would be displayed as "Department 10"

Character strings can also be constructed by concatenating a constant and a variable, as follows:

SAY "The value of A is now" || A

Numbers

Numbers must not be enclosed in quotes, and can be decimal numbers with no thousands separators. The decimal separator is assumed to be a period (.), the leading positive indicator an addition symbol (+), and the leading negative indicator a subtraction symbol (-). There is no trailing negative indicator. For numbers specified in scientific notation, both E and e are allowed. Examples of valid numbers are:

15
+6.9
-12345.6789
.0123
-1.234E+03

eol

An end-of-line or semicolon.

Instructions

An instruction is a unit within a procedure that can be executed. An instruction can be a Query Manager command or a procedure language statement or one or more instructions within a DO END pair.

Instruction list

An instruction list is a sequence of one or more instructions, each of which is ended with a semi-colon or by the end of the line on the screen.

Comments

Comments can be added to any line of a procedure. A comment begins with a /* and ends with */. All characters between the beginning of the comment and the end of the comment are considered to be part of the comment. Comments can be continued across multiple lines. Every procedure must begin with a comment.

Query Manager commands

Within a procedure, any character string that is not part of a procedure language statement is considered to be a Query Manager command. The character string can be constructed from constants or through variable substitution. For more information on Query Manager commands, see Chapter 4.



Purpose

To assign a value to a procedure variable.

Syntax

`variable = expression`

Parameters

Variable The name of a procedure variable.

Expression The expression can be any expression that evaluates to a number, a character string, or a logical value of 0 (false) or 1 (true).

Examples

The following example sets MONTH to a numeric value of 9.

```
month = 9
```

The following example sets QTR to the numeric value of the variable MONTH divided by 3.

```
qtr = month/3
```

The following example sets variables DAY and ERRMSG to the character strings.

```
day = 'Friday'  
errmsg = 'File not found'
```

**Purpose**

To delimit an instruction list, when the instruction list is to be treated as a single instruction, as in a THEN clause of an IF statement.

Syntax

— DO — *eol* — *instruction-list* — END —————

Parameters

eol An end-of-line or a semicolon.

Instruction-list A sequence of one or more instructions. The instruction list can either be a single statement or a series of statements, each ended by a semicolon (;) or by the end of the line.

Examples

The following causes the instructions between the DO/END pair to be run if MONTH is equal to yes.

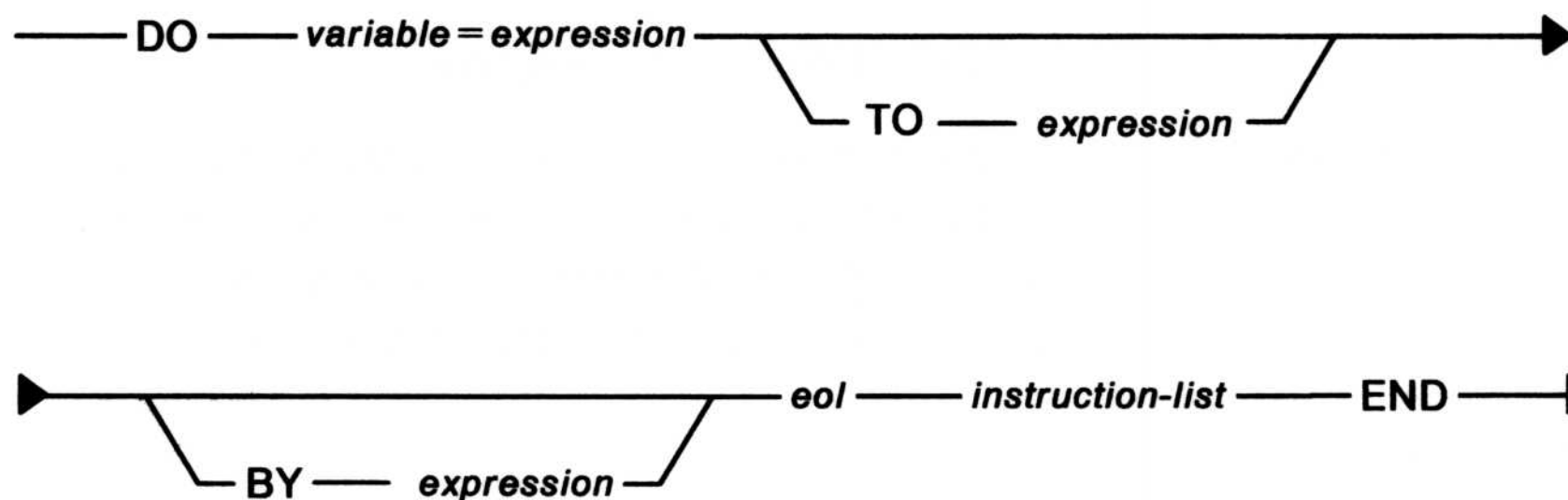
```
if month = 'yes'  
  then  
    do  
      'run query salesm'; 'print report (form=sales)'  
    end
```



Purpose

To repeat an instruction list one or more times, incrementing a variable by the BY expression (or by 1 if the BY expression is omitted). The repeated instruction list, or loop, will continue until the variable exceeds the TO expression. If the TO expression is omitted, the loop is an infinite loop.

Syntax



Parameters

Variable

The name of a procedure variable.

Expression

The expression can be any expression that evaluates to a number.

eol

An end-of-line or a semicolon.

Instruction-list

A sequence of one or more instructions. The instruction list can either be a single statement or a series of statements, each ended by a semicolon (;) or by the end of the line.



Examples

The following example causes the procedure SUMMARY_MONTH to be run 12 times.

```
DO MO = 1 TO 12; 'RUN PROC SUMMARY_MONTH'; END
```

The following example causes the query to be run four times for values of Q equal to 3, 6, 9 and 12. The Query Manager variable QUARTER will equal 1, 2, 3 and 4 for the four times the query is run; presumably QUARTER is the variable used in query QTR_SUMMARY to determine the data from which quarter to access.

```
Do q = 3 to 12 by 3  
  qtr = q / 3  
  'set global (quarter = qtr'  
  'Run query QTR_SUMMARY'  
  'Print report (form=SUMMARY)'  
End
```

Remarks

- The loop is processed at least one time, since the test to exit the loop is performed at the end of the loop.
- If the TO expression is omitted, the loop is an infinite loop.
- The loop can be exited using the LEAVE statement.



Purpose

To repeat an instruction list one or more times based on a true/false test. The instruction list is repeated until the test is false.

Syntax

```
—— DO UNTIL —— expression —— eol —— instruction-list —— END ——
```

Parameters

Expression The expression can be a predicate or any expression that evaluates to a logical value of 0 (false) or 1 (true).

eol An end-of-line or a semicolon.

Instruction-list A sequence of one or more instructions. The instruction list can either be a single statement or a series of statements, each ended by a semicolon (;) or by the end of the line.

Examples

The following example causes SUMMARY_MONTH to be run 12 times with the Query Manager variable MONTH set to values from 1 to 12. The procedure SUMMARY_MONTH contains a GET CURRENT command to get the value for the Query Manager variable MONTH.

```
q = 1
Do until q > 12
  'set local (month = q'
  'Run proc summary_month'
  q = q + 1
End
```



Remarks

- The repeated instruction list, or loop, may be exited using the LEAVE statement.
- The loop is processed at least once since the expression is evaluated at the end of the loop.



Purpose

To repeat an instruction list zero or more times based on a true/false test. While the test is true, the instruction list is repeated.

Syntax

```
— DO WHILE — expression — eol — instruction-list — END —————
```

Parameters

<i>Expression</i>	The expression can be a predicate or any expression that evaluates to a logical value of 0 (false) or 1 (true).
<i>eol</i>	An end-of-line or a semicolon.
<i>Instruction-list</i>	A sequence of one or more instructions. The instruction list can either be a single statement or a series of statements, each ended by a semicolon (;) or by the end of the line.

Examples

The following example will run the procedure ALWAYSUP ten times.

```
a = 10  
Do while a > 0; 'run proc AlWaYsUp'; a = a - 1; end
```

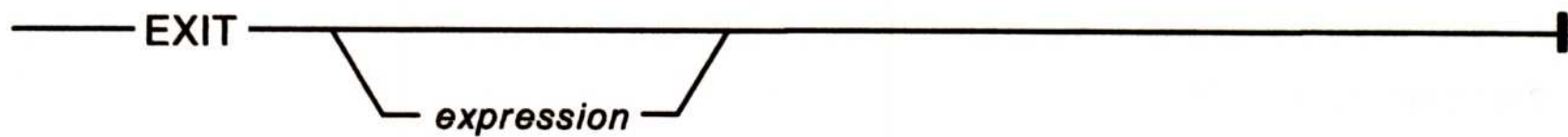
Remarks

- The repeated instruction list, or loop, may be exited using the LEAVE statement.
- The loop may not be processed at all since the expression is evaluated at the beginning of the loop.



Purpose

To end the procedure wherever it is encountered. The expression is returned to the calling function as a return code, and is contained in the variable RC.

Syntax**Parameters*****Expression***

The expression can be any expression that evaluates to a number. It cannot be a character string.

Examples

In the following example, all the EXITs are equivalent because they return a return code of zero.

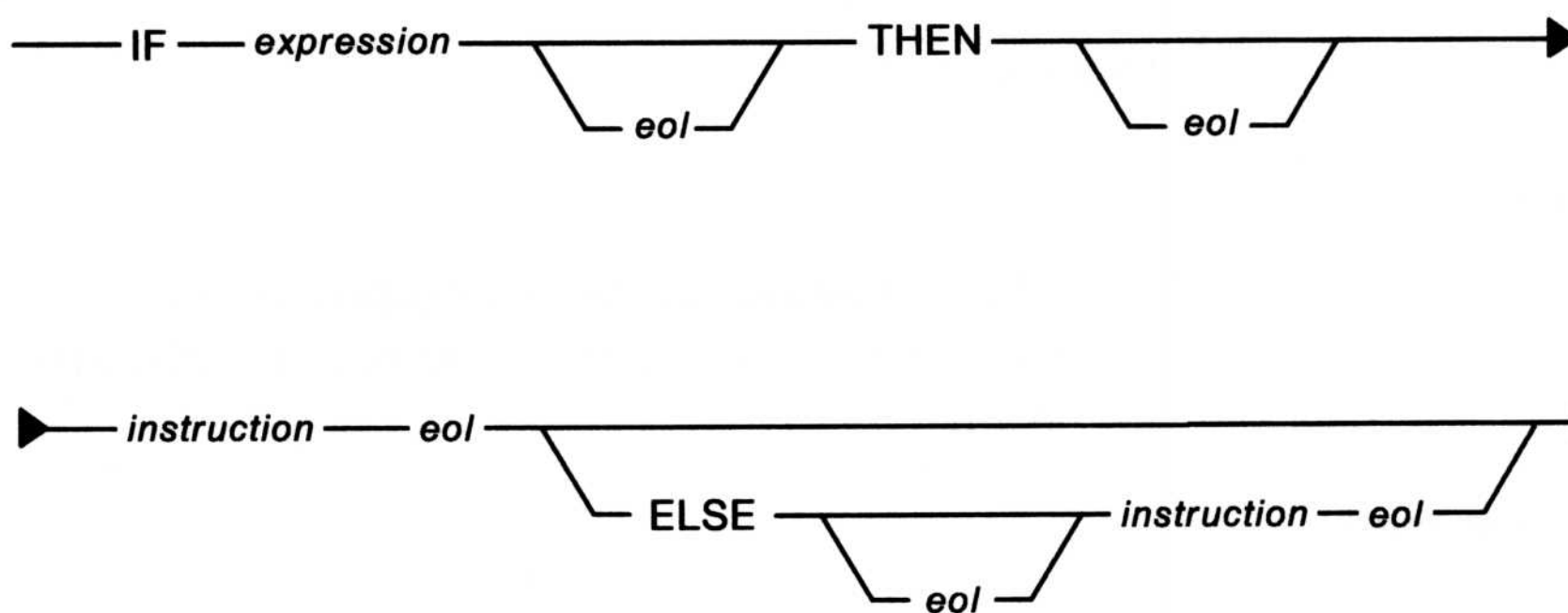
```
GOOD = 0
Exit GOOD      /* GOOD is a procedure variable */
Exit good
Exit           /* equivalent to EXIT 0 */
Exit 0
```



Purpose

To test an expression for true or false. If the expression is true, the instruction following the THEN will be executed. If the test is false, the THEN instruction is skipped and the instruction following the ELSE is executed.

Syntax



Parameters

Expression

The expression can be a predicate or an expression that evaluates to logical value of 0 (false) or 1 (true).

eol

An end-of-line or a semicolon.

Instruction

A single instruction or a DO END group.



IF THEN ELSE

Examples

In the following example, the procedure SALES_BY_MONTH is run if MONTH is equal to YES.

```
if MONTH = 'YES' then 'RUN PROC SALES_BY_MONTH'
```

In the following example, the procedure SALES_BY_MONTH is run if MONTH is equal to YES; otherwise the procedure SALES_BY_WEEK is run.

```
IF MONTH = 'YES'  
  THEN  
  'RUN PROC SALES_BY_MONTH'  
  ELSE  
  'RUN PROC SALES_BY_WEEK'
```

In the following example, if MONTH is equal to NO then the report is printed based on query SALESW; otherwise the report is printed based on query SALESM.

```
if month = 'NO'  
  then  
  do  
    'run query salesw'  
    'print report (form=sales)'  
  end  
else  
  do  
    'run query salesm'  
    'print report (form=sales)'  
  end
```

LEAVE



Purpose

To exit from within any DO statement except DO END.

Syntax

— LEAVE —

Examples

In the following, LEAVE causes the DO loop to be exited if there is an error while running the query. The query would have the global variable RPTDAY in it.

```
do day = 1 to 5
  'set global (rptday = day)'
  'Run QUERY retype'
  if rc > 0; then leave
  'Print REPORT (Layout=retype)'
end
```



Purpose

To write a character string to the display. The character string is displayed in a panel.

Syntax

```
— SAY — expression —————
```

Parameters

Expression The expression can be any expression that evaluates to a number, a character string, or a logical value of 0 (false) or 1 (true).

Examples

The following example causes the error message in variable ERRMSG to be displayed to the user.

```
errmsg = 'Error in report'  
say errmsg
```

The following example causes the successful message to be displayed to the user.

```
say "Payroll report completed successfully"
```

The following example causes the completion message (with time) to be displayed to the user.

```
"get global (T1 = time"           /* access the global variable  
                                  to get the system time */  
say "FUNCTION COMPLETED AT " || t1 /* say when the function  
                                  is completed      */
```



Purpose

To control the tracing of execution of a procedure. TRACE is primarily used for debugging. TRACE produces an OS/2 file called QRWPROC.TRC that contains each procedure line that was executed and the result of any expressions. QRWPROC.TRC will be created in the directory SQLLIB.

Syntax



Parameters

O(ff) Specifies that procedure lines and the final results of any expressions evaluated will not be written to an OS/2 file as each line is executed. This is the default if no TRACE statement is specified in a procedure.

R(esults) Specifies that procedure lines and the final results of any expressions evaluated will be written to an OS/2 file as each line is executed.

Examples

The following turns off the trace.

```
trace o
```

The following specifies that the trace is to be turned on.

```
TRACE R
```

Remarks

- The QRWPROC.TRC file is created in the SQLLIB directory, if this file does not exist. When a procedure starts tracing and a file by that name already exists, the trace information will be appended to that file without issuing a message. If the file already exists and cannot be opened because it is being used by another process, an error message is issued and the procedure ends.



TRACE

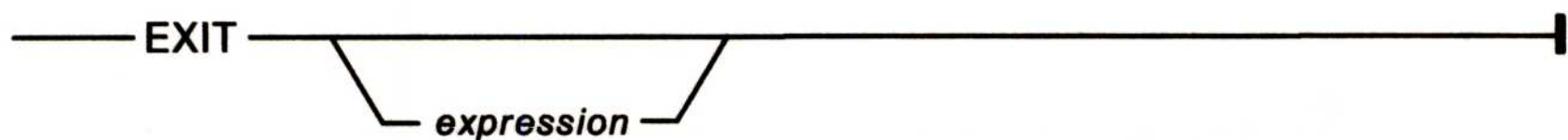
If the QRWPROC.TRC file exists when you start Query Manager, it is erased without issuing a message.

- If the QRWPROC.TRC file exists when a procedure begins tracing (because of a trace in a previous procedure), the new trace will append to the existing file.
- You can turn the TRACE command on and off within a procedure.
- To avoid possible conflicts, it is recommended that you run only one instance of Query Manager using TRACE.
- If a procedure using TRACE calls another procedure that uses TRACE, the trace information will be interspersed in the file.

Appendix A. How to Read the Syntax Diagrams

The syntax diagram shows you how to enter a command so it can be interpreted by the OS/2 program. Items in a syntax diagram are called the *command syntax*.

The command name and items required to make the command work are displayed on the baseline. Items below the baseline are optional. For example, *expression* is optional in the following:

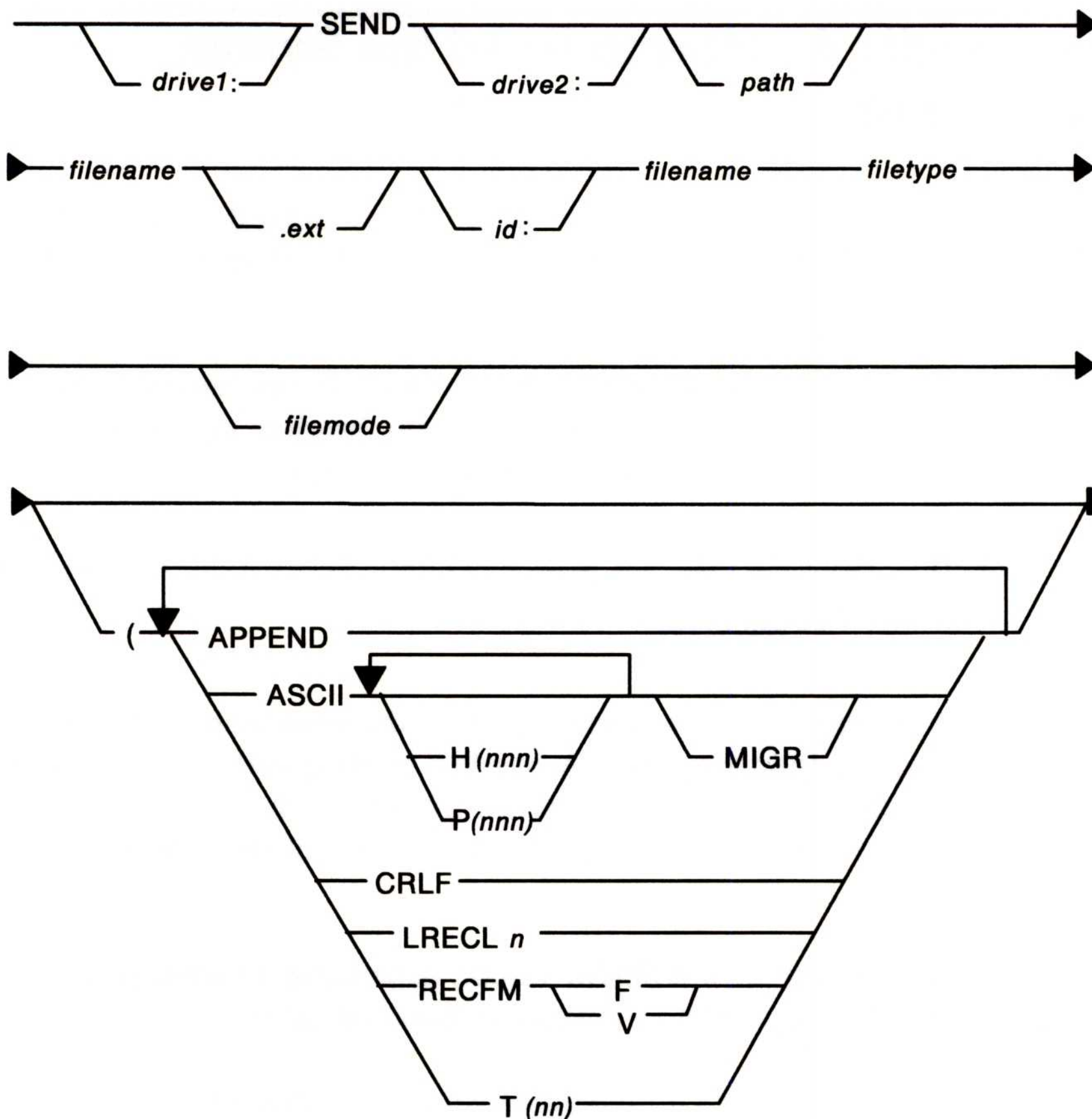


A baseline ending with an arrowhead means the command syntax is continued on the following line. A baseline starting with an arrowhead means the command syntax is continued from the line above. A vertical bar at the end of a baseline marks the end of the command syntax.

Variable parameters have different values, depending on what you enter. Variables are shown in lowercase italicized letters.

A keyword parameter has a predefined value. A command name is a keyword. Keywords are shown in capitalized letters. Note that a parameter can be a combination of a keyword and a variable.

The following example shows the use of arrowheads, vertical bars, variable parameters, and keywords:



Include all punctuation shown in the preceding example, such as colons, commas, parentheses, and question marks. In command syntax, the underscored parameter indicates the default. In command syntax, the `b` symbol indicates a required space; a comma separates repetitive variable parameters. In command examples, spaces are used for readability.

Commands and their parameters can be entered, for example, from the OS/2 full-screen command prompt, with uppercase or lowercase letters, or a combination of both. In this book, uppercase letters are used in command examples.

Internal and External Commands

If a command is an internal command, the command name is the first item in the syntax, as in this example:

EXIT _____|

Since EXIT has no parameters, type:

EXIT

If the command is an external program file, its name is preceded by an optional drive and path, as in this example:



_____|
 drive path FDISK

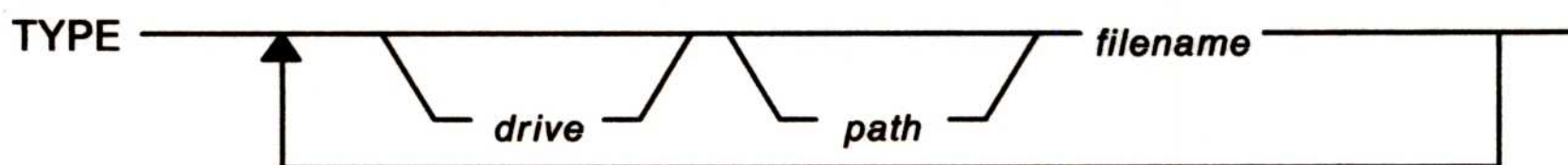
This means that you may have to enter the path to the directory that contains OS/2 external commands. For example, if the PATH command created by system installation for your CONFIG.SYS file has been changed, and a path to OS/2 external commands no longer exists, you may have to type:

C:\OS2\FDISK

The symbol tells you that this command can be used only at the OS/2 command prompt. You can type up to 254 characters, including the command name, at the OS/2 command prompt. The DOS command prompt is still restricted to 128 characters. You can type up to 1024 characters, including the command name, in an OS/2 batch file.

Specifying a Sequence More Than Once

An arrow that returns to a previous point on the baseline means that the sequence of items (such as drive, path, and file name) included by the arrow can be specified more than once, as in this example:



There are some commands that allow you to specify multiple file names in OS/2 mode. You can type up to 254 characters, including the command name, at the command prompt. Be sure that you do not separate the parts of a file specification: drive, path, and file name. For example, you can type:

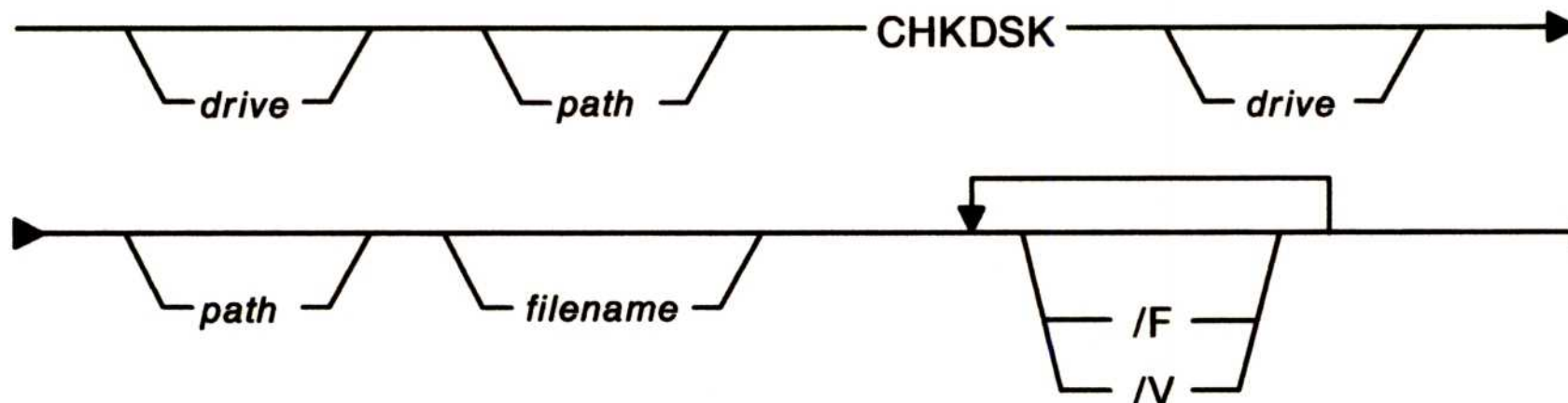
```
TYPE C:\MYDIR\MYFILE.TXT A:\MEMO.TXT D:\REPORTS\APRIL
```

Note: In command syntax, *filename* can consist of one through eight characters followed by an extension of zero through three characters. The extension for *filename* in the syntax diagram is optional, unless otherwise indicated in the list of parameters that follows the syntax of the command. When an extension is specified for a file name, it must be separated from the first part of the *filename* by a period, as in this example:

```
MYFILE.TXT
```

Specifying Items Once in Any Order

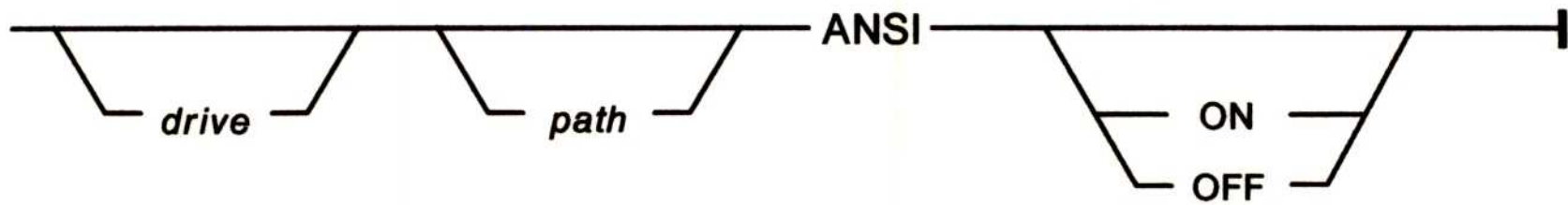
A stack of parameters with a return arrow above the baseline means that one or more of the parameters can be specified in any order, but each parameter can be specified only once.



In the preceding example, if you want to specify both the /F and /V parameters, you can specify either one before the other.

Specifying One from a Stack

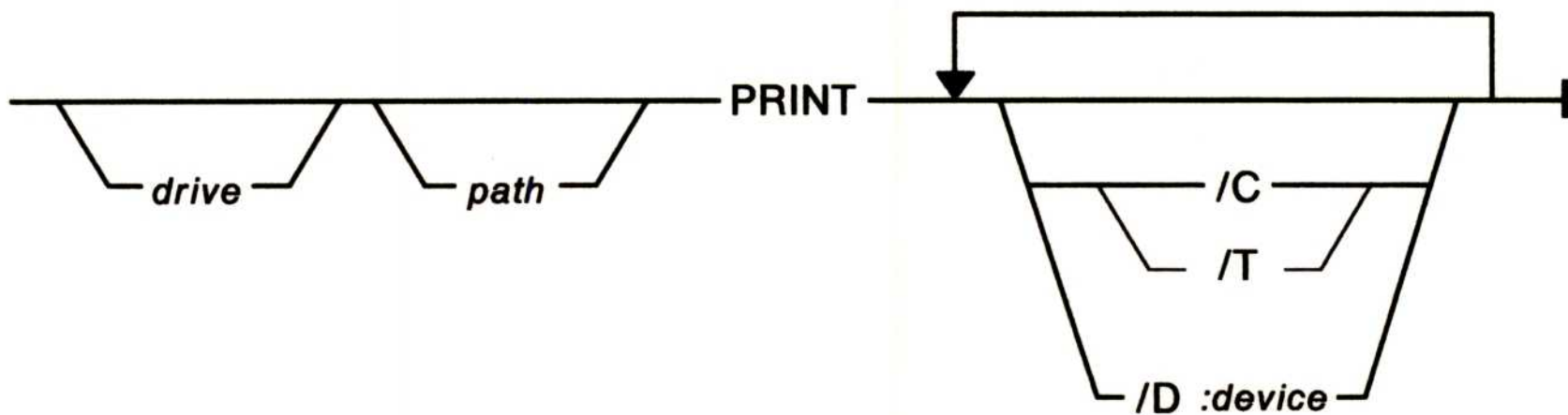
A stack of parameters with no return arrow means that you cannot choose more than one from the stack.



In the preceding example, you can choose ON or OFF.

Choosing One from within a Stack

A stack of parameters with no return arrow within a stack that has a return arrow above the baseline means that you can choose only one from the smaller stack, in any order with the parameters in the larger stack, as in this example:



In this example, you can specify /D:device and /C once in any order, or you can specify /D:device and /T once in any order, but you cannot specify /C and /T together.

Appendix B. Quick Reference Tables for Commands

The tables in this appendix give an overview of OS/2 commands. The commands are separated into eleven task-oriented tables to help you associate groups of related commands and to identify particular tasks with their appropriate command. More detailed information and the syntax for a particular command can be found on the page listed in the *Task* column.

Using Configuration Statements
Using Device Statements
Obtaining or Changing System Operations
Managing Your Files
Creating and Maintaining Your Directories
Preparing and Maintaining Your Disks and Diskettes
Piping and Filtering Input and Output
Managing Your Country Information
Using Batch File Commands
Using Memory Statements
Problem-Determination and System Software Repair.

Commands for One Mode Only

OS/2 has two modes of operation:

OS/2 mode
DOS mode

There are some commands that provide functions specific to one mode. In this book, these symbols identify commands that work in only one mode:

OS/2 only



DOS only



If you type a command that has a symbol for one mode while you are in the other mode, an error message is displayed.

Note: There are some IBM Personal Computer DOS Version 3.30 commands that are not supported or have changed slightly for DOS mode. See DOS Compatibility in the “Using the Base Operating System” section of the *User's Guide*.

Using Configuration Statements

The CONFIG.SYS file contains a list of statements that set up your system. The following tasks can be performed by including the appropriate configuration statement in your CONFIG.SYS file:






Statement	Task
<input type="checkbox"/> BREAK	Turns ON or OFF the check for whether you have pressed the Ctrl and Break keys together during program processing (page 2-9).
BUFFERS	Sets the number of disk buffers for the system (page 2-11).
CODEPAGE	Selects system code pages to be prepared by OS/2 for code page switching (page 2-18).
COUNTRY	Sets the country-dependent information (page 2-25).
DEVICE	Specifies the path and file name of a device driver to be installed (page 2-31). See also page B-5 for the table of device statements.
DEVINFO	Prepares a device for code page switching (page 2-63).
DISKCACHE	Assigns storage to be used for a disk cache (page 2-67).
<input type="checkbox"/> FCBS	Sets file control block information (page 2-76).
<input type="checkbox"/> IOPL	Allows I/O privilege to be granted or denied to requesting processes (page 2-86).
<input type="checkbox"/> LIBPATH	Identifies the locations of dynamic link libraries (page 2-93).
MAXWAIT	Sets time limit for lack of access to processor resource (page 2-94).
<input type="checkbox"/> MEMMAN	Selects storage allocation options for (page 2-95).

Statement	Task
PAUSEONERROR	Allows or prevents pausing when error or warning message is issued during the processing of CONFIG.SYS (page 2-109).
PRIORITY	Selects priority calculation in scheduling regular class threads (page 2-112).
PROTECTONLY	Specifies an OS/2-only operating environment (page 2-115).
<input checked="" type="checkbox"/> PROTSHELL	Specifies the user interface program and the OS/2 command processor (page 2-116).
REM	Allows you to add remarks to your CONFIG.SYS file (page 2-118).
<input type="checkbox"/> RMSIZE	Sets the amount of storage used for DOS programs (page 2-124).
<input checked="" type="checkbox"/> RUN	Loads and starts a system during system startup (page 2-125).
SET	Sets one string value in the environment equal to another string for later use in programs (page 2-128).
<input type="checkbox"/> SHELL	Specifies the command processor for DOS mode (page 2-132).
<input checked="" type="checkbox"/> SWAPPATH	Specifies the location of the swap file and the minimum free disk space (page 2-143).
<input checked="" type="checkbox"/> THREADS	Determines the maximum number of independent actions, known as threads (page 2-145).
TIMESLICE	Sets the amount of processor time the task receives before it must give up the processor (page 2-147).
TRACE	Selects or sets the tracing of system events (page 2-148).
TRACEBUF	Sets the size of the system trace buffer (page 2-150).

Using Device Statements







A device statement is a configuration statement that specifies a device driver to be installed. You can perform the following tasks by including the appropriate DEVICE statement in your CONFIG.SYS file. For more information on device drivers, refer to the page numbers given in the table or find them in the *User's Guide*. To install devices after OS/2 installation, see the DDINSTAL command (page 2-29).

Device Driver	Task
<input type="checkbox"/> ANSI.SYS	Allows extended display and keyboard support mode (page 2-32).
<input checked="" type="checkbox"/> ASYNCDDx.SYS	Allows OS/2 application programs to use serial devices COM ports through the ACDI (page 2-33).
<input checked="" type="checkbox"/> COM0x.SYS	Allows OS/2 application programs or system programs, such as SPOOL, to use serial devices (page 2-35).
<input checked="" type="checkbox"/> DFTDD.SYS	Allows Communications Manager 3270 terminal emulation (DFT) to access 3270 emulation adapter cards for use in communicating with 3274/3174 control units in DFT mode as well as the System/9370 (page 2-38).
<input type="checkbox"/> EGA.SYS	Supports the EGA register interface (page 2-39).
EXTDSKDD.SYS	Allows access to an external diskette drive by using a logical drive letter (page 2-40).
<input checked="" type="checkbox"/> LANDD.SYS	Allows OS/2 application programs to use the NETBIOS or Device Driver Interface (DDI) (page 2-44).
MOUSExxx.SYS	Implements support for pointing devices (page 2-45).

Device Driver	Task
 NETBDD.SYS	Allows OS/2 application programs to use the NETBIOS application programming interface (API) (page 2-49).
 PCNETDD.SYS	Allows users to use the IBM PC Network (page 2-51).
 PMDD.SYS	Provides pointer draw device driver support for the Presentation Manager (page 2-53).
POINTDD.SYS	Provides mouse pointer draw support (page 2-54).
 SDLCDD.SYS	Allows OS/2 application programs (such as Communications Manager 3270 terminal emulation) to use SDLC (Synchronous Data Link Control) devices for communications (page 2-55).
 TRNETDD.SYS	Allows user to use the IBM Token-Ring Network (page 2-58).
VDISK.SYS	Installs a virtual disk (page 2-60).

Obtaining or Changing System Operations

The following commands help you to obtain the status of different commands and change some of your system's operation from the command prompt. For more information on the individual commands, refer to the page numbers given in the table or find them in the *User's Guide*.

Command	Task
 ANSI	Allows or prevents extended display and keyboard support (page 2-2).
 ASSIGN	Assigns a drive letter to a different drive (page 2-5).
CLS	Clears the display screen (page 2-16).
 CMD	Starts another OS/2 command processor (page 2-17).
 COMMAND	Starts another DOS command processor (page 2-19).
DATE	Displays or sets the system date from the command prompt (page 2-28).
 DETACH	Starts a noninteractive program (page 2-30).
EXIT	Ends the current command processor and returns to the previous one, if one exists (page 2-74).
HELP	Provides a help line as part of the command prompt, a help screen, and information related to warning and error messages (page 2-84).
 JOIN	Joins a disk drive to a specific path (page 2-87).
MODE	Sets operational modes for devices (page 2-97).
PRINT	Prints or cancels printing of one or more files (page 2-110).

Command	Task
PROMPT	Selects and changes the command prompt (page 2-113).
SET	Sets one string value in the environment equal to another string for later use in programs (page 2-128).
<input type="checkbox"/> SETCOM40	Sets the COM port (page 2-130).
<input checked="" type="checkbox"/> SPOOL	Intercepts and separates data from different sources going to the printer so that printer output is not intermixed (page 2-135).
<input checked="" type="checkbox"/> START	Starts another OS/2 mode program in another session (page 2-137).
<input type="checkbox"/> SUBST	Substitutes a drive letter for another drive and path (page 2-141).
TIME	Displays or changes the time known to the system and resets the time on your computer's clock (page 2-146).
VER	Displays the OS/2 version number (page 2-155).


Managing Your Files

The following commands help you maintain and work with your files. For more information on file commands, refer to the page numbers given for each command in the table, or find them in the *User's Guide*.

Command	Task
ATTRIB	Turns ON or OFF the read-only and archive attributes of a file (page 2-6).
BACKUP	Backs up one or more files from one disk to another (page 2-7).
COMP	Compares the contents of the first set of specified files with the contents of the second set of specified files (page 2-21).
COPY	Copies one or more files and combines files (page 2-22).
ERASE or DEL	Deletes one or more files (page 2-73).
FIND	Searches for a specific string of text in a file (page 2-78).
PRINT	Prints or cancels printing of one or more files (page 2-110).
RENAME or REN	Changes the name of a file (page 2-119).
REPLACE	Selectively replaces files (page 2-120).
RESTORE	Restores one or more backup files from one disk to another (page 2-121).
TYPE	Displays the contents of a file (page 2-153).
UNPACK	Decompresses files that have been compressed. Copies files that are not compressed. Compressed files are designated by an @ in their file extension (page 2-154).



Creating and Maintaining Your Directories

The following commands help you maintain and work with your directories. For more information on directory commands, refer to the page numbers given for each command in the table or find them in the *User's Guide*.

Command	Task
 APPEND	Tells the system where to locate data files outside of the current directory when this command is added to your AUTOEXEC.BAT file (page 2-3).
CHDIR or CD	Changes the current directory or displays its name (page 2-14).
DIR	Lists the files in a directory (page 2-66).
MKDIR or MD	Creates a new directory (page 2-96).
RMDIR or RD	Removes a directory (page 2-123).
TREE	Displays all of the directory paths (page 2-152).
XCOPY	Selectively copies groups of files, including those in subdirectories, from one disk to another (page 2-158).

Preparing and Maintaining Your Disks and Diskettes

The following commands help you prepare and maintain your disks and diskettes. For more information on disk and diskette commands, refer to the page numbers given for each command in the table or find them in the *User's Guide*.

Command	Task
CHKDSK	Scans a disk and checks it for errors (page 2-15).
 DDINSTAL	Provides an automated way to install device drivers. (page 2-29).
DISKCOMP	Compares the contents of two diskettes (page 2-68).
DISKCOPY	Copies the contents of one diskette to another diskette (page 2-69).
 FDISK	Creates and deletes partitions and logical drives, changes the primary partition, displays partition data, and selects the next fixed disk drive for partitioning (page 2-77).
FORMAT	Formats the disk to accept OS/2 files (page 2-80).
LABEL	Creates, or changes the volume identification label on a disk (page 2-92).
RECOVER	Recovers files from a disk containing defective sectors (page 2-117).
REPLACE	Selectively replaces files (page 2-120).
VERIFY	Confirms that data written to a disk is correct (page 2-156).
VOL	Displays the disk volume label (page 2-157).




Piping and Filtering Input and Output

Piping is the chaining of programs with automatic redirection of standard input and output. A filter is a program or command that reads data from the standard input device, modifies the data, and writes the result to the standard output device. The following commands help you to redirect your standard input and output. For more information on these commands, refer to the pages given in the table or find them in the *User's Guide*.

Command	Task
FIND	Searches for a specific string of text in a file (page 2-78).
MORE	Sends output from a file to the screen, one full screen at a time (page 2-105).
SORT	Sorts information by letter or number (page 2-134).

Using Batch File Commands

A batch file is a file containing one or more commands that the base operating system starts one at a time. For more detailed information on batch file commands, refer to the page numbers given for each command in the table or find them in the *User's Guide*.

Command	Task
CALL	Nests a batch file within a batch file (page 2-12).
ECHO	Allows or prevents the display of OS/2 commands while a batch file is running (page 2-71).
 ENDLOCAL	Restores the drive, directory, and variables that were in effect before a SETLOCAL command was issued (page 2-72).
 EXTPROC	Defines an external batch file processor (page 2-75).
FOR	Allows repetitive processing of commands within a batch file (page 2-79).
GOTO	Transfers batch processing to a specified label (page 2-82).
IF	Allows conditional processing of commands within a batch file (page 2-85).
PAUSE	Suspends processing of the batch file (page 2-108).
REM	Displays remarks from within a batch file (page 2-118).
SET	Sets an environment variable equal to a string for later use (page 2-128).
 SETLOCAL	Sets the drive, directory, and variables that are local to the current batch file (page 2-131).
SHIFT	Allows more than 10 replaceable parameters in a batch file (page 2-133).



Using Memory Statements

The following statements can be used for tasks requiring the use of memory if you include the appropriate statement in your CONFIG.SYS file. For more information on memory statements, refer to the page numbers given for each statement in the table or find them in the *User's Guide*.

Statement	Task
BUFFERS	Sets the number of disk buffers for use (page 2-11).
DISKCACHE	Assigns storage to be used for a disk cache (page 2-67).
<input checked="" type="checkbox"/> MEMMAN	Selects storage allocation options (page 2-95).
PROTECTONLY	Specifies an OS/2-only operating environment (page 2-115).
<input type="checkbox"/> RMSIZE	Sets the amount of storage used for DOS programs (page 2-124).
<input checked="" type="checkbox"/> SWAPPATH	Specifies the size and location of the swap file (page 2-143).

Managing Your Country Information



The following commands help you manage and maintain the country information necessary when changing to a different language code. For more information on country commands, refer to the page numbers given for each command in the table or find them in the *User's Guide*.

Command	Task
CHCP	Displays or changes the current system code page (page 2-13).
CODEPAGE	Selects system code page prepared by OS/2 for code page switching (page 2-18).
COUNTRY	Sets the country-dependent information (page 2-25).
DEVINFO	Prepares a device for code page switching (page 2-63).
 GRAFTABL	Allows the extended code page that is displayed when using display adapters in graphics mode (page 2-83).
 KEYB	Specifies a special keyboard layout that replaces the current keyboard layout (page 2-89).

Problem Determination and System Software Repair

The base operating system has programs that monitor the operating system and assist in problem determination. These specialized programs are to be used with assistance from an IBM service representative. The following commands are included in this book so that you can refer to them when talking to your IBM service representative.

The base operating system has a program that applies IBM-supplied patches to software problems. Occasionally, this command is intended to be used with assistance from an IBM service representative only after IBM is contacted and problem determination takes place. For more information on problem-determination or system software repair commands, refer to the page numbers given for each command in the table or find them in the *User's Guide*.

Command	Task
 CREATEDD	Creates a dump data diskette (page 2-27).
PATCH	Allows you to apply IBM-supplied patches to make repairs to software (page 2-106).
TRACE	Selects or sets system trace (page 2-148).
 TRACEFMT	Displays formatted trace records in reverse time stamp order (page 2-151).
TRACEBUF	Sets the size of the system trace buffer (page 2-150).

Glossary

A

absolute priority. In the OS/2 program, pertaining to a priority of a process that is not varied by the operating system. Contrast with *dynamic priority*.

access. The manner in which files or data are referred to by a computer.

access path. In Database Manager, the path used to get to data specified in SQL statements. An access path can involve an index or a sequential search, or a combination of the two.

access plan. In Database Manager, a database object stored in the database that includes all of the information needed to process the Database Services statements of a single application program. An access plan is generated processing of the SQLBIND program or through the precompile process if the bind option is used. See *application plan*.

access priority. In the IBM Token-Ring Network, the maximum priority a token can have for the adapter to use for transmission.

access unit. See *multistation access unit*.

ACDI. See *Asynchronous Communications Device Interface*.

ACDI port. A serial port such as COM1, COM2, or COM3 that can

be programmed for asynchronous communications through ACDI.

action bar. The highlighted area at the top of a panel that contains the choices currently available in the application program that a user is running.

action text. Text added to panels and menus created in Query Manager that indicates the action a user needs to perform.

activate. To make an object, program, system, or device available for use.

active. Pertaining to anything that is current or operational.

active program. A program currently running on the computer. Contrast with *inactive program*.

active window. The window with which the user is currently interacting. The active window has a colored title bar, a yellow or shaded border, and is usually the topmost window.

adapter. A piece of hardware that connects a computer and an external device.

adapter number. A specific number that identifies an adapter when more than one adapter is used in a workstation.

address. A value that identifies the location of a register, a particular part of storage, or a network node.

American National Standard Code for Information Interchange (ASCII). The standard code with a coded character set consisting of 7-bit coded characters (8 bits including parity check), used for information interchange among data processing systems, and data communication systems. The ASCII set consists of control characters and graphic characters.

American National Standards Institute (ANSI). An organization sponsored by the Computer and Business Equipment Manufacturers Association for establishing voluntary industry standards.

ANSI. See *American National Standards Institute*.

API. See *application programming interface*.

append. To attach a record or file to the end of another file.

application. A program or set of programs that perform a task, for example, a payroll application.

application plan. The control structure produced during the bind process and used by Database Services to process Structure Query Language (SQL) statements encountered during application processing. See *access plan*.

archive. A copy of one or more files or a copy of a database that is saved for future reference or for recovery purposes in case the original data is damaged or lost.

argument. A parameter passed between a calling program and a called program.

arithmetic expression. An expression that contains arithmetic operations and operands that can be reduced to a single numeric value.

arithmetic function. A function that represents one of the basic arithmetic operations such as addition, subtraction, division, and multiplication. It can also be a mathematical operation such as finding the average, minimum, or maximum value of a set of values.

arithmetic operator. A symbol used to represent a mathematical operation, such as the plus sign (+), which is used to represent addition, or the minus sign (-), which is used to represent subtraction.

arithmetic overflow. A condition in which the result of a mathematical operation exceeds the storage or register capacity.

ASCII. See *American National Standard Code for Information Interchange*.

ASCII Terminal Emulation. A feature of Communications Manager that emulates the function of an asynchronous terminal.

asynchronous. Pertaining to the lack of regular time relationship; unexpected or unpredictable with respect to the execution of program instructions. Contrast with *synchronous*.

asynchronous communication.
(1) Data communication in which

communication of a character or a block of characters can begin at any time, but in which the bits that represent the character or block have equal time duration. (2) Data transmission in which transmission of a character or a block of characters can begin any any time but in which the bits that represent the character or block have equal time duration.

Asynchronous Communications Device Interface (ACDI). An API for asynchronous communications provided by Communications Manager.

attribute. A characteristic or property, for example, the color of a line, or the length of a data field.

authorization ID. In Database Manager, a sequence of characters that identifies a user.

automatic bind. In Database Manager, a feature that automatically binds an invalidated application plan without requiring a manually entered SQLBIND command.

B

backspace (character backspace). A typing action that deletes the character to the left of the cursor and then shifts to the left all the characters remaining the field or line.

base operating system. IBM Operating System/2.

base table. In Database Manager, a table created with the CREATE

TABLE statement or through the Query Manager prompted interface. A table with both its description and data physically stored in the database.

batch file. A file containing DOS commands organized for sequential processing while in DOS mode. For OS/2 mode, see *command file*.

bind file. A file produced by the Database Services precompiler when the NOBIND option is specified. This file includes information on all Structure Query Language (SQL) statements in the application program.

binding. The process of installing an application into a database. Binding is performed either directly during an application program precompilation, or through a SQLBIND program execution that uses the output of a precompilation.

block. (1) A string of data elements recorded or transmitted as a unit. (2) To wait, usually for an I/O event to complete or for a resource to become available.

block size. The number of records, words, or characters in a block.

break levels. In Query Manager, the feature used to group data from a query which is then displayed in a report.

buffer. A memory area reserved for use in performing input/output operations.

buffer pool. A memory area reserved to satisfy the buffering requirements for a function.

built-in function. (1) A function that is supplied by a programming language. (2) In Database Manager, a scalar function or column function.

C

cancel. (1) To end a task before it is completed. (2) An action that removes the current panel or window without processing it and returns to the previous one. See also *escape key*.

Caps Lock. A typing action resulting from pressing the Caps Lock key that causes the character keys to produce uppercase letters. The keyboard remains in Caps Lock mode until the user presses the Caps Lock key again.

carriage return. An operation that prepares for the next character or cursor to be printed or displayed at the specified first position on the next line.

carrier return. An indication to continue printing at the left margin of the next line.

catalog. A set of system tables maintained by Database Services. Catalog tables are created when the database is created and contain information about tables, views, and indexes.

CHAR(n). In Structured Query Language (SQL), a column data type that is a character string with a fixed length of n , where n can be any value between 1 and 254. This is also known as character (fixed length) in Query Manager.

character. (1) A letter, digit, or other symbol that is used as part of the organization, control, or representation of data. (2) A column data type in Query Manager.

character constraint. A limitation placed by an information processing system on character formats.

character data. A type of data in the form of letters, digits, and special characters such as punctuation marks.

character string. (1) A sequence of bytes or characters associated with a single byte character set. (2) A sequence of printable characters. (3) A string of characters, such as a command and its parameters, used to communicate with the operating system.

character string delimiter. In Database Manager, the characters used to enclose character strings in delimited ASCII (DEL) files that are imported or exported. The default is a quotation mark.

choice. An item a user can select.

clause. In Structured Query Language (SQL), a distinct part of a statement, such as a WHERE clause.

code page. (1) A table that defines a coded character set by assignment of a character meaning to each code point in the table for a language or country. (2) A mapping between characters and their internal (binary) representation.

column. (1) A vertical arrangement of data. (2) In Database Manager, the vertical component of a table.

column data type. A data type used in Database Manager to specify the characteristics of a column when defining a table for a database.

column delimiter. In Database Manager, the character used to enclose columns in delimited ASCII (DEL) files that are imported or exported. The default is a comma.

column function. In Database Manager, an operation performed on a column, or columns, that produces one value from a set of values. A column function is expressed in the form of a function name followed by an argument enclosed in parentheses, for example, SUM(COMM + SALARY).

column heading. In Query Manager, an alternative to the column name that a user can specify on a form, table, or report.

column name. A unique name given to each column of a table in a database.

command. The name and any parameters associated with an action that can be performed by a program. The command is entered by the user; the computer performs the action requested by the command name.

command file. A file containing OS/2 commands organized for sequential processing while in OS/2 mode. For DOS mode, see *batch file*.

command interface. (1) The method used to enter commands at the OS/2 or DOS prompt. (2) A statement that is in a procedure or entered on

the Query Manager command line that processes a specified Query Manager command.

command name. The verb in a command that specifies the action to be performed.

command processor. A program that performs an operation specified by a command.

command prompt. A displayed symbol, such as C: > that requests input from a user.

comment. Optional text that describes an object or statement.

commit. A process that causes data changed by an application or user to become part of a database.

commit point. See *point of consistency*.

communication. The transmission and reception of data.

communication port. (1) An access point for data entry or exit to or from a communication device such as a terminal. (2) On a personal computer, a serial port to which a modem can be attached.

Communications Manager. A component of OS/2 Extended Edition that lets a workstation connect to a host computer and use the host resources as well as the resources of other personal computers to which the workstation is attached, either directly or through a host. Communications Manager provides APIs so users can develop their own applications.

comparison operator. A symbol (such as =, >, or <) used to specify a relationship between two values.

component. A functional part of an operating system or program.

computed fields. In Query Manager panel definition, fields that take their value from a computation of other field values or row values.

concatenation. The method of combining two strings into a single string by appending the second to the first.

concurrency control. In Database Manager, a feature that allows multiple users to execute database transactions simultaneously without interfering with each other.

configuration. The task of defining the devices, features, parameters, and programs for a system. Synonymous with *customize*.

configuration file. (1) In Communications Manager, a file that describes the devices, optional features, communications parameters, and programs installed on a workstation. (2) In Database Manager, a file containing values that can be set to adjust the performance of Database Manager. (3) For the base operating system, the CONFIG.SYS file that describes the devices, system parameters, and resource options of a workstation.

configure. (1) To prepare a workstation component or program for operational use. (2) To describe to a system, the devices, optional

features, and programs installed on the system.

CONFIG.SYS. A file that contains configuration options for an OS/2 Extended Edition workstation. See also *configuration file*.

connection. An association established between functional units for conveying information.

consistency of data. In Database Manager, a feature that ensures that no transaction can see another transaction's uncommitted updates.

constant. A fixed value.

context-sensitive help. See *contextual help*.

contextual help. Information about a field or other panel element on which the cursor is positioned that is provided to the user upon request.

Control (Ctrl) mode. A mode that is obtained by pressing and holding the Ctrl key. Pressing and holding the Ctrl key places the keyboard in a special control mode, where pressing a second key gives predefined functions.

correlated reference. In Database Manager, the combined correlation name and column name referring to a specific column within a SELECT statement.

correlated subquery. In Database Manager, a subquery (part of a WHERE or HAVING clause) applied to a row or group of rows of the table or view names in the outer SELECT statement.

country code. A 3-digit number specifying a country and that country's preferred formats for date and time values, currency, and numeric data.

cursor. In Database Services, a named control structure used by an application program to point to a row of data.

customization tasks. A feature supported by Query Manager that includes a control language, Query Manager Command Language, customized panels, customized menus, and customized procedures.

customize. See *configuration*.

cylinder. (1) The fixed-disk or diskette tracks that can be read or written without moving the disk drive or diskette drive read/write mechanism. (2) The number of tracks for space allocation.

D

data. The coded representation of information for use in a computer. Data has certain attributes, such as type and length.

Data Definition Language (DDL). In Database Manager, a series of SQL commands used to define objects.

data type. An attribute used for defining data.

database. (1) A systemized collection of data that can be accessed and operated upon by an information processing system. (2) In Database Manager, a

collection of information such as tables, views, and indexes. With Query Manager a database can also include such other information as report forms, queries, panels, menus, and procedures.

database directory. A file maintained by Database Services that contains information about the location of databases. A volume database directory exists on every OS/2 file system where a database exists. A system database directory exists on the drive into which Database Services is installed.

database environment commands. In Database Manager, a group of programmable interfaces allowing an application program to start and stop Database Manager, connect to and disconnect from databases, create and delete databases, change the database password, scan database directories, and initiate database recovery.

database environment routines. See *database environment commands*.

Database Management System (DBMS). A computer program that manages data by providing the services of centralized control, data independence, and complex physical structures for efficient access, integrity, recovery, concurrency control, privacy, and security.

Database Manager. A component of OS/2 Extended Edition consisting of Database Services and Query Manager. Database Manager is based on the relational model of data and allows users to create, update, and access databases.

Database Services. The part of Database Manager providing the base functions and configuration files needed to use Database Manager.

database tools. Utilities which enable manipulation of database for example, import, export, and restore.

DATE. In Structured Query Language (SQL), a column data type in the form of a three-part value that designates a day, month, and year. This is also known as date in Query Manager.

DECIMAL. In Structured Query Language (SQL), a column data type that consists of numeric data that contains a decimal point. There can be a maximum of 31 digits in the number. This is also known as decimal in Query Manager.

decimal point character. In Database Manager, the character used to represent the decimal point in numeric data in delimited ASCII (DEL) files that are imported or exported. The default is a period (.).

dedicate. To assign a system resource, for example, an I/O device, a program, or a whole system, to one application or purpose.

default. An attribute, value or option assumed when none is explicitly specified.

default definition. A definition used in panels in Query Manager. A default definition is used for an uncustomized panel and is based on one table.

default form. The form provided by Query Manager to format reports for queries when a form is not specified.

DEL. A format used to export and import data formatted with delimited ASCII.

delete. To remove, for example, to delete a file.

delimited identifier. In Database Manager, an identifier enclosed between two escape characters ("").

delimiter. (1) A character or flag that groups or separates items of data. (2) In the IBM Token-Ring Network, a bit pattern that defines the limits of a frame or token on a ring network.

device. An input/output unit such as a terminal, display, or printer.

device driver. The executable code needed to attach and use a device such as a display, printer, or plotter, or communications adapter.

DFT. See *Distributed Function Terminal*.

diagnostic tool. One of the OS/2 Extended Edition utilities designed to gather and process data to help identify the cause of a problem.

directory. (1) A named grouping of files in a file system. (2) In Database Manager, a table of identifiers and references to the corresponding items of data.

disk. A magnetic disk unit. See also *diskette*.

disk operating system (DOS). An operating system for computer systems that use disks and diskettes for auxiliary storage of programs and data.

diskette. A disk enclosed in a protective container. See *disk*.

Distributed Function Terminal (DFT). (1) An operational mode that allows multiple concurrent logical terminal sessions. (2) A hardware or software protocol used for communication between a terminal and an IBM 3274/3174 control unit.

DOS. See *disk operating system*.

drive. The device used to read and write data on disks or diskettes.

dump diskette. (1) A diskette that contains a dump or that is prepared to receive a dump. (2) In the OS/2 program, a diskette (created using the **CREATEDD** command) that contains the contents of storage at a specified point in time.

duplicates. In Database Manager, (1) an option used in prompted query and prompted view definition to specify whether duplicate data rows in a query or view should be discarded or kept. (2) An option used when creating an index for a table to specify whether or not duplicate values are allowed for the set of columns that comprise the index.

duration. An interval of time, as in simple duration, date duration, or time duration.

dynamic bind. A process where dynamic Structured Query Language (SQL) statements are run in an application program and binding is performed at the moment a statement is run.

dynamic link routine (DLR). A program or routine that can be loaded by an application or as part of a program.

dynamic linking. In the OS/2 program, the delayed linking of a program to a routine so that the routine is not linked until load or run time.

dynamic (priority). In the OS/2 program, pertaining to a priority of a process that is varied by the operating system. Contrast with *absolute (priority)*.

E

EBCDIC. See *Extended Binary-Coded Decimal Interchange Code*.

edit. To add, change, delete, or rearrange data.

edit code. In Query Manager, a code used in forms and panels to indicate how the data in a particular column is formatted for displaying or printing.

embedded SQL. SQL statements embedded within a program, and prepared during the program preparation process before the program is executed.

emulation. The imitation of all or part of one system by another so

that the imitating system accepts the same data, executes the same programs, and achieves the same results as the imitated computer system.

enable. (1) To make functional. (2) The state of a processing unit that allows the occurrence of certain types of interruptions. (3) In the OS/2 program, to initiate the operation of a circuit or device.

enter. An action performed by pressing the enter key or selecting Enter from a function key area which causes the computer to receive and process user input.

error. An unexpected result from a program command or action.

escape character. The symbol used to enclose an SQL delimited identifier. This symbol is the quotation mark (").

escape key. A key that removes the current panel or window without processing it and returns to the previous one. See also *cancel*.

export. To copy data from Database Manager tables to an OS/2 file using PC/IXF, DEL, or WSF formats.

expression. An operand or a combination of operands and operators yielding a single value.

Extended Binary-Coded Decimal Interchange Code (EBCDIC). A coded character set consisting of 8-bit coded characters used by host computers.

extended partition. The area beyond the primary partition. This area can be divided into smaller areas or partitions, each of which can be assigned a drive letter and be accessed as though it were a separate fixed disk.

F

FCB. See *File Control Block*.

feature. A programming or hardware option.

field. (1) An area in a record or on a panel used to contain data. (2) In Database Manager, the smallest unit of data that can be referred to in a table from a database. A field contains one unit of information from a column within a row of data in a table.

file. A collection of related data that is stored and retrieved by an assigned name. Synonymous with *data set*.

File Control Block (FCB). A record that contains all of the information about a file for example, its structure, length, and name.

file name. (1) The name used by a program to identify a file. (2) The portion of the identifying name that precedes the extension.

file system. The collection of files and file management structures on a physical or logical mass storage device such as a disk.

filemode. The third field in the VM/CMS file identifier. The filemode indicates the mode letter

currently assigned to the virtual disk in which the file resides.

filespec. The name and location of a file. The format is dependent on the storage medium of a file, for example, c:\path\filename.ext.

filetype. Second field in the VM/CMS file identification.

final summary. In Query Manager, the summary totals for a query displayed in a report.

fixed disk. A flat, circular, nonremovable plate with a surface layer on which data can be stored by magnetic recording.

fixed-length string. A character or graphic string whose length is specified and cannot be changed.

formatted diskette. A diskette on which track and sector control information has been written. Contrast with *unformatted diskette*.

frame. A data structure that consists of fields predetermined by a protocol for the transmission of user data and control data.

full-screen mode. A form of screen presentation in which the contents of an entire screen can be displayed at once.

G

global variable. In Query Manager, a variable that exists from the time it is initially set until Query Manager is ended.

H

Help. A feature that provides assistance and information to the user.

hexadecimal. Pertaining to a numbering system with base of sixteen; valid numbers use the digits 0 through 9 and characters A through F, where A represents 10 and F represents 15.

host computer. (1) In a computer network, a computer providing services such as computation, database access, and network control functions. (2) The primary or controlling computer in a multiple computer installation.

host language. In Database Manager, a programming language in which SQL statements are embedded.

host processor. A processor that controls all or part of a network.

host program. (1) For Database Manager, a program written in a host language containing embedded SQL statements. (2) In Communications Manager, a program that runs on a host computer.

host system. The controlling or highest level system in a data communication configuration.

host variable. In a (host) application program, a variable referenced by embedded Structured Query Language statements. Host variables are programming variables in the application program, and are

the primary mechanism for transmitting data between Database Manager tables and application program work areas.

I

IBM Operating System/2 Extended Edition. A program that contains the features of OS/2 Standard Edition Version 1.1. It also contains an advanced relational Database Manager, a Communications Manager which provides intersystems communications, improved connectivity, and terminal emulation, and the LAN Requester.

icon. A pictorial representation of a choice for the user to select. Icons can represent things (such as a document or file) the user wants to work on and they can represent actions the user wants to perform.

idles. In an IBM Token-Ring Network, the signals sent when neither frames nor tokens are being transmitted.

IEEE 802.2 interface. An interface adhering to the 802.2 logical link control (LLC) Standard of the Institute of Electrical and Electronics Engineers (IEEE). This standard is one of several standards for local area networks approved by the IEEE.

import. To copy data from OS/2 files into tables in a database.

increment. To increase by a constant value.

index. In Database Manager, a collection of data about the locations of records in a table,

allowing faster access to a record using a specified key.

Index key. In Database Manager, the set of columns in a table used to determine the order of index entries. See *key*.

IND\$FILE. The default name for the host file transfer program used by the host computer to communicate with Communications Manager.

initial field value. The information in an entry field provided by an application when the field is first presented.

initial search. An operation in Query Manager that locates data for a panel which meets specified criteria.

initialize. (1) To set counters, switches, addresses, or contents of storage to zero or other starting values at the beginning of, or at prescribed points, in the operation of a computer routine. (2) To prepare an adapter for use by a program.

input. The information entered into a computer for processing or storage.

input device. A device such as a keyboard in a data processing system through which data may be entered into the system.

input field. In Query Manager panel definition, a field that accepts a value.

installation. The process of placing one or more OS/2 components on a workstation's fixed disk.

interactive processing. (1) A processing method in which each user action causes a response from a program or the system. (2) In Database Manager, a method of processing that allows users to interact with the Query Manager panels and menus while a procedure is running.

interactive program. A program that is running (active) and ready to receive (or is currently receiving) input from the user. See *active program*.

interactive session. The group of processes or tasks that currently own the keyboard, mouse, and display, or other interactive input device.

interface. (1) A set of verbs used by a program to communicate with another program. (2) A shared boundary between two or more entities. An interface might be a hardware component to link two devices together or it might be a portion of memory or registers accessed by two or more computer programs. (3) See *user interface*.

J

join. In Database Manager, a relational operation that allows for retrieval of data from two or more tables based on matching column values.

join condition. In Database Manager, a condition where two tables are joined together and compared; rows from one table are selected when columns from that table match (over a condition) columns from the other table.

K

KB. See *kilobyte*.

kernel. (1) The part of an operating system that performs basic functions such as allocating hardware resources. (2) The central part of the Database Services product. The kernel is a relational command processor.

key. (1) One or more characters used to identify the record and establish the order of the record within an indexed file. (2) In Database Manager, a column or an ordered collection of columns on which an index is created. See *Index key*.

keyword. One of the predefined words of a computer or command language.

keys help. An action in help panels that gives users a listing of all of the key assignments for the current application.

kilobyte (KB). 1024 bytes.

L

LAN. See *local area network*.

leading zero. A zero that is displayed beside the leftmost digit in a number.

line feed. An ASCII character that causes an output device to move forward one line.

link. (1) The physical medium of transmission, the protocol, and associated devices and programming

used to communicate between computers. (2) To interconnect items of data or portions of one or more computer programs; for example, the linking of object programs by a linkage editor, or the linking of data items by pointers.

link-edit. To create a loadable computer program by means of a linkage editor.

load. (1) To move data or programs into memory. (2) To place a diskette into a diskette drive.

load time. (1) The time during which a program is being loaded into memory for execution. (2) In programming, the time it takes to enter data into memory or working registers.

local area network (LAN). (1) Two or more computing units connected for local resource sharing. (2) A network in which communications are limited to a moderate-sized geographic area such as a single office building, warehouse, or campus, and that do not extend across public rights-of-way.

local variable. (1) A parameter that is defined and used only within a specified portion of the program in which it is declared. (2) In Query Manager, a variable that exists as long as the object that created it is running.

lock. (1) In Database Manager, (a) a means of serializing events or access to data; (b) an SQL statement used to acquire control of tables prior to executing statements that use them. (2) In Communications Manager, a password-protection

system that can be used to prevent access to some advanced functions. See also *keylock*.

lock escalation. In Database Services, the response that occurs when the number of locks issued exceeds the capacity specified in the database configuration. During a lock escalation, locks are freed by converting record locks for a table into one table lock. This is repeated until enough locks are freed by one or more processes.

locking. The process by which Database Manager ensures integrity of data. Locking prevents users from accessing inconsistent data.

log record. In Database Services, a record of an update to a database performed during a unit of work.

logical connector. In SQL, a condition that connects expressions within a WHERE or HAVING clause. The valid logical connectors are *and* and *or*.

logical record. A set of related data or words considered to be a record from a logical viewpoint.

long string. A fixed-or variable-length string where the maximum length is greater than 254 bytes.

M

memory. (1) The storage on electronic chips. Examples of memory are random access memory, read only memory, or registers. (2) The program-addressable storage from which instructions and other

data can be loaded directly into registers for subsequent execution or processing.

memory allocation. An operating system function that assigns memory areas to tasks.

message. (1) The information not requested by users but presented to users by the computer in response to a user action or internal process. (2) In Presentation Manager, a packet of data used for communication between Presentation Interface and windowed applications.

mode key. A key, such as CapsLock, NumLock, or Alt, that sets the keyboard in a special mode of operation different from its base state.

mouse. A device used to move a pointer on the screen.

multitasking. A mode of operation that provides for concurrent performance or interleaved execution of two or more tasks.

N

navigation. The process of moving through panels and menus.

NETBIOS. An API between a local area network adapter and programs.

non-switched line. A connection between computers or devices that does not have to be established by dialing.

notification message. A type of message that describes the status of

the user's work or some state of the system.

null. A special value that indicates the absence of information.

null character. (1) In Query Manager profiles, a character used to indicate a null data field. (2) The character hex 00, used to represent the absence of a printed or displayed character.

NumLock. A typing action that puts the keyboard into NumLock mode. As a result, the numeric keypad keys produce numbers when pressed rather than their base value. The keyboard remains in NumLock mode until the user presses the NumLock key a second time.

NumLock mode. A typing mode in which a dual-purpose numeric keypad on the keyboard is locked in the secondary purpose of inserting a number for each key depression.

numeric constant. A fixed value that is a number.

O

object. A table, view, index, query, form, procedure, profile, panel, or menu created or manipulated by using Database Manager.

object name. A sequence of characters identifying an object created by a Database Manager user.

object names menu. In Query Manager, a menu listing objects such as tables and views, queries, forms, procedures, panels, or menus.

online. (1) Pertaining to a user's ability to interact with a computer. A description of a user's access to a computer by way of a display.

(2) Pertaining to the operation of a functional unit that is under the continuous control of a computer.

online information. The information stored in a computer system that can be displayed, used, and modified in an interactive manner without any need to obtain hard copy.

operand. (1) An entity on which an operation is performed.

(2) Information entered with a command name to define the data on which a command processor operates and to control the execution of the command processor.

operating system. The software that controls the running of programs. An operating system may provide services such as resource allocation, scheduling, input/output control, and data management.

operation. (1) A well-defined action that, when applied to any permissible combination of known entities, produces a new entity; for example, the process of addition.

(2) A program step performed by a computer.

operator. A symbol that represents an operation to be performed; for example, the plus sign (+).

optimization. The determination of an efficient access strategy for satisfying a database access.

OS/2. See *IBM Operating System/2 Extended Edition*.

OS/2 file system. A file system provided by the OS/2 program and accessible through the OS/2 file I/O functions. A file system is restricted to a single medium, either an entire diskette or a single partition of one fixed disk.

output area. The area of paper within which the picture is to be printed or plotted.

output device. A device such as a printer or display in a data processing system.

P

page. (1) In Database Manager, a unit of storage within a table or index whose size is 4KB. (2) In a virtual storage system, a fixed-length block that has a virtual address and is transferred as a unit between memory and disk storage.

parameter. (1) The information supplied by a program or user to a command or function. (2) The data passed between programs or procedures.

parameter marker. The question mark symbol (?) used in a dynamic SQL statement to indicate that an application program variable will be substituted for the marker at processing time.

password. A string of characters that a program, computer operator, or user must supply to meet security requirements before gaining access to data.

path. The route used to locate files on a disk or diskette, consisting of a drive and directories.

PC/IXF. An OS/2 file format used to export and import table data.

pipe. To direct the data so that the output from one process becomes the input to another process.

placeholder. A symbol that can assume any of a given set of values.

plan. See *application plan*.

plan name. The name of an application plan. A Database Manager application plan is the output from the bind process.

point of consistency. In Database Manager, a point in time when all the recoverable data a program accesses is consistent. The point of consistency occurs when updates, inserts, and deletions are either committed to the physical database or rolled back (not committed and discarded). Synonymous with *commit point*.

pointer. The symbol that is displayed on the screen that is moved by a pointing device such as a mouse.

pointing device. An instrument such as a mouse, trackball, or joystick used to move a pointer on the screen.

precision attribute. In Database Manager, the total number of digits in a decimal type column. The precision cannot be greater than 31, and it must be odd. If precision is

specified as even, it is rounded up to the next odd value.

precompilation. The processing of a program containing Structured Query Language (SQL) statements that takes place before compilation. SQL statements are replaced with statements that is be recognized by the host language compiler.

precompiler. A program supporting precompilation of application programs with embedded Structured Query Language (SQL) statements.

predicate. In Database Manager, an element of a search condition expressing a comparison operation.

Presentation Interface. An API that allows users to write graphics applications.

procedure. (1) In a programming language, a block of code, with or without formal parameters, whose execution is invoked by means of a procedure call. (2) In Database Manager, a set consisting of Query Manager commands and procedure language statements or both. A procedure allows a single command to initiate operations.

procedure language statements. In Query Manager, the programming statements that are used in procedures.

process. A collection of system resources including one or more threads of execution that performs a task. In Communications Manager, a part of a configuration file.

prompt. (1) An action that users request while the cursor is in an

entry field. (2) A displayed message that requests input from the user or gives operational information.

prompted interface. An interface that consists of messages, menus, and panels that guides the user through the steps necessary to perform a task.

protocol. The set of rules governing the operation of functional units of a communication system that must be followed if communication is to take place.

Q

Query Manager. The part of Database Manager that provides menus, panels, and messages to assist in creating databases, editing data, generating reports, and making changes to Database Services configuration files.

R

RAM. See *random access memory*.

random access. An access mode in which records can be read from, written to, or removed from a file in any order.

random access memory (RAM). A memory device into which data is entered and from which data is retrieved in a nonsequential manner.

receive. (1) To obtain a message or file from another computer. Contrast with *send*. (2) In Communications Manager, the command used to transfer a file from a host.

record. (1) A set of data treated as a unit. (2) In Database Manager, the storage representation of a single row of a table.

record format. The definition of how data is structured in the records contained in a file. The definition includes record names, field names, and field descriptions, such as length and data type.

recovery. (1) The act of resetting a system or data stored in a system to an operable state following damage. (2) In Database Manager, the process of rebuilding databases after a system failure.

recovery log. A collection of records describing the sequence of events that occur while running Database Manager. The information is used for recovery in the event of a system failure while Database Manager is running.

relational command processor. The part of Database Services that processes Structured Query Language (SQL) statements.

relational database. (1) A database that is organized and accessed according to relationships between data items. (2) A data structure perceived by its users as a collection of tables.

replace mode. An entry mode that replaces existing characters with new characters at the text cursor location.

request. In SNA, a message unit that signals initiation of an action or protocol.

row. In Database Manager, the horizontal component of a table consisting of a sequence of values, one for each column of the table.

run. To cause a program, object, utility, or other machine function to be performed.

S

serial device. A resource (such as a modem or plotter) attached to an LPT or COM port for direct I/O use. Some printers are designated *serial* as well.

service coordinator. A person in the customer account that is responsible for ensuring problem determination support for the OS/2 program.

session. (1) A logical connection between two stations or network addressable units (NAUs) that allows them to communicate. (2) The period of time during which a user can communicate with an interactive system. (3) For the OS/2 program, see *screen group*. (4) In Database Manager, a group of processes (or tasks) associated with an application. See *half session*.

storage. A media used to save information, such as a fixed disk. Contrast with *memory*.

subdirectory. A directory contained within another directory in a file system hierarchy.

swap file. A file that contains segments of a program or data temporarily moved out of main memory.

swapping. A process that moves segments between memory and storage.

synchronous. Pertaining to two or more processes that depend upon the occurrences of specific events such as a common timing signal. Contrast with *asynchronous*.

system. A computer and its associated devices and programs. See *workstation*.

system administrator. Person or persons responsible for planning Communications Manager installation and ensuring the successful installation and use of the product by other users.

system diskette. A fixed disk or diskette that can be used to start (IPL) a system or workstation.

System Editor. An ASCII text editor provided as part of the OS/2 program.

system trace utility. In the OS/2 program, a diagnostic tool used to capture a sequence of system events, function calls, or data for analysis.

T

table. In Database Manager, a named collection of data consisting of rows and columns.

task. A set of one or more sequences of instructions treated by a control program as an element of work to be accomplished.

Task Manager. In the OS/2 program, the function that controls

the starting and stopping of programs, and which program has the input focus. It also allows the user to shut down the system.

template. In Database Manager, a copy of an existing table or view that is modified and saved to create a new table or view.

temporary storage. (1) In computer programming, the memory locations reserved for intermediate results. Synonymous with *working storage*. (2) In Database Manager, an area where the query, form, procedure, panel, and menu objects can be viewed on the screen by the user.

terminal. In data communication, a device, usually equipped with a keyboard and display, capable of sending and receiving information.

token. A character string in a specific format that has some defined significance in a Structured Query Language (SQL) or Query Manager application program.

U

user. A person who uses the system to perform nonadministrative tasks.

user interface. The hardware, software, or both that allows a user to interact with and perform operations on a system, program, or device.

user profile. An OS/2 command file containing commands that set environment values and run programs automatically when a user logs on.

V

value. (1) A specific occurrence of an attribute; for example, *blue* for the attribute *color*. (2) A quantity assigned to a constant, a variable, parameter, or symbol. See *argument*. (3) In Database Services, a data element with an assigned row and column in a table.

variable. A quantity that can assume any of a given set of values.

video input/output (VIO). An API used to write data directly to the screen.

VIO. See *video input/output*.

W

warning message. The information provided by the computer to the user to alert them to a possible error condition in the system. Warning messages are also sent to warn the user that the action they are attempting can cause an undesirable condition or consequence.

work area. (1) An area in which terminal devices (such as displays, keyboards, and printers) are located. (2) An area reserved for temporary storage of data.

work sheet formats (WSF). An OS/2 file format used to import and export data in work-sheet formats supported by the LOTUS products.

workstation. A terminal or personal computer, usually one that is connected to a mainframe or to a network, at which a user can run applications.

Numerics

3270 terminal emulation. A program installed into a personal computer to allow it to emulate a 3270 workstation.

Index

A

ACSPCSYS.EXE system
 process 2-126
ACSTRSYS.EXE system
 process 2-127
active partitions 2-77
ANSI command 2-2
ANSI.SYS device driver 2-32
APPEND command 2-3
append option 3-7, 3-14, 3-20, 3-27
appending files 2-22
applying software repairs 2-106
Arabic-speaking
 code pages supported 2-26
 country code 2-26
archive bit of a file 2-6
ASCII comparison 2-21
ASCII option 3-8, 3-15, 3-20, 3-27,
 3-33, 3-40
ASCII (text) file copying 2-22
ASSIGN command 2-5
assignment statements 5-6
 expressions in 5-6
 variables 5-6
ASYNCCDDA.SYS device
 driver 2-33
ASYNCCDDB.SYS device
 driver 2-33
asynchronous communication port,
 status in OS/2 mode 2-97
asynchronous communications
 device driver 2-35
asynchronous communications
 mode 2-99
ATTRIB command 2-6
Australia
 code pages supported 2-26
 country code 2-26
automated device driver 2-29

AVBLOCK(n) parameter 3-23

B

backing up
 disks 2-7
 files 2-7
 subdirectories 2-7
BACKUP command 2-7
backup files
 creating 2-7
 restoring 2-121
backup log file 2-7
BACKUP.LOG file 2-7
batch commands
 CALL 2-12
 ECHO 2-71
 ENDLOCAL 2-72
 EXTPROC 2-75
 FOR 2-79
 GOTO 2-82
 IF 2-85
 PAUSE 2-108
 REM (remarks) 2-118
 SET 2-128
 SETLOCAL 2-131
 SHIFT 2-133
batch file
batch files
 displaying remarks in 2-118
 suspending processing of 2-108
BEGIN WORK command 4-2
Belgium
 code pages supported 2-26
 country code 2-26
 keyboard 2-90
binary comparison 2-21
binary file copying 2-22
binary option 3-34, 3-41
blksize(n) parameter 3-21

BREAK

DOS command 2-9

BUFFERS statement in
CONFIG.SYS 2-11

C

CALL batch command 2-12

Canada (French-speaking)

code pages supported 2-26

country code 2-26

Canada (French)

keyboard 2-90

CANCEL WORK command 4-4

CD command 2-14

changing a search path

APPEND 2-3

DPATH 2-70

PATH 2-107

changing code pages 2-13

changing directories 2-14

changing file attributes 2-6

changing file names 2-119

changing keyboards 2-89

changing the clock

DATE command 2-28

TIME command 2-146

changing the current directory 2-14

changing the disk name 2-92

changing the prompt 2-113

character

strings 5-3

formats for 5-3

characters per line 2-103

CHCP command 2-13

CHDIR command 2-14

CHKDSK command 2-15

CICS format, receive 3-42

CICS format, send 3-31

clearing the screen 2-16

clock, resetting 2-28, 2-146

CLS command 2-16

CM 3-2

CMD command 2-17

code page switching 2-13

preparing devices for 2-63

code pages

names of 2-18

preparing devices to use 2-63

specifying with

CODEPAGE 2-18

supported by OS/2 2-18

CODEPAGE statement in
CONFIG.SYS 2-18

collating sequence used by
SORT 2-25

COM port in DOS mode 2-130

combining files 2-22

COMMAND command 2-19

command diagram A-3

external commands A-3

command file 3-2, 3-3

command format 3-1

CICS, receive 3-42

CICS, send 3-31

TSO, receive 3-25

TSO, SEND 3-18

VM, receive 3-16

VM, SEND 3-10

command parameters A-1

command processor

DOS mode 2-19

OS/2 mode 2-17

command prompt

changing the default 2-113

for DOS mode 2-113

for OS/2 mode 2-113

commands 3-1

ANSI 2-2

APPEND 2-3

ASSIGN 2-5

ATTRIB 2-6

BACKUP 2-7

batch

CALL 2-12

ECHO 2-71

ENDLOCAL 2-72

EXTPROC 2-75

FOR 2-79

commands (*continued*)

batch (*continued*)

GOTO 2-82
IF 2-85
PAUSE 2-108
REM (remarks) 2-118
SETLOCAL 2-131
SHIFT 2-133
BEGIN WORK 4-2
BREAK 2-9
CANCEL WORK 4-4
CHCP 2-13
CHDIR 2-14
CHKDSK 2-15
CLS 2-16
CMD 2-17
COMMAND 2-19
COMP 2-21
COPY 2-22
CREATEDD 2-27
DATE 2-28
DDINSTAL 2-29
DEL (see ERASE)
DETACH 2-30
DIR 2-66
DISKCOMP 2-68
DISKCOPY 2-69
DPATH 2-70
END WORK 4-5
ERASE 2-73, 4-7
EXIT 2-74
EXPORT 4-9
EXPORT command 4-10
FDISK 2-77
file transfer 3-5
FIND 2-78
FORMAT 2-80
GET variable 4-11
GRAFTABL 2-83
HELP 2-84
how to enter A-1
IMPORT 4-14
JOIN 2-87
KEYB 2-89

commands (*continued*)

LABEL 2-92
MKDIR 2-96
MODE 2-97
MORE 2-105
PATCH 2-106
PATH 2-107
PRINT 2-110, 4-17
PROMPT 2-113
RECOVER 2-117
REN 2-119
RENAME 2-119
REPLACE 2-120
RESTORE 2-121
RMDIR 2-123
RUN 4-21
SAVE DATA AS 4-24
SET 2-128
SET variable 4-26
SETCOM40 2-130
SHIFT 2-133
SORT 2-134
SPOOL 2-135
SQLQMFM 4-28
SQLQMFV 4-30
START 2-137
starting 3-2, 3-3
SUBST 2-141
TIME 2-146
TRACE 2-148
TRACEFMT 2-151
TREE 2-152
TYPE 2-153
UNPACK 2-154
VER 2-155
VERIFY 2-156
VOL 2-157
XCOPY 2-158
COMMENT keyword in SAVE
DATA AS 4-24
comment-string in SAVE DATA
AS 4-24
comments
displaying by a batch file 2-108

comments (*continued*)
 in a batch file 2-118
 in CONFIG.SYS file 2-118
 in procedure language
 statements 5-5
 COMP command 2-21
 comparing diskettes 2-68
 comparing files 2-21
 COM0x.SYS device driver 2-35
 CONFIG.SYS statements
 BREAK 2-9
 BUFFERS 2-11
 CODEPAGE 2-18
 COUNTRY 2-25
 DDINSTAL 2-29
 DEVICE 2-31
 DEVINFO 2-63
 DISKCACHE 2-67
 FCBS 2-76
 IOPL 2-86
 LIBPATH 2-93
 MAXWAIT 2-94
 MEMMAN 2-95
 PAUSEONERROR 2-109
 PRIORITY 2-112
 PROTECTONLY 2-115
 PROTSHELL 2-116
 REM (remarks) 2-118
 RMSIZE 2-124
 RUN 2-125
 SET 2-128
 SHELL 2-132
 SWAPPATH 2-143
 THREADS 2-145
 TIMESLICE 2-147
 TRACE 2-148
 TRACEBUF 2-150
 CONFIRM keyword
 in EXPORT 4-10
 in PRINT 4-19
 in SAVE DATA AS 4-24
 constant value in SET 4-27
 constants 5-3
 control keys 1-2
 COPY command 2-22
 copying
 all files on a diskette 2-69
 archived files 2-159
 changed files 2-158
 diskettes 2-69
 entire tree structure 2-159
 files 2-22
 groups of files 2-158
 subdirectories 2-158
 correcting disk errors 2-15
 correcting file errors 2-15
 countries
 code pages 2-25
 country codes 2-25
 country codes
 Arabic-speaking 2-26
 Australia 2-26
 Belgium 2-26
 Canada (French-speaking) 2-26
 Denmark 2-26
 Finland 2-26
 France 2-26
 Germany 2-26
 Hebrew-speaking 2-26
 Italy 2-26
 Japan 2-26
 Korea 2-26
 Latin America 2-26
 Netherlands 2-26
 Norway 2-26
 Portugal 2-26
 Simplified Chinese 2-26
 Spain 2-26
 Sweden 2-26
 Switzerland 2-26
 Traditional Chinese 2-26
 United Kingdom 2-26
 United States 2-26
 COUNTRY statement in
 CONFIG.SYS 2-25
 CREATEDD command 2-27

- creating a directory 2-96
- crlf option 3-9, 3-16, 3-21, 3-29, 3-34, 3-41
- Ctrl and Z keys 2-23
- Ctrl+Break 2-9
- Ctrl+Num Lock keys 1-2
- current directory
 - changing 2-14
 - displaying the name of 2-14
- CURRENT parameter for GET 4-12
- CURRENT parameter in SET 4-26

D

- data set name 3-19, 3-27
- DATAFORMAT keyword in EXPORT 4-10
- DATAFORMAT keyword in IMPORT 4-15
- DATE command 2-28
- date format 2-25
- DATETIME keyword in PRINT 4-18
- DDINSTAL command 2-29
- decimal separator symbol 2-25
- defining code pages 2-18
- DEL command (see ERASE)
- deleting a directory 2-123
- deleting a joined drive 2-87
- deleting a substituted drive 2-141
- deleting files 2-73
- deleting the disk name 2-92
- Denmark
 - code pages supported 2-26
 - country code 2-26
 - keyboard 2-90
- DETACH command 2-30
- detaching programs 2-30
- device driver installation 2-29
- device drivers 2-31
 - ACSPCSYS.EXE system process 2-126
 - ACSTRSYS.EXE system process 2-127
- device drivers (*continued*)
 - ANSI.SYS 2-32
 - ASYNCCDDA.SYS 2-33
 - ASYNCCDDB.SYS 2-33
 - COM0x.SYS 2-35
 - DFTDD.SYS 2-38
 - EGA.SYS 2-39
 - EXTDSKDD.SYS 2-40
 - LANDD.SYS 2-44
 - MOUSExxx.SYS 2-45
 - PMDD.SYS 2-53
 - POINTDD.SYS 2-54
 - SDLCDD.SYS 2-55
 - specifying path and file name 2-31
 - VDISK.SYS 2-60
- device driver, automated 2-29
- device driver, external 2-29
- device names, reserved 1-2
- device operation modes 2-97
- DEVICE statement in CONFIG.SYS 2-31
- device support 2-29
- devices, preparing for code page switching 2-63
- DEVINFO statement in CONFIG.SYS 2-63
- DFTDD.SYS device driver 2-38
- differences
 - OS/2 mode and DOS mode commands B-1
- DIR command 2-66
- directories
 - changing 2-14
 - directories, creating 2-96
 - directories, listing files in 2-66
 - directories, making 2-96
 - displaying files in 2-66, 2-152
 - displaying name of 2-14
 - searching for data files 2-3, 2-70
 - searching for programs and commands 2-107
- directory append 2-3
 - APPEND 2-3

- disk
 - backing up files 2-7
 - checking for errors 2-15
 - displaying name of 2-157
 - formatting 2-80
 - naming 2-81, 2-92
 - organizing 2-96
 - partitioning a fixed disk 2-77
 - restoring backed up files 2-121
 - restoring files 2-121
 - DISKCACHE statement in CONFIG.SYS 2-67
 - DISKCOMP command 2-68
 - DISKCOPY command 2-69
 - diskette comparing 2-68
 - diskette copying 2-69
 - display adapters supporting code page switching 2-63
 - displaying
 - code pages 2-13
 - directories 2-152
 - disk name 2-157
 - environment values 2-128
 - file contents 2-153
 - file names in directory 2-66
 - joined drives 2-87
 - name of current directory 2-14
 - paths to data files 2-3, 2-70
 - paths to executable files 2-107
 - subdirectories 2-152
 - system prompt settings 2-113
 - version number of OS/2 2-155
 - volume label 2-157
 - DO END 5-7
 - eol in 5-7
 - instruction lists 5-7
 - DO TO BY 5-8
 - eol in 5-8
 - expressions in 5-8
 - variables in 5-8
 - DO UNTIL 5-10
 - eol in 5-10
 - expressions in 5-10
 - instruction lists 5-10
 - DO UNTIL (*continued*)
 - LEAVE statement 5-11
 - DO WHILE 5-12
 - eol in 5-12
 - expressions in 5-12
 - instruction list in 5-12
 - LEAVE statement 5-12
 - DOS asynchronous communication mode 2-97
 - DOS mode
 - displaying extended character set 2-83
 - extended display and keyboard support 2-32
 - Presentation Manager support 2-53
 - DOS mode only commands
 - APPEND 2-3
 - ASSIGN 2-5
 - BREAK 2-9
 - COMMAND 2-19
 - FCBS 2-76
 - GRAFTABL 2-83
 - JOIN 2-87
 - RMSIZE 2-124
 - SETCOM40 2-130
 - SHELL 2-132
 - SUBST 2-141
 - DPATH command 2-70
 - dynamic link libraries 2-93
- E**
- ECHO batch command 2-71
 - EGA.SYS device driver 2-39
 - END WORK command 4-5
 - end-of-file mark 2-22
 - ENDLOCAL batch command 2-72
 - environment string 2-128
 - environment values 2-128
 - environment variables 2-128
 - eol
 - in DO END 5-7
 - in DO TO BY 5-8
 - in DO UNTIL 5-10

eol (*continued*)

- in DO WHILE 5-12
- in IF THEN ELSE 5-14
- in procedure language statements 5-4

ERASE command 2-73, 4-7

- CONFIRM keyword 4-7
- object-name parameter 4-7

erasing a directory 2-123

erasing files 2-73

error messages

- in TRACE 5-19

error messages, online help for 2-84

EXIT 5-13

- expressions in 5-13

EXIT command 2-74

exiting a program 2-9

EXPORT command 4-9, 4-10

- CONFIRM keyword 4-10
- DATAFORMAT keyword 4-10
- filename for 4-9
- Import/Export messages 4-10
- object-name parameter 4-9
- QRWEXPRT.LOG file 4-10
- running the 4-10
- TO keyword 4-9

expressions 5-6

- in DO TO BY 5-8
- in DO UNTIL 5-10
- in DO WHILE 5-12
- in EXIT 5-13
- in IF THEN ELSE 5-14
- in SAY 5-17
- operators 5-2

ext 3-6, 3-13, 3-19, 3-26, 3-32, 3-39

EXTDSKDD.SYS device

- driver 2-40

extended display and keyboard support

- DOS mode 2-32
- OS/2 mode 2-2

extended partitions 2-77

external commands

- ANSI 2-2
- APPEND 2-3
- ASSIGN 2-5
- ATTRIB 2-6
- BACKUP 2-7
- CHKDSK 2-15
- CMD 2-17
- COMMAND 2-19
- COMP 2-21
- CREATEDD 2-27
- definition 1-1
- DISKCOMP 2-68
- DISKCOPY 2-69
- FDISK 2-77
- FIND 2-78
- FORMAT 2-80
- GRAFTABL 2-83
- HELP 2-84
- JOIN 2-87
- KEYB 2-89
- LABEL 2-92
- MODE 2-97
- MORE 2-105
- PATCH 2-106
- PRINT 2-110
- RECOVER 2-117
- REPLACE 2-120
- RESTORE 2-121
- SETCOM40 2-130
- SORT 2-134
- SPOOL 2-135
- SUBST 2-141
- TRACE 2-148
- TRACEFMT 2-151
- TREE 2-152
- UNPACK 2-154
- XCOPY 2-158

EXTPROC batch command 2-75

F

FCBS statement in
CONFIG.SYS 2-76

FDISK command 2-77
 file attributes, changing 2-6
 file control block 2-76
 FILE keyword in PRINT 4-19
 file name displaying 2-66
 file name listing 2-66
 file names, specifying with DEVICE statements 2-31
 filemode 3-7, 3-14
 filename 3-6, 3-7, 3-13, 3-19, 3-26, 3-32, 3-33, 3-39, 3-40
 for EXPORT 4-9
 for IMPORT 4-15
 filename for IMPORT 4-15
 filename in syntax diagram A-4
 files
 backing up 2-7
 combining 2-22
 comparing 2-21, 2-68
 copying 2-22, 2-158
 copying all on a diskette 2-69
 displaying contents of 2-153
 displaying names of 2-66, 2-152
 erasing 2-73
 organizing 2-96
 printing 2-110, 2-135
 renaming 2-119
 restoring backed up 2-121
 sorting contents of 2-134
 UNPACK 2-154
 verifying data in 2-15, 2-156
 filetype 3-7, 3-14
 filtering B-12
 FIND command 2-78
 finding dynamic link libraries 2-93
 finding matching lines 2-78
 finding strings 2-78
 Finland
 code pages supported 2-26
 country code 2-26
 keyboard 2-90
 fixing disk errors 2-15
 fixing file errors 2-15
 FOR batch command 2-79
 foreign languages
 code pages for 2-25
 country dependent information for 2-25
 keyboard layouts for 2-89
 FORM keyword
 in PRINT 4-17
 in RUN 4-21
 form-name parameter in PRINT 4-17
 form-name parameter in RUN 4-21
 FORMAT command 2-80
 format for commands 3-1
 formatting diskettes 2-69
 formatting fixed disks and diskettes 2-80
 formatting to less than drive capacity 2-80
 France
 code pages supported 2-26
 country code 2-26
 keyboard 2-90
 FROM keyword in IMPORT 4-15

G

Germany
 code pages supported 2-26
 country code 2-26
 keyboard 2-90
 GET variable
 CURRENT parameter 4-12
 GLOBAL parameter 4-11
 LOCAL parameter 4-11
 procedure-variable 4-12
 varname for GET 4-12
 GET variable command 4-11
 global file name characters 1-2
 GLOBAL parameter for GET 4-11
 GLOBAL parameter in SET 4-26
 GOTO batch command 2-82
 GRAFTABL command 2-83

H

- Hebrew-speaking
 - code pages supported 2-26
 - country code 2-26
- HELP command 2-84
- help with warning and error messages 2-84
- high capacity diskette
 - formatting 2-80
- H(nnn) option 3-8, 3-15, 3-20, 3-28, 3-33, 3-40

I

- icons, meaning of 1-2
- id 3-6, 3-13, 3-19, 3-26, 3-32, 3-39
- IF batch command 2-85
- IF THEN ELSE 5-14
 - eol in 5-14
 - expressions in 5-14
 - instructions in 5-14
- IMPORT command 4-14
 - DATAFORMAT keyword 4-15
 - filename for 4-15
 - FROM keyword 4-15
 - Import/Export messages 4-16
 - object-name parameter 4-14
 - QRWIMPRT.LOG file 4-16
- Import/Export messages 4-10, 4-16
- installation, device driver 2-29
- instruction lists 5-4, 5-5, 5-7
 - in DO UNTIL 5-10
 - in DO WHILE 5-12
- instructions
 - in IF THEN ELSE 5-14
- internal commands
 - BREAK 2-9
 - CHCP 2-13
 - CHDIR 2-14
 - CLS 2-16
 - COPY 2-22
 - DATE 2-28
 - definition 1-1
 - DETACH 2-30

internal commands (*continued*)

- DIR 2-66
 - DPATH 2-70
 - ECHO 2-71
 - ERASE 2-73
 - EXIT 2-74
 - FOR 2-79
 - GOTO 2-82
 - IF 2-85
 - MKDIR 2-96
 - PATH 2-107
 - PAUSE 2-108
 - PROMPT 2-113
 - REM 2-118
 - RENAME 2-119
 - RMDIR 2-123
 - SET 2-128
 - SHIFT 2-133
 - START 2-137
 - TIME 2-146
 - TYPE 2-153
 - VER 2-155
 - VERIFY 2-156
 - VOL 2-157
 - IOPL statement in
 - CONFIG.SYS 2-86
 - Italy
 - code pages supported 2-26
 - country code 2-26
 - keyboard 2-90
- ## J
- Japan
 - code pages supported 2-26
 - country code 2-26
 - JOIN command 2-87
 - joining drives 2-87
- ## K
- KEYB command 2-89
 - keyword command parameters A-1
 - keywords
 - COMMENT 4-24

keywords (*continued*)

CONFIRM 4-7, 4-10, 4-19, 4-24
DATAFORMAT 4-10, 4-15
DATETIME 4-18
FILE 4-19
FORM 4-17
FROM 4-15
LENGTH 4-18
MODE 4-22
PAGENO 4-19
PRINTER 4-18
PRINTTYPE 4-19
REPORT 4-22
TO 4-9
WIDTH 4-18

Korea

code pages supported 2-26
country code 2-26

L

LABEL command 2-92
LANDD.SYS device driver 2-44
languages, foreign
code pages for 2-25
country dependent information
for 2-25
keyboard layouts for 2-89

Latin America

code pages supported 2-26
country code 2-26
keyboard 2-90

LEAVE 5-16

statement 5-12

LENGTH keyword in PRINT 4-18

LIBPATH statement in
CONFIG.SYS 2-93

listing file names 2-66

LOCAL parameter
for GET 4-11
for SET 4-26

locating data files 2-3, 2-70

locating dynamic link libraries 2-93

locating matching lines 2-78

locating programs and
commands 2-107

locating strings 2-78

log file (backup) 2-7

lrecl n option 3-9, 3-22

M

making a directory 2-96

MAXWAIT statement in
CONFIG.SYS 2-94

MD command 2-96

member name 3-20, 3-27

MEMMAN statement in
CONFIG.SYS 2-95

messages

import/export 4-10, 4-16

messages, online help for 2-84

migr option 3-9, 3-15, 3-21, 3-28,
3-34, 3-41

MKDIR command 2-96

MODE command 2-97

MODE keyword in RUN 4-22

MORE command 2-105

mouse installation 2-45

MOUSExxx.SYS device
driver 2-45

N

naming disk during format 2-81

national language code pages 2-25

nesting batch files 2-12

NETBDD.SYS

device drivers 2-49

NETBDD.SYS device
driver 2-49

Netherlands

code pages supported 2-26

country code 2-26

keyboard 2-90

nocrlf option 3-35, 3-41

noninteractive processing 2-30

Norway

code pages supported 2-26

Norway (*continued*)
country code 2-26
keyboard 2-90
numbers 5-4

O

object names
in ERASE 4-7
object-name parameter
in EXPORT 4-9
in IMPORT 4-14
in PRINT 4-17
in RUN 4-21
operating modes
command differences B-1
optional command parameters A-2
OS/2 asynchronous communication
mode 2-97
OS/2 mode
extended display and keyboard
support 2-2
OS/2 mode only commands
ANSI 2-2
CMD 2-17
CREATEDD 2-27
DETACH 2-30
DISKCACHE 2-67
DPATH 2-70
FDISK 2-77
IOPL 2-86
KEYB 2-89
LIBPATH 2-93
MEMMAN 2-95
PROTSHELL 2-116
RUN 2-125
SPOOL 2-135
START 2-137
SWAPPATH 2-143
THREADS 2-145
TRACE 2-148
TRACEFMT 2-151
OS/2 version number 2-155
overview
batch commands B-13

overview (*continued*)
CONFIG.SYS statements B-3
country commands B-15
device statements B-5
directory commands B-10
disk and diskette
commands B-11
file commands B-9
memory statements B-14
OS/2 commands B-1
piping and filtering B-12
problem determination
commands B-16
system information
commands B-7
system software repair B-16
O(off) in TRACE 5-18

P

PAGENO keyword in PRINT 4-19
parallel printer mode 2-103
parameters for start command 3-2,
3-3
partitioning the fixed disk 2-77
partitions
active 2-77
changing 2-77
creating 2-77
deleting 2-77
displaying 2-77
password 3-20, 3-27
PATCH command 2-106
PATH command 2-107
paths to data files in DOS
mode 2-3
paths to data files in OS/2
mode 2-70
paths to device drivers 2-31
paths to executable files 2-107
PAUSE batch command 2-108
PAUSEONERROR statement in
CONFIG.SYS 2-109
PCNETDD.SYS
device drivers 2-51

PCNETDD.SYS (*continued*)
 PCNETDD.SYS device
 driver 2-51
 piping B-12
 PMDD.SYS device driver 2-53
 POINTDD.SYS device driver 2-54
 pointing device 2-45
 Portugal
 code pages supported 2-26
 country code 2-26
 keyboard 2-90
 predicates 5-3
 preparing devices for code page
 switching 2-63
 prerequisite publications vi
 Presentation Manager pointer draw
 support
 DOS mode 2-53
 primary partitions 2-77
 PRINT 4-17
 CONFIRM keyword 4-19
 DATETIME keyword 4-18
 FILE keyword 4-19
 FORM keyword 4-17
 form-name parameter 4-17
 LENGTH keyword 4-18
 object-name parameter 4-17
 PAGENO keyword 4-19
 print options parameter 4-18
 PRINTER keyword 4-18
 PRINTTYPE keyword in
 PRINT 4-19
 WIDTH keyword 4-18
 PRINT command 2-110
 print options parameter 4-18
 print queue 2-110
 print spooler 2-135
 PRINTER keyword in
 PRINT 4-18
 printers supporting code page
 switching 2-65
 printing 2-110
 printing characters 2-103
 printing 6 or 8 lines per inch 2-103
 PRINTTYPE keyword 4-19
 priority calculation
 absolute 2-112
 dynamic 2-112
 PRIORITY statement in
 CONFIG.SYS 2-112
 problem determination
 with service programs B-16
 procedural language
 parameters 5-1
 procedure language statements 5-1
 assignment statements 5-6
 character strings 5-3
 comments in 5-5
 constants in 5-3
 DO END 5-7
 DO TO BY 5-8
 DO UNTIL 5-10
 DO WHILE 5-12
 eol 5-4
 EXIT 5-13
 expressions 5-2
 IF THEN ELSE 5-14
 instruction lists in 5-5
 instructions in 5-4
 language syntax 5-1
 LEAVE 5-16
 notational conventions 5-1
 numbers 5-4
 parameters 5-1
 predicates in 5-3
 procedure variables in 5-2
 Query Manager commands 5-5
 Query Manager variables 5-2
 SAY 5-17
 TRACE 5-18
 variables in 5-1
 procedure variables 5-2
 for GET 4-12
 for SET 4-27
 PROMPT command 2-113
 PROTECTONLY statement in
 CONFIG.SYS 2-115

PROTSHELL statement in
 CONFIG.SYS 2-116
punctuation in commands A-2
p(nnn) option 3-8, 3-15, 3-21, 3-28,
 3-34, 3-41

Q

QRWEXPRT.LOG file 4-10
QRWIMPRT.LOG file 4-16
QRWPROC.TRC file in
 TRACE 5-19
Query Manager
 commands 4-1
 alphabetical list of 4-1
 BEGIN WORK 4-2
 CANCEL WORK 4-4
 END WORK 4-5
 entered on the Command
 Line 4-1
 ERASE 4-7
 EXPORT 4-9
 EXPORT command 4-10
 GET 4-11
 IMPORT 4-14
 in procedures 4-1
 PRINT 4-17
 PRINT REPORT
 command 4-20
 RUN 4-21, 4-23
 running IMPORT 4-16
 SAVE 4-25
 SAVE DATA AS 4-24
 SET 4-26, 4-27, 4-28, 4-30
 syntax diagrams 4-1
 use of blanks 4-1
 use of quotes 4-1
 procedure language
 statements 5-1
 variables 5-2
Query Manager commands 5-5
querying code pages 2-13
quick reference to
 batch commands B-13
 country commands B-15

quick reference to (*continued*)
 directory commands B-10
 disk and diskette
 commands B-11
 file commands B-9
 memory statements B-14
 OS/2 commands B-1
 piping and filtering B-12
 problem determination
 commands B-16
 statements for
 CONFIG.SYS B-3
 statements for devices B-5
 system information
 commands B-7
quitting a program, BREAK
 command 2-9

R

RD command 2-123
read-only attribute of a file 2-6
reading symbols 1-2
reading syntax diagrams A-1
receive command 3-12, 3-25, 3-38
recfm f/v option 3-9
recfm f/v/u option 3-22
RECOVER command 2-117
recovering files 2-117
related publications vi
REM (remarks)
 batch command 2-118
remarks
 in batch file 2-118
 in CONFIG.SYS file 2-118
removing a directory 2-123
removing files 2-73
REN command 2-119
RENAME command 2-119
renaming files 2-119
repair, system software B-16
REPLACE command 2-120
REPORT keyword in RUN 4-22
reserved device names 1-2

RESTORE command 2-121
 restoring backed up files 2-121
 restoring subdirectories 2-122
 reverse sort 2-134
 RMDIR command 2-123
 RMSIZE statement in
 CONFIG.SYS 2-124
 rules for commands 1-1
 RUN command 4-21
 FORM command 4-21
 form-name parameter 4-21
 MODE keyword 4-22
 object-name parameter 4-21
 REPORT keyword 4-22
 RUN statement in
 CONFIG.SYS 2-125
 R(esults) in TRACE 5-18

S

SAVE command
 running the 4-25
 SAVE DATA AS
 COMMENT keyword 4-24
 SAVE DATA AS command 4-24
 comment-string in 4-24
 CONFIRM keyword 4-24
 table-name parameter 4-24
 SAY 5-17
 expressions in 5-17
 SDLCDD.SYS device driver 2-55
 searching directories
 for data files 2-3, 2-70
 for programs and
 commands 2-107
 searching for matching lines 2-78
 searching for strings 2-78
 segment swapping 2-95
 SEND command 3-5, 3-19, 3-31
 serial device driver 2-35
 serial DEVICE (COM) ports 2-97
 serial devices
 device driver for 2-35
 setting operation mode for 2-99
 setting the DOS port
 address 2-130

serial devices in DOS mode 2-130
 service programs B-16
 SET command 2-128
 constant value 4-27
 CURRENT parameter 4-26
 GLOBAL parameter 4-26
 LOCAL parameter 4-26
 procedure-variable in 4-27
 running 4-27
 varname in 4-26
 SET variable command 4-26
 SETCOM40 command 2-130
 SETLOCAL batch command 2-131
 setting a search path
 for data files 2-3, 2-70
 for programs and
 commands 2-107
 setting a search path to null
 APPEND 2-3
 DPATH 2-70
 PATH 2-107
 setting characters per line 2-103
 setting environment 2-128
 setting lines per inch 2-103
 setting parallel printer mode 2-103
 setting vertical spacing 2-103
 setting video mode 2-102
 SHELL statement in
 CONFIG.SYS 2-132
 SHIFT batch command 2-133
 showing a search path
 APPEND 2-3
 DPATH 2-70
 PATH 2-107
 showing environment 2-128
 showing joined drives 2-87
 showing name of current
 directory 2-14
 showing the prompt 2-113
 Simplified Chinese
 code pages supported 2-26
 country code 2-26
 SORT command 2-134

- sorting a file 2-134
- SPACE(q < ,i >) parameter 3-22
- Spain
 - code pages supported 2-26
 - country code 2-26
 - keyboard 2-90
- SPOOL command 2-135
- spooling printer files 2-135
- SQLQMFV command 4-28
- SQLQMFV command 4-30
- START command 2-137
- STARTCM 3-3
- starting programs automatically
- stopping a program 2-9
- storage compaction 2-95
- subdirectories
 - backing up 2-8
 - copying 2-158
 - restoring 2-122
- SUBST command 2-141
- substituting a drive letter for a drive and path 2-141
- SWAPPATH statement in CONFIG.SYS 2-143
- Sweden
 - code pages supported 2-26
 - country code 2-26
 - keyboard 2-90
- switching code pages 2-13
- Switzerland
 - code pages supported 2-26
 - country code 2-26
 - Swiss-French keyboard 2-90
 - Swiss-German keyboard 2-90
- symbols
 - decimal separator 2-25
- syntax diagrams
 - Query Manager 4-1
- syntax diagrams, reading A-1
- syntax of commands A-1
- SYSLEVEL
- system code pages 2-18
 - defining in CONFIG.SYS 2-18
 - names of 2-18

- system date 2-28
- system initialization
 - pausing for error display 2-109
- system software repair B-16
- system time 2-146
- system tracing 2-148, 2-150

T

- table of
 - batch commands B-13
 - CONFIG.SYS statements B-3
 - country commands B-15
 - device statements B-5
 - directory commands B-10
 - disk and diskette
 - commands B-11
 - file commands B-9
 - memory statements B-14
 - OS/2 commands B-1
 - pipng and filtering B-12
 - problem determination
 - commands B-16
 - system information
 - commands B-7
 - system software commands B-16
- table-name parameter in SAVE DATA AS 4-24
- THREADS statement in CONFIG.SYS 2-145
- time
 - displaying 2-146
 - setting 2-146
- TIME command 2-146
- time format 2-25
- TIMESLICE statement in CONFIG.SYS 2-147
- TO keyword 4-9
- TRACE 5-18
 - C:\SQLLIB directory 5-19
 - OS/2 command 2-148
 - O(off) in 5-18
 - QRWPROC.TRC file 5-19
 - R(esults) in 5-18

TRACEBUF statement in
 CONFIG.SYS 2-150
TRACEFMT command 2-151
tracks(n) parameter 3-23
Traditional Chinese
 code pages supported 2-26
 country code 2-26
TREE command 2-152
TRNETDD.SYS
 device drivers 2-58
 TRNETDD.SYS device
 driver 2-58
TSO format, receive 3-25
TSO format, SEND 3-18
TYPE command 2-153

U

United Kingdom
 code pages supported 2-26
 country code 2-26
 keyboard 2-90
United States
 code pages supported 2-26
 country code 2-26
 keyboard 2-90
UNPACK command 2-154
user interface 2-116

V

variable command parameters A-1
variables 5-1, 5-6
 in DO TO BY 5-8
varname
 for GET 4-12
 for SET 4-26
VDISK.SYS device driver 2-60
VER command 2-155
VERIFY command 2-156
verifying copy 2-159
verifying data in files 2-156
verifying during copy 2-23
version number 2-155

vertical spacing 2-103
video mode, setting 2-102
viewing a file 2-153
VM format, receive 3-16
VM format, SEND 3-10
VOL command 2-157
volume label (disk name)
 changing 2-92
 creating during format 2-81
 deleting 2-92
 viewing 2-157

W

warning messages, online help
 for 2-84
WIDTH keyword in PRINT 4-18

X

XCOPY command 2-158

Special Characters

.. (parent directory) 2-14